



**packetvideo™**

OSCL API

Build Version: CORE\_8.511.1.1

April 23, 2010

# Contents

<b>1 oscl Module Index</b>	<b>1</b>
1.1 oscl Modules . . . . .	1
<b>2 oscl Hierarchical Index</b>	<b>2</b>
2.1 oscl Class Hierarchy . . . . .	2
<b>3 oscl Data Structure Index</b>	<b>9</b>
3.1 oscl Data Structures . . . . .	9
<b>4 oscl File Index</b>	<b>15</b>
4.1 oscl File List . . . . .	15
<b>5 oscl Page Index</b>	<b>20</b>
5.1 oscl Related Pages . . . . .	20
<b>6 oscl Module Documentation</b>	<b>21</b>
6.1 OSCL config . . . . .	21
6.2 OSCL Base . . . . .	25
6.3 OSCL Memory . . . . .	47
6.4 OSCL Util . . . . .	63
6.5 OSCL Error . . . . .	85
6.6 OSCL IO . . . . .	95
6.7 OSCL Proc . . . . .	103
6.8 OSCL Init . . . . .	107
<b>7 oscl Data Structure Documentation</b>	<b>108</b>
7.1 _OsclBasicAllocator Class Reference . . . . .	108
7.2 _OsclHeapBase Class Reference . . . . .	110
7.3 AcceptParam Class Reference . . . . .	112
7.4 allocator Class Reference . . . . .	113

7.5	AllPassFilter Class Reference . . . . .	114
7.6	BindParam Class Reference . . . . .	116
7.7	BufferFragment Class Reference . . . . .	117
7.8	BufferMgr Class Reference . . . . .	118
7.9	BufferState Class Reference . . . . .	119
7.10	BuFragGroup< ChainClass, max_frags > Class Template Reference . . . . .	120
7.11	BuffFragStatusClass Class Reference . . . . .	123
7.12	CallbackTimer< Alloc > Class Template Reference . . . . .	124
7.13	CallbackTimerObserver Class Reference . . . . .	126
7.14	CFastRep Class Reference . . . . .	127
7.15	CHheapRep Class Reference . . . . .	129
7.16	ConnectParam Class Reference . . . . .	131
7.17	CStackRep Class Reference . . . . .	132
7.18	DNSRequestParam Class Reference . . . . .	133
7.19	GetHostByNameParam Class Reference . . . . .	135
7.20	HeapBase Class Reference . . . . .	137
7.21	internalLeave Class Reference . . . . .	139
7.22	LinkedListElement< LLClass > Class Template Reference . . . . .	140
7.23	ListenParam Class Reference . . . . .	141
7.24	MediaData< ChainClass, max_frags, local_bufsize > Class Template Reference . . . . .	142
7.25	MediaStatusClass Class Reference . . . . .	145
7.26	MemAllocator< T > Class Template Reference . . . . .	146
7.27	MM_AllocBlockFence Struct Reference . . . . .	147
7.28	MM_AllocBlockHdr Struct Reference . . . . .	148
7.29	MM_AllocInfo Struct Reference . . . . .	149
7.30	MM_AllocNode Struct Reference . . . . .	151
7.31	MM_AllocQueryInfo Struct Reference . . . . .	152
7.32	MM_Audit_Imp Class Reference . . . . .	153
7.33	MM_AuditOverheadStats Struct Reference . . . . .	161
7.34	MM_FailInsertParam Struct Reference . . . . .	162
7.35	MM_Stats_CB Struct Reference . . . . .	163
7.36	MM_Stats_t Struct Reference . . . . .	164
7.37	NTPTTime Class Reference . . . . .	166
7.38	Oscl_Alloc Class Reference . . . . .	170
7.39	Oscl_Dealloc Class Reference . . . . .	171
7.40	Oscl_DefAlloc Class Reference . . . . .	172

7.41 Oscl_DefAllocWithRefCounter< DefAlloc > Class Template Reference . . . . .	173
7.42 OSCL_FastString Class Reference . . . . .	175
7.43 Oscl_File Class Reference . . . . .	179
7.44 Oscl_File::OsclCacheObserver Class Reference . . . . .	187
7.45 Oscl_File::OsclFixedCacheParam Class Reference . . . . .	188
7.46 Oscl_FileFind Class Reference . . . . .	189
7.47 Oscl_FileServer Class Reference . . . . .	193
7.48 oscl_fsstat Struct Reference . . . . .	195
7.49 OSCL_HeapString< Alloc > Class Template Reference . . . . .	196
7.50 OSCL_HeapStringA Class Reference . . . . .	198
7.51 Oscl_Int64_Utils Class Reference . . . . .	203
7.52 Oscl_Less< T > Struct Template Reference . . . . .	205
7.53 Oscl_Linked_List< LLClass, Alloc > Class Template Reference . . . . .	206
7.54 Oscl_Linked_List_Base Class Reference . . . . .	211
7.55 Oscl_Map< Key, T, Alloc, Compare > Class Template Reference . . . . .	216
7.56 Oscl_Map< Key, T, Alloc, Compare >::value_compare Class Reference . . . . .	223
7.57 Oscl_MTLinked_List< LLClass, Alloc, TheLock > Class Template Reference . . . . .	225
7.58 Oscl_Opaque_Type_Alloc Class Reference . . . . .	229
7.59 Oscl_Opaque_Type_Alloc_LL Class Reference . . . . .	231
7.60 Oscl_Opaque_Type_Compare Class Reference . . . . .	233
7.61 Oscl_Pair< T1, T2 > Struct Template Reference . . . . .	235
7.62 Oscl_Queue< T, Alloc > Class Template Reference . . . . .	236
7.63 Oscl_Queue_Base Class Reference . . . . .	239
7.64 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference . . . . .	242
7.65 Oscl_Rb_Tree_Base Class Reference . . . . .	246
7.66 Oscl_Rb_Tree_Const_Iterator< Value > Struct Template Reference . . . . .	247
7.67 Oscl_Rb_Tree_Iterator< Value > Struct Template Reference . . . . .	250
7.68 Oscl_Rb_Tree_Node< Value > Struct Template Reference . . . . .	253
7.69 Oscl_Rb_Tree_Node_Base Struct Reference . . . . .	254
7.70 Oscl_Select1st< V, U > Struct Template Reference . . . . .	256
7.71 OSCL_StackString< MaxBufSize > Class Template Reference . . . . .	257
7.72 oscl_stat_buf Struct Reference . . . . .	259
7.73 OSCL_String Class Reference . . . . .	260
7.74 Oscl_Tag< Alloc > Struct Template Reference . . . . .	265
7.75 Oscl_Tag_Base Struct Reference . . . . .	267
7.76 Oscl_TagTree< T, Alloc > Class Template Reference . . . . .	269

7.77 Oscl_TagTree< T, Alloc >::const_iterator Struct Reference . . . . .	273
7.78 Oscl_TagTree< T, Alloc >::iterator Struct Reference . . . . .	276
7.79 Oscl_TagTree< T, Alloc >::Node Struct Reference . . . . .	279
7.80 Oscl_TAlloc< T, Alloc > Class Template Reference . . . . .	281
7.81 Oscl_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference . . . . .	284
7.82 Oscl_Vector< T, Alloc > Class Template Reference . . . . .	285
7.83 Oscl_Vector_Base Class Reference . . . . .	290
7.84 OSCL_wFastString Class Reference . . . . .	294
7.85 OSCL_wHeapString< Alloc > Class Template Reference . . . . .	297
7.86 OSCL_wHeapStringA Class Reference . . . . .	299
7.87 OSCL_wStackString< MaxBufSize > Class Template Reference . . . . .	302
7.88 OSCL_wString Class Reference . . . . .	304
7.89 OsclAcceptMethod Class Reference . . . . .	308
7.90 OsclAcceptRequest Class Reference . . . . .	309
7.91 OsclActiveObject Class Reference . . . . .	310
7.92 OsclAllocDestructDealloc Class Reference . . . . .	314
7.93 OsclAOStatus Class Reference . . . . .	315
7.94 OsclAsyncFile Class Reference . . . . .	316
7.95 OsclAsyncFileBuffer Class Reference . . . . .	319
7.96 OsclAuditCB Class Reference . . . . .	321
7.97 OsclBindMethod Class Reference . . . . .	322
7.98 OsclBindRequest Class Reference . . . . .	323
7.99 OsclBinIStream Class Reference . . . . .	324
7.100 OsclBinIStreamBigEndian Class Reference . . . . .	326
7.101 OsclBinIStreamLittleEndian Class Reference . . . . .	329
7.102 OsclBinOStream Class Reference . . . . .	331
7.103 OsclBinOStreamBigEndian Class Reference . . . . .	332
7.104 OsclBinOStreamLittleEndian Class Reference . . . . .	334
7.105 OsclBinStream Class Reference . . . . .	336
7.106 OsclBuf Class Reference . . . . .	340
7.107 OsclCompareLess< T > Class Template Reference . . . . .	342
7.108 OsclComponentRegistry Class Reference . . . . .	343
7.109 OsclComponentRegistryData Class Reference . . . . .	345
7.110 OsclComponentRegistryElement Class Reference . . . . .	346
7.111 OsclConnectMethod Class Reference . . . . .	348
7.112 OsclConnectRequest Class Reference . . . . .	349

7.113OsclDestructDealloc Class Reference . . . . .	350
7.114OsclDNS Class Reference . . . . .	351
7.115OsclDNSI Class Reference . . . . .	353
7.116OsclDNSIBase Class Reference . . . . .	355
7.117OsclDNSMethod Class Reference . . . . .	358
7.118OsclDNSObserver Class Reference . . . . .	361
7.119OsclDNSRequest Class Reference . . . . .	362
7.120OsclDNSRequestAO Class Reference . . . . .	363
7.121OsclDoubleLink Class Reference . . . . .	366
7.122OsclDoubleList< T > Class Template Reference . . . . .	367
7.123OsclDoubleListBase Class Reference . . . . .	368
7.124OsclDoubleRunner< T > Class Template Reference . . . . .	370
7.125OsclError Class Reference . . . . .	372
7.126OsclErrorAllocator Class Reference . . . . .	374
7.127OsclErrorTrap Class Reference . . . . .	376
7.128OsclErrorTrapImp Class Reference . . . . .	377
7.129OsclException< LeaveCode > Class Template Reference . . . . .	379
7.130OsclExclusiveArrayPtr< T > Class Template Reference . . . . .	380
7.131OsclExclusivePtr< T > Class Template Reference . . . . .	383
7.132OsclExclusivePtrA< T, Alloc > Class Template Reference . . . . .	386
7.133OsclExecScheduler Class Reference . . . . .	389
7.134OsclExecSchedulerBase Class Reference . . . . .	391
7.135OsclExecSchedulerCommonBase Class Reference . . . . .	392
7.136OsclFileCache Class Reference . . . . .	401
7.137OsclFileCacheBuffer Class Reference . . . . .	403
7.138OsclFileHandle Class Reference . . . . .	405
7.139OsclFileManager Class Reference . . . . .	406
7.140OsclFileStats Class Reference . . . . .	411
7.141OsclFileStatsItem Class Reference . . . . .	412
7.142OsclGetHostByNameMethod Class Reference . . . . .	413
7.143OsclGetHostByNameRequest Class Reference . . . . .	414
7.144OsclInit Class Reference . . . . .	415
7.145OsclInteger64Transport Struct Reference . . . . .	416
7.146OsclIpMReq Class Reference . . . . .	417
7.147OsclIPSocketI Class Reference . . . . .	418
7.148OsclJump Class Reference . . . . .	421

7.149OsclListenMethod Class Reference . . . . .	422
7.150OsclListenRequest Class Reference . . . . .	423
7.151OsclLockBase Class Reference . . . . .	424
7.152OsclMem Class Reference . . . . .	425
7.153OsclMemAllocator Class Reference . . . . .	426
7.154OsclMemAllocDestructDealloc< T > Class Template Reference . . . . .	427
7.155OsclMemAudit Class Reference . . . . .	429
7.156OSCLMemAutoPtr< T, _Allocator > Class Template Reference . . . . .	435
7.157OsclMemBasicAllocator Class Reference . . . . .	439
7.158OsclMemBasicAllocDestructDealloc< T > Class Template Reference . . . . .	440
7.159OsclMemGlobalAuditObject Class Reference . . . . .	441
7.160OsclMemoryFragment Struct Reference . . . . .	442
7.161OsclMemPoolFixedChunkAllocator Class Reference . . . . .	443
7.162OsclMemPoolFixedChunkAllocatorObserver Class Reference . . . . .	447
7.163OsclMemPoolResizableAllocator Class Reference . . . . .	448
7.164OsclMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference . . . . .	454
7.165OsclMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference . . . . .	455
7.166OsclMemPoolResizableAllocatorMemoryObserver Class Reference . . . . .	456
7.167OsclMemPoolResizableAllocatorObserver Class Reference . . . . .	457
7.168OsclMemStatsNode Class Reference . . . . .	458
7.169OsclMutex Class Reference . . . . .	459
7.170OsclNameString< __len > Class Template Reference . . . . .	461
7.171OsclNativeFile Class Reference . . . . .	462
7.172OsclNativeFileParams Class Reference . . . . .	465
7.173OsclNetworkAddress Class Reference . . . . .	466
7.174OsclNullLock Class Reference . . . . .	467
7.175OsclPriorityLink Class Reference . . . . .	468
7.176OsclPriorityList< T > Class Template Reference . . . . .	469
7.177OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference . . . . .	470
7.178OsclPriorityQueueBase Class Reference . . . . .	474
7.179OsclProcStatus Class Reference . . . . .	475
7.180OsclPtr Class Reference . . . . .	477
7.181OsclPtrC Class Reference . . . . .	479
7.182OsclRand Class Reference . . . . .	481
7.183OsclReadyAlloc Class Reference . . . . .	482
7.184OsclReadyCompare Class Reference . . . . .	483

7.185OsclReadyQ Class Reference . . . . .	484
7.186OsclRecvFromMethod Class Reference . . . . .	486
7.187OsclRecvFromRequest Class Reference . . . . .	488
7.188OsclRecvMethod Class Reference . . . . .	490
7.189OsclRecvRequest Class Reference . . . . .	491
7.190OsclRefCounter Class Reference . . . . .	492
7.191OsclRefCounterDA Class Reference . . . . .	494
7.192OsclRefCounterMemFrag Class Reference . . . . .	496
7.193OsclRefCounterMTDA< LockType > Class Template Reference . . . . .	498
7.194OsclRefCounterMTSA< DeallocType, LockType > Class Template Reference . . . . .	500
7.195OsclRefCounterSA< DeallocType > Class Template Reference . . . . .	502
7.196OsclRegistryAccessClient Class Reference . . . . .	504
7.197OsclRegistryAccessClientImpl Class Reference . . . . .	506
7.198OsclRegistryAccessClientTlsImpl Class Reference . . . . .	507
7.199OsclRegistryAccessElement Class Reference . . . . .	508
7.200OsclRegistryClient Class Reference . . . . .	509
7.201OsclRegistryClientImpl Class Reference . . . . .	511
7.202OsclRegistryClientTlsImpl Class Reference . . . . .	513
7.203OsclRegistryServTlsImpl Class Reference . . . . .	514
7.204OsclScheduler Class Reference . . . . .	516
7.205OsclSchedulerObserver Class Reference . . . . .	517
7.206OsclScopedLock< LockClass > Class Template Reference . . . . .	518
7.207OsclSelect Class Reference . . . . .	519
7.208OsclSemaphore Class Reference . . . . .	521
7.209OsclSendMethod Class Reference . . . . .	523
7.210OsclSendRequest Class Reference . . . . .	524
7.211OsclSendToMethod Class Reference . . . . .	525
7.212OsclSendToRequest Class Reference . . . . .	526
7.213OsclSharedPtr< TheClass > Class Template Reference . . . . .	527
7.214OsclShutdownMethod Class Reference . . . . .	530
7.215OsclShutdownRequest Class Reference . . . . .	531
7.216OsclSingleton< T, ID, Registry > Class Template Reference . . . . .	532
7.217OsclSingletonRegistry Class Reference . . . . .	534
7.218OsclSocketI Class Reference . . . . .	535
7.219OsclSocketIBase Class Reference . . . . .	540
7.220OsclSocketMethod Class Reference . . . . .	545

7.221 OsclSocketObserver Class Reference . . . . .	548
7.222 OsclSocketRequest Class Reference . . . . .	549
7.223 OsclSocketRequestAO Class Reference . . . . .	550
7.224 OsclSocketServ Class Reference . . . . .	554
7.225 OsclSocketServI Class Reference . . . . .	556
7.226 OsclSocketServIBase Class Reference . . . . .	558
7.227 OsclSocketServRequestList Class Reference . . . . .	560
7.228 OsclSocketServRequestQElem Class Reference . . . . .	562
7.229 OsclSocketTOS Class Reference . . . . .	563
7.230 OsclTCPSocket Class Reference . . . . .	565
7.231 OsclTCPSocketI Class Reference . . . . .	572
7.232 OsclThread Class Reference . . . . .	575
7.233 OsclThreadLock Class Reference . . . . .	579
7.234 OsclTickCount Class Reference . . . . .	580
7.235 OsclTimer< Alloc > Class Template Reference . . . . .	582
7.236 OsclTimerCompare Class Reference . . . . .	585
7.237 OsclTimerObject Class Reference . . . . .	586
7.238 OsclTimerObserver Class Reference . . . . .	590
7.239 OsclTimerQ Class Reference . . . . .	591
7.240 OsclTLS< T, ID, Registry > Class Template Reference . . . . .	592
7.241 OsclTLSE< T, ID, Registry > Class Template Reference . . . . .	594
7.242 OsclTLSRegistry Class Reference . . . . .	596
7.243 OsclTLSRegistryEx Class Reference . . . . .	597
7.244 OsclTrapItem Class Reference . . . . .	598
7.245 OsclTrapStack Class Reference . . . . .	599
7.246 OsclTrapStackItem Class Reference . . . . .	600
7.247 OsclUDPSocket Class Reference . . . . .	601
7.248 OsclUDPSocketI Class Reference . . . . .	607
7.249 OsclUuid Struct Reference . . . . .	610
7.250 PVActiveBase Class Reference . . . . .	612
7.251 PVActiveStats Class Reference . . . . .	616
7.252 PVLogger Class Reference . . . . .	617
7.253 PVLoggerAppender Class Reference . . . . .	623
7.254 PVLoggerFilter Class Reference . . . . .	624
7.255 PVLoggerLayout Class Reference . . . . .	626
7.256 PVLoggerRegistry Class Reference . . . . .	628

7.257PVSchedulerStopper Class Reference . . . . .	631
7.258PVSockBufRecv Class Reference . . . . .	632
7.259PVSockBufSend Class Reference . . . . .	633
7.260PVThreadContext Class Reference . . . . .	634
7.261RecvFromParam Class Reference . . . . .	636
7.262RecvParam Class Reference . . . . .	638
7.263SendParam Class Reference . . . . .	639
7.264SendToParam Class Reference . . . . .	640
7.265ShutdownParam Class Reference . . . . .	641
7.266SocketRequestParam Class Reference . . . . .	642
7.267StrCSumPtrLen Struct Reference . . . . .	644
7.268StrPtrLen Struct Reference . . . . .	647
7.269TimeValue Class Reference . . . . .	649
7.270TLSStorageOps Class Reference . . . . .	656
7.271TReadyQueLink Class Reference . . . . .	657
7.272WStrPtrLen Struct Reference . . . . .	658
<b>8 oscl File Documentation . . . . .</b>	<b>660</b>
8.1 oscl_aostatus.h File Reference . . . . .	660
8.2 oscl_assert.h File Reference . . . . .	661
8.3 oscl_base.h File Reference . . . . .	662
8.4 oscl_base_alloc.h File Reference . . . . .	663
8.5 oscl_base_macros.h File Reference . . . . .	664
8.6 oscl_bin_stream.h File Reference . . . . .	665
8.7 oscl_byte_order.h File Reference . . . . .	666
8.8 oscl_defalloc.h File Reference . . . . .	667
8.9 oscl_dll.h File Reference . . . . .	668
8.10 oscl_dns.h File Reference . . . . .	669
8.11 oscl_dns_gethostname.h File Reference . . . . .	670
8.12 oscl_dns_imp.h File Reference . . . . .	671
8.13 oscl_dns_imp_base.h File Reference . . . . .	672
8.14 oscl_dns_imp_pv.h File Reference . . . . .	673
8.15 oscl_dns_method.h File Reference . . . . .	674
8.16 oscl_dns_param.h File Reference . . . . .	675
8.17 oscl_dns_request.h File Reference . . . . .	676
8.18 oscl_dns_tuneables.h File Reference . . . . .	677
8.19 oscl_double_list.h File Reference . . . . .	678

8.20 oscl_errno.h File Reference . . . . .	679
8.21 oscl_error.h File Reference . . . . .	680
8.22 oscl_error_allocator.h File Reference . . . . .	681
8.23 oscl_error_codes.h File Reference . . . . .	682
8.24 oscl_error_imp.h File Reference . . . . .	683
8.25 oscl_error_imp_cppexceptions.h File Reference . . . . .	684
8.26 oscl_error_imp_fatalerror.h File Reference . . . . .	685
8.27 oscl_error_imp_jumps.h File Reference . . . . .	686
8.28 oscl_error_trapcleanup.h File Reference . . . . .	688
8.29 oscl_exception.h File Reference . . . . .	689
8.30 oscl_exclusive_ptr.h File Reference . . . . .	690
8.31 oscl_file_async_read.h File Reference . . . . .	691
8.32 oscl_file_cache.h File Reference . . . . .	692
8.33 oscl_file_dir_utils.h File Reference . . . . .	693
8.34 oscl_file_find.h File Reference . . . . .	695
8.35 oscl_file_handle.h File Reference . . . . .	696
8.36 oscl_file_io.h File Reference . . . . .	697
8.37 oscl_file_manager.h File Reference . . . . .	698
8.38 oscl_file_native.h File Reference . . . . .	699
8.39 oscl_file_server.h File Reference . . . . .	700
8.40 oscl_file_stats.h File Reference . . . . .	701
8.41 oscl_file_types.h File Reference . . . . .	702
8.42 oscl_heapbase.h File Reference . . . . .	703
8.43 oscl_init.h File Reference . . . . .	704
8.44 oscl_int64_utils.h File Reference . . . . .	705
8.45 oscl_ip_socket.h File Reference . . . . .	706
8.46 oscl_linked_list.h File Reference . . . . .	707
8.47 oscl_lock_base.h File Reference . . . . .	708
8.48 oscl_map.h File Reference . . . . .	709
8.49 oscl_math.h File Reference . . . . .	710
8.50 oscl_media_data.h File Reference . . . . .	711
8.51 oscl_media_status.h File Reference . . . . .	712
8.52 oscl_mem.h File Reference . . . . .	713
8.53 oscl_mem_audit.h File Reference . . . . .	716
8.54 oscl_mem_audit_internals.h File Reference . . . . .	718
8.55 oscl_mem_auto_ptr.h File Reference . . . . .	719

8.56 <a href="#">oscl_mem_basic_functions.h File Reference</a>	720
8.57 <a href="#">oscl_mem_inst.h File Reference</a>	721
8.58 <a href="#">oscl_mem_mempool.h File Reference</a>	722
8.59 <a href="#">oscl_mutex.h File Reference</a>	723
8.60 <a href="#">oscl_namestring.h File Reference</a>	724
8.61 <a href="#">oscl_opaque_type.h File Reference</a>	725
8.62 <a href="#">oscl_priqueue.h File Reference</a>	726
8.63 <a href="#">oscl_procstatus.h File Reference</a>	727
8.64 <a href="#">oscl_queue.h File Reference</a>	728
8.65 <a href="#">oscl_rand.h File Reference</a>	729
8.66 <a href="#">oscl_refcounter.h File Reference</a>	730
8.67 <a href="#">oscl_refcounter_memfrag.h File Reference</a>	731
8.68 <a href="#">oscl_registry_access_client.h File Reference</a>	732
8.69 <a href="#">oscl_registry_client.h File Reference</a>	733
8.70 <a href="#">oscl_registry_client_impl.h File Reference</a>	734
8.71 <a href="#">oscl_registry_serv_impl.h File Reference</a>	735
8.72 <a href="#">oscl_registry_serv_impl_global.h File Reference</a>	736
8.73 <a href="#">oscl_registry_serv_impl_tls.h File Reference</a>	737
8.74 <a href="#">oscl_registry_types.h File Reference</a>	738
8.75 <a href="#">oscl_scheduler.h File Reference</a>	739
8.76 <a href="#">oscl_scheduler_ao.h File Reference</a>	740
8.77 <a href="#">oscl_scheduler_aobase.h File Reference</a>	741
8.78 <a href="#">oscl_scheduler_readyq.h File Reference</a>	742
8.79 <a href="#">oscl_scheduler_threadcontext.h File Reference</a>	743
8.80 <a href="#">oscl_scheduler_tuneables.h File Reference</a>	744
8.81 <a href="#">oscl_scheduler_types.h File Reference</a>	745
8.82 <a href="#">oscl_semaphore.h File Reference</a>	746
8.83 <a href="#">oscl_shared_ptr.h File Reference</a>	747
8.84 <a href="#">oscl_singleton.h File Reference</a>	748
8.85 <a href="#">oscl_snprintf.h File Reference</a>	750
8.86 <a href="#">oscl_socket.h File Reference</a>	751
8.87 <a href="#">oscl_socket_accept.h File Reference</a>	752
8.88 <a href="#">oscl_socket_bind.h File Reference</a>	753
8.89 <a href="#">oscl_socket_connect.h File Reference</a>	754
8.90 <a href="#">oscl_socket_imp.h File Reference</a>	755
8.91 <a href="#">oscl_socket_imp_base.h File Reference</a>	756

8.92 oscl_socket_imp_pv.h File Reference . . . . .	757
8.93 oscl_socket_listen.h File Reference . . . . .	758
8.94 oscl_socket_method.h File Reference . . . . .	759
8.95 oscl_socket_recv.h File Reference . . . . .	760
8.96 oscl_socket_recv_from.h File Reference . . . . .	761
8.97 oscl_socket_request.h File Reference . . . . .	762
8.98 oscl_socket_send.h File Reference . . . . .	763
8.99 oscl_socket_send_to.h File Reference . . . . .	764
8.100oscl_socket_serv_imp.h File Reference . . . . .	765
8.101oscl_socket_serv_imp_base.h File Reference . . . . .	766
8.102oscl_socket_serv_imp_pv.h File Reference . . . . .	767
8.103oscl_socket_serv_imp_reqlist.h File Reference . . . . .	768
8.104oscl_socket_shutdown.h File Reference . . . . .	769
8.105oscl_socket_stats.h File Reference . . . . .	770
8.106oscl_socket_tuneables.h File Reference . . . . .	772
8.107oscl_socket_types.h File Reference . . . . .	774
8.108oscl_stdstring.h File Reference . . . . .	776
8.109oscl_str_ptr_len.h File Reference . . . . .	778
8.110oscl_string.h File Reference . . . . .	779
8.111oscl_string_containers.h File Reference . . . . .	780
8.112oscl_string_rep.h File Reference . . . . .	781
8.113oscl_string_uri.h File Reference . . . . .	782
8.114oscl_string_utf8.h File Reference . . . . .	783
8.115oscl_string_utils.h File Reference . . . . .	784
8.116oscl_string_xml.h File Reference . . . . .	785
8.117oscl_tagtree.h File Reference . . . . .	786
8.118oscl_tcp_socket.h File Reference . . . . .	787
8.119oscl_thread.h File Reference . . . . .	788
8.120oscl_tickcount.h File Reference . . . . .	790
8.121oscl_time.h File Reference . . . . .	791
8.122oscl_timer.h File Reference . . . . .	793
8.123oscl_tls.h File Reference . . . . .	794
8.124oscl_tree.h File Reference . . . . .	795
8.125oscl_types.h File Reference . . . . .	796
8.126oscl_udp_socket.h File Reference . . . . .	797
8.127oscl_utf8conv.h File Reference . . . . .	798

8.128oscl_uuid.h File Reference . . . . .	799
8.129oscl_uuid_utils.h File Reference . . . . .	800
8.130oscl_vector.h File Reference . . . . .	801
8.131osclconfig.h File Reference . . . . .	802
8.132osclconfig_ansi_memory.h File Reference . . . . .	804
8.133osclconfig_check.h File Reference . . . . .	805
8.134osclconfig_compiler_warnings.h File Reference . . . . .	806
8.135osclconfig_error.h File Reference . . . . .	807
8.136osclconfig_error_check.h File Reference . . . . .	808
8.137osclconfig_global_new_delete.h File Reference . . . . .	809
8.138osclconfig_global_placement_new.h File Reference . . . . .	810
8.139osclconfig_io.h File Reference . . . . .	811
8.140osclconfig_io_check.h File Reference . . . . .	822
8.141osclconfig_ix86.h File Reference . . . . .	823
8.142osclconfig_lib.h File Reference . . . . .	824
8.143osclconfig_lib_check.h File Reference . . . . .	825
8.144osclconfig_limits_typedefs.h File Reference . . . . .	826
8.145osclconfig_memory.h File Reference . . . . .	827
8.146osclconfig_memory_check.h File Reference . . . . .	828
8.147osclconfig_no_os.h File Reference . . . . .	829
8.148osclconfig_proc.h File Reference . . . . .	830
8.149osclconfig_proc_check.h File Reference . . . . .	831
8.150osclconfig_proc_unix_android.h File Reference . . . . .	833
8.151osclconfig_proc_unix_common.h File Reference . . . . .	835
8.152osclconfig_time.h File Reference . . . . .	837
8.153osclconfig_time_check.h File Reference . . . . .	838
8.154osclconfig_unix_android.h File Reference . . . . .	839
8.155osclconfig_unix_common.h File Reference . . . . .	843
8.156osclconfig_util.h File Reference . . . . .	847
8.157osclconfig_util_check.h File Reference . . . . .	848
8.158pvlogger.h File Reference . . . . .	849
8.159pvlogger_accessories.h File Reference . . . . .	857
8.160pvlogger_c.h File Reference . . . . .	858
8.161pvlogger_registry.h File Reference . . . . .	860
<b>9 oscl Page Documentation</b>	<b>861</b>
9.1 Todo List . . . . .	861

# Chapter 1

## oscl Module Index

### 1.1 oscl Modules

Here is a list of all modules:

OSCL config . . . . .	21
OSCL Base . . . . .	25
OSCL Memory . . . . .	47
OSCL Util . . . . .	63
OSCL Error . . . . .	85
OSCL IO . . . . .	95
OSCL Proc . . . . .	103
OSCL Init . . . . .	107

# Chapter 2

## oscl Hierarchical Index

### 2.1 oscl Class Hierarchy

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

_OscIHeapBase . . . . .	110
HeapBase . . . . .	137
Oscl_File . . . . .	179
OSCL_String . . . . .	260
OSCL_FastString . . . . .	175
OSCL_HeapString< Alloc > . . . . .	196
OSCL_HeapStringA . . . . .	198
OSCL_StackString< MaxBufSize > . . . . .	257
OsclActiveObject . . . . .	310
OsclAsyncFile . . . . .	316
OsclDNSRequestAO . . . . .	363
OsclGetHostNameRequest . . . . .	414
OsclSocketRequestAO . . . . .	550
OsclAcceptRequest . . . . .	309
OsclBindRequest . . . . .	323
OsclConnectRequest . . . . .	349
OsclListenRequest . . . . .	423
OsclRecvFromRequest . . . . .	488
OsclRecvRequest . . . . .	491
OsclSendRequest . . . . .	524
OsclSendToRequest . . . . .	526
OsclShutdownRequest . . . . .	531
PVSchedulerStopper . . . . .	631
OsclAsyncFileBuffer . . . . .	319
OsclBuf . . . . .	340
OsclDNS . . . . .	351
OsclFileCache . . . . .	401
OsclNativeFile . . . . .	462
OsclPtr . . . . .	477
OsclPtrC . . . . .	479
OsclRegistryClient . . . . .	509
OsclSocketServ . . . . .	554
OsclTCPSocket . . . . .	565

OsclTimerObject . . . . .	586
CallbackTimer< Alloc > . . . . .	124
OsclDNSMethod . . . . .	358
OsclGetHostByNameMethod . . . . .	413
OsclSocketMethod . . . . .	545
OsclAcceptMethod . . . . .	308
OsclBindMethod . . . . .	322
OsclConnectMethod . . . . .	348
OsclListenMethod . . . . .	422
OsclRecvFromMethod . . . . .	486
OsclRecvMethod . . . . .	490
OsclSendMethod . . . . .	523
OsclSendToMethod . . . . .	525
OsclShutdownMethod . . . . .	530
OsclSocketServI . . . . .	556
OsclUDPSocket . . . . .	601
OsclExecSchedulerBase . . . . .	391
OsclExecScheduler . . . . .	389
allocator . . . . .	113
BufferMgr . . . . .	118
BufferState . . . . .	119
BufFragGroup< ChainClass, max_frags > . . . . .	120
MediaData< ChainClass, max_frags, local_bufsize > . . . . .	142
BufFragStatusClass . . . . .	123
MediaStatusClass . . . . .	145
CallbackTimerObserver . . . . .	126
OsclTimer< Alloc > . . . . .	582
CFastRep . . . . .	127
CHheapRep . . . . .	129
CStackRep . . . . .	132
DNSRequestParam . . . . .	133
GetHostByNameParam . . . . .	135
internalLeave . . . . .	139
LinkedListElement< LLClass > . . . . .	140
MemAllocator< T > . . . . .	146
MM_AllocBlockFence . . . . .	147
MM_AllocBlockHdr . . . . .	148
MM_AllocInfo . . . . .	149
MM_AllocNode . . . . .	151
MM_AllocQueryInfo . . . . .	152
MM_Audit_Imp . . . . .	153
MM_AuditOverheadStats . . . . .	161
MM_FailInsertParam . . . . .	162
MM_Stats_CB . . . . .	163
MM_Stats_t . . . . .	164
NTPTime . . . . .	166
Oscl_Alloc . . . . .	170
Oscl_DefAlloc . . . . .	172
_OsclBasicAllocator . . . . .	108
OsclAllocDestructDealloc . . . . .	314
OsclMemAllocDestructDealloc< T > . . . . .	427
OsclMemBasicAllocDestructDealloc< T > . . . . .	440

OsclMemAllocator . . . . .	426
OsclMemBasicAllocator . . . . .	439
OsclMemPoolFixedChunkAllocator . . . . .	443
OsclMemPoolResizableAllocator . . . . .	448
OsclReadyAlloc . . . . .	482
Oscl_Dealloc . . . . .	171
Oscl_DefAlloc . . . . .	172
Oscl_File::OsclCacheObserver . . . . .	187
Oscl_File::OsclFixedCacheParam . . . . .	188
Oscl_FileFind . . . . .	189
Oscl_FileServer . . . . .	193
oscl_fsstat . . . . .	195
Oscl_Int64_Utils . . . . .	203
Oscl_Less< T > . . . . .	205
Oscl_Linked_List_Base . . . . .	211
Oscl_Linked_List< LLClass, Alloc > . . . . .	206
Oscl_Map< Key, T, Alloc, Compare > . . . . .	216
Oscl_Map< Key, T, Alloc, Compare >::value_compare . . . . .	223
Oscl_MTLinked_List< LLClass, Alloc, TheLock > . . . . .	225
Oscl_Opaque_Type_Alloc . . . . .	229
Oscl_Queue< T, Alloc > . . . . .	236
Oscl_Vector< T, Alloc > . . . . .	285
Oscl_Vector< TOsclReady, OsclReadyAlloc > . . . . .	285
Oscl_Opaque_Type_Alloc_LL . . . . .	231
Oscl_Linked_List< LLClass, Alloc > . . . . .	206
Oscl_Opaque_Type_Compare . . . . .	233
OsclPriorityQueue< Qelem, Alloc, Container, Compare > . . . . .	470
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclReadyCompare > . . . . .	470
OsclReadyQ . . . . .	484
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclTimerCompare > . . . . .	470
OsclTimerQ . . . . .	591
Oscl_Pair< T1, T2 > . . . . .	235
Oscl_Queue_Base . . . . .	239
Oscl_Queue< T, Alloc > . . . . .	236
Oscl_Rb_Tree_Base . . . . .	246
Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > . . . . .	242
Oscl_Rb_Tree_Const_Iterator< Value > . . . . .	247
Oscl_Rb_Tree_Iterator< Value > . . . . .	250
Oscl_Rb_Tree_Node_Base . . . . .	254
Oscl_Rb_Tree_Node< Value > . . . . .	253
Oscl_Select1st< V, U > . . . . .	256
oscl_stat_buf . . . . .	259
Oscl_Tag_Base . . . . .	267
Oscl_Tag< Alloc > . . . . .	265
Oscl_TagTree< T, Alloc > . . . . .	269
Oscl_TagTree< T, Alloc >::const_iterator . . . . .	273
Oscl_TagTree< T, Alloc >::iterator . . . . .	276
Oscl_TagTree< T, Alloc >::Node . . . . .	279
Oscl_TAlloc< T, Alloc >::rebind< U, V > . . . . .	284

Oscl_Vector_Base . . . . .	290
Oscl_Vector< T, Alloc > . . . . .	285
Oscl_Vector< TOsclReady, OsclReadyAlloc > . . . . .	285
OSCL_wString . . . . .	304
OSCL_wFastString . . . . .	294
OSCL_wHeapString< Alloc > . . . . .	297
OSCL_wHeapStringA . . . . .	299
OSCL_wStackString< MaxBufSize > . . . . .	302
OsclAOStatus . . . . .	315
OsclAuditCB . . . . .	321
OsclBinStream . . . . .	336
OsclBinIStream . . . . .	324
OsclBinIStreamBigEndian . . . . .	326
OsclBinIStreamLittleEndian . . . . .	329
OsclBinOStream . . . . .	331
OsclBinOStreamBigEndian . . . . .	332
OsclBinOStreamLittleEndian . . . . .	334
OsclCompareLess< T > . . . . .	342
OsclComponentRegistry . . . . .	343
OsclComponentRegistryData . . . . .	345
OsclComponentRegistryElement . . . . .	346
OsclDestructDealloc . . . . .	350
Oscl_TAlloc< T, Alloc > . . . . .	281
OsclAllocDestructDealloc . . . . .	314
OsclDNSIBase . . . . .	355
OsclDNSI . . . . .	353
OsclDNSObserver . . . . .	361
OsclDNSRequest . . . . .	362
OsclDoubleLink . . . . .	366
OsclPriorityLink . . . . .	468
OsclDoubleListBase . . . . .	368
OsclDoubleList< T > . . . . .	367
OsclPriorityList< T > . . . . .	469
OsclDoubleRunner< T > . . . . .	370
OsclError . . . . .	372
OsclErrorAllocator . . . . .	374
OsclErrorTrap . . . . .	376
OsclErrorTrapImp . . . . .	377
OsclException< LeaveCode > . . . . .	379
OsclExclusiveArrayPtr< T > . . . . .	380
OsclExclusivePtr< T > . . . . .	383
OsclExclusivePtrA< T, Alloc > . . . . .	386
OsclExecSchedulerCommonBase . . . . .	392
OsclExecScheduler . . . . .	389
OsclFileCacheBuffer . . . . .	403
OsclFileHandle . . . . .	405
OsclFileManager . . . . .	406
OsclFileStats . . . . .	411
OsclFileStatsItem . . . . .	412
OsclInit . . . . .	415
OsclInteger64Transport . . . . .	416

OsclIpMReq . . . . .	417
OsclIPSocketI . . . . .	418
OsclTCPSocketI . . . . .	572
OsclUDPSocketI . . . . .	607
OsclJump . . . . .	421
OsclLockBase . . . . .	424
OsclMutex . . . . .	459
OsclNullLock . . . . .	467
OsclThreadLock . . . . .	579
OsclMem . . . . .	425
OsclMemAudit . . . . .	429
OSCLMemAutoPtr< T, _Allocator > . . . . .	435
OsclMemGlobalAuditObject . . . . .	441
OsclMemoryFragment . . . . .	442
BufferFragment . . . . .	117
OsclMemPoolFixedChunkAllocatorObserver . . . . .	447
OsclMemPoolResizableAllocator::MemPoolBlockInfo . . . . .	454
OsclMemPoolResizableAllocator::MemPoolBufferInfo . . . . .	455
OsclMemPoolResizableAllocatorMemoryObserver . . . . .	456
OsclMemPoolResizableAllocatorObserver . . . . .	457
OsclMemStatsNode . . . . .	458
OsclNameString< __len > . . . . .	461
OsclNativeFileParams . . . . .	465
OsclNetworkAddress . . . . .	466
OsclPriorityQueueBase . . . . .	474
OsclPriorityQueue< Qelem, Alloc, Container, Compare > . . . . .	470
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclReadyCompare > . . . . .	470
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclTimerCompare > . . . . .	470
OsclProcStatus . . . . .	475
OsclRand . . . . .	481
OsclReadyCompare . . . . .	483
OsclRefCounter . . . . .	492
Oscl_DefAllocWithRefCounter< DefAlloc > . . . . .	173
OsclRefCounterDA . . . . .	494
OsclRefCounterMTDA< LockType > . . . . .	498
OsclRefCounterMTSA< DeallocType, LockType > . . . . .	500
OsclRefCounterSA< DeallocType > . . . . .	502
OsclRefCounterMemFrag . . . . .	496
OsclRegistryAccessClient . . . . .	504
OsclRegistryAccessElement . . . . .	508
OsclRegistryClientImpl . . . . .	511
OsclRegistryAccessClientImpl . . . . .	506
OsclRegistryServTlsImpl . . . . .	514
OsclRegistryAccessClientTlsImpl . . . . .	507
OsclRegistryClientTlsImpl . . . . .	513
OsclScheduler . . . . .	516
OsclSchedulerObserver . . . . .	517
OsclScopedLock< LockClass > . . . . .	518
OsclSelect . . . . .	519
OsclSemaphore . . . . .	521

OsclSharedPtr< TheClass > . . . . .	527
OsclSingleton< T, ID, Registry > . . . . .	532
OsclSingletonRegistry . . . . .	534
OsclSocketIBase . . . . .	540
OsclSocketI . . . . .	535
OsclSocketObserver . . . . .	548
OsclSocketRequest . . . . .	549
OsclSocketServIBase . . . . .	558
OsclSocketServI . . . . .	556
OsclSocketServRequestList . . . . .	560
OsclSocketServRequestQElem . . . . .	562
OsclSocketTOS . . . . .	563
OsclThread . . . . .	575
OsclTickCount . . . . .	580
OsclTimerCompare . . . . .	585
OsclTimerObserver . . . . .	590
OsclTLS< T, ID, Registry > . . . . .	592
OsclTLSE< T, ID, Registry > . . . . .	594
OsclTLSRegistry . . . . .	596
OsclTLSRegistryEx . . . . .	597
OsclTrapItem . . . . .	598
OsclTrapStack . . . . .	599
OsclTrapStackItem . . . . .	600
OsclUuid . . . . .	610
PVActiveBase . . . . .	612
OsclActiveObject . . . . .	310
OsclTimerObject . . . . .	586
PVActiveStats . . . . .	616
PVLogger . . . . .	617
PVLoggerAppender . . . . .	623
PVLoggerFilter . . . . .	624
AllPassFilter . . . . .	114
PVLoggerLayout . . . . .	626
PVLoggerRegistry . . . . .	628
PVSockBufRecv . . . . .	632
PVSockBufSend . . . . .	633
PVThreadContext . . . . .	634
SocketRequestParam . . . . .	642
AcceptParam . . . . .	112
BindParam . . . . .	116
ConnectParam . . . . .	131
ListenParam . . . . .	141
RecvFromParam . . . . .	636
RecvParam . . . . .	638
SendParam . . . . .	639
SendToParam . . . . .	640
ShutdownParam . . . . .	641
StrPtrLen . . . . .	647
StrCSumPtrLen . . . . .	644
TimeValue . . . . .	649
TLSStorageOps . . . . .	656
TReadyQueLink . . . . .	657

WStrPtrLen . . . . . 658

# Chapter 3

## oscl Data Structure Index

### 3.1 oscl Data Structures

Here are the data structures with brief descriptions:

_OsclBasicAllocator . . . . .	108
_OsclHeapBase . . . . .	110
AcceptParam . . . . .	112
allocator . . . . .	113
AllPassFilter . . . . .	114
BindParam . . . . .	116
BufferFragment . . . . .	117
BufferMgr . . . . .	118
BufferState . . . . .	119
BufFragGroup< ChainClass, max_frags >	120
BufFragStatusClass . . . . .	123
CallbackTimer< Alloc > . . . . .	124
CallbackTimerObserver . . . . .	126
CFastRep . . . . .	127
CHheapRep . . . . .	129
ConnectParam . . . . .	131
CStackRep . . . . .	132
DNSRequestParam . . . . .	133
GetHostNameParam . . . . .	135
HeapBase . . . . .	137
internalLeave . . . . .	139
LinkedListElement< LLClass > . . . . .	140
ListenParam . . . . .	141
MediaData< ChainClass, max_frags, local_bufsize >	142
MediaStatusClass . . . . .	145
MemAllocator< T > . . . . .	146
MM_AllocBlockFence . . . . .	147
MM_AllocBlockHdr . . . . .	148
MM_AllocInfo . . . . .	149
MM_AllocNode . . . . .	151
MM_AllocQueryInfo . . . . .	152
MM_Audit_Imp . . . . .	153
MM_AuditOverheadStats . . . . .	161

MM_FailInsertParam . . . . .	162
MM_Stats_CB . . . . .	163
MM_Stats_t . . . . .	164
NTPTime (Time value as the number of seconds since 0h (UTC) Jan. 1, 1900) . . . . .	166
OscI_Alloc . . . . .	170
OscI_Dealloc . . . . .	171
OscI_DefAlloc . . . . .	172
OscI_DefAllocWithRefCounter< DefAlloc > . . . . .	173
OSCL_FastString . . . . .	175
OscI_File . . . . .	179
OscI_File::OscICacheObserver . . . . .	187
OscI_File::OscIFixedCacheParam . . . . .	188
OscI_FileFind . . . . .	189
OscI_FileServer . . . . .	193
oscl_fstat . . . . .	195
OSCL_HeapString< Alloc > . . . . .	196
OSCL_HeapStringA . . . . .	198
OscI_Int64_Utils (Wrapper for commonly used int64/uint64 operations) . . . . .	203
OscI_Less< T > . . . . .	205
OscI_Linked_List< LLClass, Alloc > . . . . .	206
OscI_Linked_List_Base . . . . .	211
OscI_Map< Key, T, Alloc, Compare > . . . . .	216
OscI_Map< Key, T, Alloc, Compare >::value_compare . . . . .	223
OscI_MTLinked_List< LLClass, Alloc, TheLock > . . . . .	225
OscI_Opaque_Type_Alloc . . . . .	229
OscI_Opaque_Type_Alloc_LL . . . . .	231
OscI_Opaque_Type_Compare . . . . .	233
OscI_Pair< T1, T2 > . . . . .	235
OscI_Queue< T, Alloc > . . . . .	236
OscI_Queue_Base . . . . .	239
OscI_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > . . . . .	242
OscI_Rb_Tree_Base . . . . .	246
OscI_Rb_Tree_Const_Iterator< Value > . . . . .	247
OscI_Rb_Tree_Iterator< Value > . . . . .	250
OscI_Rb_Tree_Node< Value > . . . . .	253
OscI_Rb_Tree_Node_Base . . . . .	254
OscI_Select1st< V, U > . . . . .	256
OSCL_StackString< MaxBufSize > . . . . .	257
oscl_stat_buf . . . . .	259
OSCL_String . . . . .	260
OscI_Tag< Alloc > . . . . .	265
OscI_Tag_Base . . . . .	267
OscI_TagTree< T, Alloc > . . . . .	269
OscI_TagTree< T, Alloc >::const_iterator . . . . .	273
OscI_TagTree< T, Alloc >::iterator . . . . .	276
OscI_TagTree< T, Alloc >::Node . . . . .	279
OscI_TAlloc< T, Alloc > . . . . .	281
OscI_TAlloc< T, Alloc >::rebind< U, V > . . . . .	284
OscI_Vector< T, Alloc > . . . . .	285
OscI_Vector_Base . . . . .	290
OSCL_wFastString . . . . .	294
OSCL_wHeapString< Alloc > . . . . .	297
OSCL_wHeapStringA . . . . .	299
OSCL_wStackString< MaxBufSize > . . . . .	302

<b>OSCL_wString</b>	304
<b>OsclAcceptMethod</b>	308
<b>OsclAcceptRequest</b>	309
<b>OsclActiveObject</b>	310
<b>OsclAllocDestructDealloc</b>	314
<b>OsclAOStatus</b>	315
<b>OsclAsyncFile</b>	316
<b>OsclAsyncFileBuffer</b>	319
<b>OsclAuditCB</b>	321
<b>OsclBindMethod</b>	322
<b>OsclBindRequest</b>	323
<b>OsclBinIStream</b>	324
<b>OsclBinIStreamBigEndian</b>	326
<b>OsclBinIStreamLittleEndian</b>	329
<b>OsclBinOStream</b> (Class OsclBinOStream implements the basic stream functions for an output stream)	331
<b>OsclBinOStreamBigEndian</b> (Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering)	332
<b>OsclBinOStreamLittleEndian</b> (Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering)	334
<b>OsclBinStream</b>	336
<b>OsclBuf</b>	340
<b>OsclCompareLess&lt; T &gt;</b>	342
<b>OsclComponentRegistry</b>	343
<b>OsclComponentRegistryData</b>	345
<b>OsclComponentRegistryElement</b>	346
<b>OsclConnectMethod</b>	348
<b>OsclConnectRequest</b>	349
<b>OsclDestructDealloc</b>	350
<b>OsclDNS</b>	351
<b>OsclDNSI</b>	353
<b>OsclDNSIBase</b>	355
<b>OsclDNSMethod</b>	358
<b>OsclDNSObserver</b>	361
<b>OsclDNSRequest</b>	362
<b>OsclDNSRequestAO</b>	363
<b>OsclDoubleLink</b>	366
<b>OsclDoubleList&lt; T &gt;</b>	367
<b>OsclDoubleListBase</b>	368
<b>OsclDoubleRunner&lt; T &gt;</b>	370
<b>OsclError</b>	372
<b>OsclErrorAllocator</b> (This class provides static methods to invoke the user defined memory allocation routines)	374
<b>OsclErrorTrap</b>	376
<b>OsclErrorTrapImp</b>	377
<b>OsclException&lt; LeaveCode &gt;</b> (Oscl_exception.h contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from)	379
<b>OsclExclusiveArrayPtr&lt; T &gt;</b> (Template class that defines an array pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory)	380
<b>OsclExclusivePtr&lt; T &gt;</b> (Template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory)	383

<b>OsclExclusivePtrA&lt; T, Alloc &gt;</b> (Template class that defines any pointer like object intended to be assigned an address obtained (directly or or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory) . . . . .	386
<b>OsclExecScheduler</b> . . . . .	389
<b>OsclExecSchedulerBase</b> . . . . .	391
<b>OsclExecSchedulerCommonBase</b> . . . . .	392
<b>OsclFileCache</b> . . . . .	401
<b>OsclFileCacheBuffer</b> . . . . .	403
<b>OsclFileHandle</b> . . . . .	405
<b>OsclFileManager</b> . . . . .	406
<b>OsclFileStats</b> . . . . .	411
<b>OsclFileStatsItem</b> . . . . .	412
<b>OsclGetHostNameMethod</b> . . . . .	413
<b>OsclGetHostNameRequest</b> . . . . .	414
<b>OsclInit</b> . . . . .	415
<b>OsclInteger64Transport</b> . . . . .	416
<b>OsclIpMReq</b> . . . . .	417
<b>OsclIPSocketI</b> . . . . .	418
<b>OsclJump</b> . . . . .	421
<b>OsclListenMethod</b> . . . . .	422
<b>OsclListenRequest</b> . . . . .	423
<b>OsclLockBase</b> . . . . .	424
<b>OsclMem</b> . . . . .	425
<b>OsclMemAllocator</b> . . . . .	426
<b>OsclMemAllocDestructDealloc&lt; T &gt;</b>	427
<b>OsclMemAudit</b> . . . . .	429
<b>OSCLMemAutoPtr&lt; T, _Allocator &gt;</b> (The oscl_auto_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the oscl_auto_ptr expires, its destructor uses delete to free the memory) . . . . .	435
<b>OsclMemBasicAllocator</b> . . . . .	439
<b>OsclMemBasicAllocDestructDealloc&lt; T &gt;</b>	440
<b>OsclMemGlobalAuditObject</b> . . . . .	441
<b>OsclMemoryFragment</b> . . . . .	442
<b>OsclMemPoolFixedChunkAllocator</b> . . . . .	443
<b>OsclMemPoolFixedChunkAllocatorObserver</b> . . . . .	447
<b>OsclMemPoolResizableAllocator</b> . . . . .	448
<b>OsclMemPoolResizableAllocator::MemPoolBlockInfo</b> . . . . .	454
<b>OsclMemPoolResizableAllocator::MemPoolBufferInfo</b> . . . . .	455
<b>OsclMemPoolResizableAllocatorMemoryObserver</b> . . . . .	456
<b>OsclMemPoolResizableAllocatorObserver</b> . . . . .	457
<b>OsclMemStatsNode</b> . . . . .	458
<b>OsclMutex</b> . . . . .	459
<b>OsclNameString&lt; __len &gt;</b>	461
<b>OsclNativeFile</b> . . . . .	462
<b>OsclNativeFileParams</b> . . . . .	465
<b>OsclNetworkAddress</b> . . . . .	466
<b>OsclNullLock</b> . . . . .	467
<b>OsclPriorityLink</b> . . . . .	468
<b>OsclPriorityList&lt; T &gt;</b> . . . . .	469
<b>OsclPriorityQueue&lt; Qelem, Alloc, Container, Compare &gt;</b> . . . . .	470
<b>OsclPriorityQueueBase</b> . . . . .	474
<b>OsclProcStatus</b> . . . . .	475
<b>OsclPtr</b> . . . . .	477

OsclPtrC . . . . .	479
OsclRand . . . . .	481
OsclReadyAlloc . . . . .	482
OsclReadyCompare . . . . .	483
OsclReadyQ . . . . .	484
OsclRecvFromMethod . . . . .	486
OsclRecvFromRequest . . . . .	488
OsclRecvMethod . . . . .	490
OsclRecvRequest . . . . .	491
OsclRefCounter . . . . .	492
OsclRefCounterDA . . . . .	494
OsclRefCounterMemFrag . . . . .	496
OsclRefCounterMTDA< LockType > . . . . .	498
OsclRefCounterMTSA< DeallocType, LockType > . . . . .	500
OsclRefCounterSA< DeallocType > . . . . .	502
OsclRegistryAccessClient . . . . .	504
OsclRegistryAccessClientImpl . . . . .	506
OsclRegistryAccessClientTlsImpl . . . . .	507
OsclRegistryAccessElement . . . . .	508
OsclRegistryClient . . . . .	509
OsclRegistryClientImpl . . . . .	511
OsclRegistryClientTlsImpl . . . . .	513
OsclRegistryServTlsImpl . . . . .	514
OsclScheduler . . . . .	516
OsclSchedulerObserver . . . . .	517
OsclScopedLock< LockClass > (Template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope) . . . . .	518
OsclSelect . . . . .	519
OsclSemaphore . . . . .	521
OsclSendMethod . . . . .	523
OsclSendRequest . . . . .	524
OsclSendToMethod . . . . .	525
OsclSendToRequest . . . . .	526
OsclSharedPtr< TheClass > (A parameterized smart pointer class) . . . . .	527
OsclShutdownMethod . . . . .	530
OsclShutdownRequest . . . . .	531
OsclSingleton< T, ID, Registry > . . . . .	532
OsclSingletonRegistry . . . . .	534
OsclSocketI . . . . .	535
OsclSocketIBase . . . . .	540
OsclSocketMethod . . . . .	545
OsclSocketObserver . . . . .	548
OsclSocketRequest . . . . .	549
OsclSocketRequestAO . . . . .	550
OsclSocketServ . . . . .	554
OsclSocketServI . . . . .	556
OsclSocketServIBase . . . . .	558
OsclSocketServRequestList . . . . .	560
OsclSocketServRequestQElem . . . . .	562
OsclSocketTOS . . . . .	563
OsclTCPSocket . . . . .	565
OsclTCPSocketI . . . . .	572
OsclThread . . . . .	575

OsclThreadLock	579
OsclTickCount	580
OsclTimer< Alloc >	582
OsclTimerCompare	585
OsclTimerObject	586
OsclTimerObserver	590
OsclTimerQ	591
OsclTLS< T, ID, Registry >	592
OsclTLSE< T, ID, Registry >	594
OsclTLSRegistry	596
OsclTLSRegistryEx	597
OsclTrapItem	598
OsclTrapStack	599
OsclTrapStackItem	600
OsclUDPSocket	601
OsclUDPSocketI	607
OsclUuid	610
PVActiveBase	612
PVActiveStats	616
PVLogger	617
PVLoggerAppender	623
PVLoggerFilter	624
PVLoggerLayout	626
PVLoggerRegistry	628
PVSchedulerStopper	631
PVSockBufRecv	632
PVSockBufSend	633
PVThreadContext	634
RecvFromParam	636
RecvParam	638
SendParam	639
SendToParam	640
ShutdownParam	641
SocketRequestParam	642
StrCSumPtrLen (Same as StrPtrLen, but includes checksum field and method to speed up querying)	644
StrPtrLen (This data structure encapsulates a set of functions used to perform)	647
TimeValue (Time value in a format native to the system)	649
TLSStorageOps	656
TReadyQueLink	657
WStrPtrLen (This data structure encapsulates a set of functions used to perform)	658

# Chapter 4

## oscl File Index

### 4.1 oscl File List

Here is a list of all files with brief descriptions:

<code>oscl_aostatus.h</code> (Some basic types used with active objects) . . . . .	660
<code>oscl_assert.h</code> (The file <code>oscl_assert.h</code> provides an OSCL_ASSERT macro to document assumptions and test them during development) . . . . .	661
<code>oscl_base.h</code> (The file <code>oscl_base.h</code> is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros) . . . . .	662
<code>oscl_base_alloc.h</code> (A basic allocator that does not rely on other modules) . . . . .	663
<code>oscl_base_macros.h</code> (This file defines common macros and constants for basic compilation support)	664
<code>oscl_bin_stream.h</code> (Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order) . . . . .	665
<code>oscl_byte_order.h</code> (This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders)) . . . . .	666
<code>oscl_defalloc.h</code> (The file defines simple default memory allocator classes. These allocators are used by the <code>Oscl_Vector</code> and <code>Oscl_Map</code> class, etc) . . . . .	667
<code>oscl_dll.h</code> (Defines a DLL entry point) . . . . .	668
<code>oscl_dns.h</code> (The file <code>oscl_socket.h</code> defines the OSCL DNS APIs) . . . . .	669
<code>oscl_dns_gethostbyname.h</code> . . . . .	670
<code>oscl_dns_imp.h</code> . . . . .	671
<code>oscl_dns_imp_base.h</code> . . . . .	672
<code>oscl_dns_imp_pv.h</code> . . . . .	673
<code>oscl_dns_method.h</code> . . . . .	674
<code>oscl_dns_param.h</code> . . . . .	675
<code>oscl_dns_request.h</code> . . . . .	676
<code>oscl_dns_tuneables.h</code> . . . . .	677
<code>oscl_double_list.h</code> (Internal use types for scheduler) . . . . .	678
<code>oscl_errno.h</code> (Defines functions to access additional information on errors where supported through an errno or similar service) . . . . .	679
<code>oscl_error.h</code> (OSCL Error trap and cleanup include file) . . . . .	680
<code>oscl_error_allocator.h</code> (Defines a memory allocation class used by the oscl error layer) . . . . .	681
<code>oscl_error_codes.h</code> (Defines basic error and leave codes) . . . . .	682
<code>oscl_error_imp.h</code> (Internal error implementation support) . . . . .	683
<code>oscl_error_imp_cppexceptions.h</code> (Implementation File for Leave using C++ exceptions) . . . . .	684
<code>oscl_error_imp_fatalerror.h</code> (Implementation File for Leave using system fatal error) . . . . .	685
<code>oscl_error_imp_jumps.h</code> (Implemenation of using Setjmp / Longjmp) . . . . .	686

<code>oscl_error_trapcleanup.h</code> (OSCL Error trap and cleanup implementation include file) . . . . .	688
<code>oscl_exception.h</code> (Contains all the exception handling macros and classes) . . . . .	689
<code>oscl_exclusive_ptr.h</code> (This file defines the <code>OsclExclusivePtr</code> template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error) . . . . .	690
<code>oscl_file_async_read.h</code> . . . . .	691
<code>oscl_file_cache.h</code> (The file <code>oscl_file_cache.h</code> defines the class <code>OsclFileCache</code> ) . . . . .	692
<code>oscl_file_dir_utils.h</code> (The file <code>oscl_file_dir_utils.h</code> defines some unix-style directory ops) . . . . .	693
<code>oscl_file_find.h</code> (The file <code>oscl_file_find.h</code> defines the class <code>Oscl_FileFind</code> ) . . . . .	695
<code>oscl_file_handle.h</code> (The file <code>oscl_file_handle.h</code> defines the class <code>OsclFileHandle</code> ) . . . . .	696
<code>oscl_file_io.h</code> (The file <code>oscl_file_io.h</code> defines the class <code>Oscl_File</code> . This is the public API to the basic file I/O operations) . . . . .	697
<code>oscl_file_manager.h</code> (File management class) . . . . .	698
<code>oscl_file_native.h</code> (The file <code>oscl_file_native.h</code> defines the class <code>OsclNativeFile</code> . This is the porting layer for basic file I/O operations) . . . . .	699
<code>oscl_file_server.h</code> (The file <code>oscl_file_server.h</code> defines the class <code>Oscl_FileServer</code> . This is the porting layer for file server implementations) . . . . .	700
<code>oscl_file_stats.h</code> (File stats class) . . . . .	701
<code>oscl_file_types.h</code> (The file <code>oscl_file_types.h</code> defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here) . . . . .	702
<code>oscl_heapbase.h</code> (OSCL Heap Base include file) . . . . .	703
<code>oscl_init.h</code> (Global oscl initialization) . . . . .	704
<code>oscl_int64_utils.h</code> . . . . .	705
<code>oscl_ip_socket.h</code> . . . . .	706
<code>oscl_linked_list.h</code> (The file <code>oscl_linked_list.h</code> defines the template class <code>Oscl_Linked_List</code> which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter) . . . . .	707
<code>oscl_lock_base.h</code> (This file defines an abstract lock class, <code>OsclLockBase</code> , that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, <code>OsclNullLock</code> , is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the <code>OsclScopedLock</code> class which is template class takes care of freeing the lock when the class goes out of scope) . . . . .	708
<code>oscl_map.h</code> (The file <code>oscl_map.h</code> defines the template class <code>Oscl_Map</code> which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter) . . . . .	709
<code>oscl_math.h</code> (Provides math functions) . . . . .	710
<code>oscl_media_data.h</code> (Defines a container class for media data made up of a collection of memory fragments) . . . . .	711
<code>oscl_media_status.h</code> (Defines a status values for the <code>MediaData</code> containers) . . . . .	712
<code>oscl_mem.h</code> (This file contains basic memory definitions for common use across platforms) . . . . .	713
<code>oscl_mem_audit.h</code> (This file contains the definition and partial implementation of MM_Audit class) . . . . .	716
<code>oscl_mem_audit_internals.h</code> (This file contains the internal definitions for the mem audit library) . . . . .	718
<code>oscl_mem_auto_ptr.h</code> (This file defines the <code>oscl_mem_auto_ptr</code> template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error) . . . . .	719
<code>oscl_mem_basic_functions.h</code> (This file contains prototypes for the basic memory functions) . . . . .	720
<code>oscl_mem_inst.h</code> (The file defines default memory instrumentation level) . . . . .	721
<code>oscl_mem_mempool.h</code> (This file contains the definition of memory pool allocators) . . . . .	722
<code>oscl_mutex.h</code> (This file provides implementation of mutex) . . . . .	723
<code>oscl_namestring.h</code> (Name string class include file) . . . . .	724
<code>oscl_opaque_type.h</code> (The file <code>oscl_opaque_type.h</code> defines pure virtual classes for working with opaque types) . . . . .	725

<a href="#">oscl_pqueue.h</a> (Implements a priority queue data structure similar to STL) . . . . .	726
<a href="#">oscl_procstatus.h</a> . . . . .	727
<a href="#">oscl_queue.h</a> (The file <a href="#">oscl_queue.h</a> defines the template class <b>Oscl_Queue</b> . 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) . . . . .	728
<a href="#">oscl_rand.h</a> (Provides pseudo-random number generation) . . . . .	729
<a href="#">oscl_refcounter.h</a> (A general purpose reference counter to object lifetimes) . . . . .	730
<a href="#">oscl_refcounter_memfrag.h</a> (This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount) . . . . .	731
<a href="#">oscl_registry_access_client.h</a> (Client-side implementation Registry Access implementation) . . . . .	732
<a href="#">oscl_registry_client.h</a> (Client-side implementation of OsclRegistry) . . . . .	733
<a href="#">oscl_registry_client_impl.h</a> (Client-side implementation of OsclRegistryInterface) . . . . .	734
<a href="#">oscl_registry_serv_impl.h</a> (Server-side implementation of OsclRegistry interfaces) . . . . .	735
<a href="#">oscl_registry_serv_impl_global.h</a> . . . . .	736
<a href="#">oscl_registry_serv_impl_tls.h</a> . . . . .	737
<a href="#">oscl_registry_types.h</a> (Common types used in Oscl registry interfaces) . . . . .	738
<a href="#">oscl_scheduler.h</a> . . . . .	739
<a href="#">oscl_scheduler_ao.h</a> (Oscl Scheduler user execution object classes) . . . . .	740
<a href="#">oscl_scheduler_aobase.h</a> (Oscl Scheduler internal active object classes) . . . . .	741
<a href="#">oscl_scheduler_readyq.h</a> (Ready q types for oscl scheduler) . . . . .	742
<a href="#">oscl_scheduler_threadcontext.h</a> (Thread context functions needed by oscl scheduler) . . . . .	743
<a href="#">oscl_scheduler_tuneables.h</a> (Tunable settings for Oscl Scheduler) . . . . .	744
<a href="#">oscl_scheduler_types.h</a> (Scheduler common types include file) . . . . .	745
<a href="#">oscl_semaphore.h</a> (This file provides implementation of mutex) . . . . .	746
<a href="#">oscl_shared_ptr.h</a> (This file defines a template class <b>OsclSharedPtr</b> which is a "smart pointer" to the parameterized type) . . . . .	747
<a href="#">oscl_singleton.h</a> (This file defines the <b>OsclSingleton</b> 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) . . . . .	748
<a href="#">oscl_snprintf.h</a> (Provides a portable implementation of snprintf) . . . . .	750
<a href="#">oscl_socket.h</a> (The file <a href="#">oscl_socket.h</a> defines the OSCL Socket APIs) . . . . .	751
<a href="#">oscl_socket_accept.h</a> . . . . .	752
<a href="#">oscl_socket_bind.h</a> . . . . .	753
<a href="#">oscl_socket_connect.h</a> . . . . .	754
<a href="#">oscl_socket_imp.h</a> . . . . .	755
<a href="#">oscl_socket_imp_base.h</a> . . . . .	756
<a href="#">oscl_socket_imp_pv.h</a> . . . . .	757
<a href="#">oscl_socket_listen.h</a> . . . . .	758
<a href="#">oscl_socket_method.h</a> . . . . .	759
<a href="#">oscl_socket_recv.h</a> . . . . .	760
<a href="#">oscl_socket_recv_from.h</a> . . . . .	761
<a href="#">oscl_socket_request.h</a> . . . . .	762
<a href="#">oscl_socket_send.h</a> . . . . .	763
<a href="#">oscl_socket_send_to.h</a> . . . . .	764
<a href="#">oscl_socket_serv_imp.h</a> . . . . .	765
<a href="#">oscl_socket_serv_imp_base.h</a> . . . . .	766
<a href="#">oscl_socket_serv_imp_pv.h</a> . . . . .	767
<a href="#">oscl_socket_serv_imp_reqlist.h</a> . . . . .	768
<a href="#">oscl_socket_shutdown.h</a> . . . . .	769
<a href="#">oscl_socket_stats.h</a> . . . . .	770
<a href="#">oscl_socket_tuneables.h</a> . . . . .	772

<a href="#">oscl_socket_types.h</a>	774
<a href="#">oscl_stdstring.h</a> (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)	776
<a href="#">oscl_str_ptr_len.h</a> (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally)	778
<a href="#">oscl_string.h</a> (Provides a standardized set of string containers that can be used in place of character arrays)	779
<a href="#">oscl_string_containers.h</a> (Provides a standardized set of string containers that can be used in place of character arrays)	780
<a href="#">oscl_string_rep.h</a> (Contains some internal implementation for string containers)	781
<a href="#">oscl_string_uri.h</a> (Utilities to unescape URIs)	782
<a href="#">oscl_string_utf8.h</a> (Utilities to validate and truncate UTF-8 encoded strings)	783
<a href="#">oscl_string_utils.h</a> (Utilities to parse and convert strings)	784
<a href="#">oscl_string_xml.h</a> (Utilities to escape special characters in XML strings)	785
<a href="#">oscl_tagtree.h</a> (The file <a href="#">oscl_tagtree.h</a> ..)	786
<a href="#">oscl_tcp_socket.h</a>	787
<a href="#">oscl_thread.h</a>	788
<a href="#">oscl_tickcount.h</a> (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally)	790
<a href="#">oscl_time.h</a> (The file <a href="#">oscl_time.h</a> defines two classes <a href="#">NTPTime</a> and <a href="#">TimeValue</a> for getting, manipulating, and formatting time values. The <a href="#">TimeValue</a> class is based on the native system time format while <a href="#">NTPTime</a> is used for the standard Network Time Protocol format)	791
<a href="#">oscl_timer.h</a>	793
<a href="#">oscl_tls.h</a>	794
<a href="#">oscl_tree.h</a> (The file <a href="#">oscl_tree.h</a> defines the template class <a href="#">Oscl_Rb_Tree</a> which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the <a href="#">Oscl_Map</a> class. Memory allocation is abstracted through the use of an allocator template parameter)	795
<a href="#">oscl_types.h</a> (This file contains basic type definitions for common use across platforms)	796
<a href="#">oscl_udp_socket.h</a>	797
<a href="#">oscl_utf8conv.h</a> (Utilities to convert unicode to utf8 and vice versa)	798
<a href="#">oscl_uuid.h</a> (This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32)	799
<a href="#">oscl_uuid_utils.h</a>	800
<a href="#">oscl_vector.h</a> (The file <a href="#">oscl_vector.h</a> defines the template class <a href="#">Oscl_Vector</a> which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter)	801
<a href="#">osclconfig.h</a> (This file contains configuration information for the linux platform)	802
<a href="#">osclconfig_ansi_memory.h</a> (This file contains common typedefs based on the ANSI C limits.h header)	804
<a href="#">osclconfig_check.h</a>	805
<a href="#">osclconfig_compiler_warnings.h</a> (This file contains the ability to turn off/on compiler warnings)	806
<a href="#">osclconfig_error.h</a> (This file contains the common typedefs and header files needed to compile osclerror)	807
<a href="#">osclconfig_error_check.h</a>	808
<a href="#">osclconfig_global_new_delete.h</a>	809
<a href="#">osclconfig_global_placement_new.h</a>	810
<a href="#">osclconfig_io.h</a> (This file contains common typedefs based on the ANSI C limits.h header)	811
<a href="#">osclconfig_io_check.h</a>	822
<a href="#">osclconfig_ix86.h</a> (This file contains configuration information for the ix86 processor family)	823
<a href="#">osclconfig_lib.h</a> (This file contains configuration information for the ANSI build)	824

osclconfig_lib_check.h . . . . .	825
osclconfig_limits_typedefs.h (This file contains common typedefs based on the ANSI C limits.h header) . . . . .	826
osclconfig_memory.h . . . . .	827
osclconfig_memory_check.h . . . . .	828
osclconfig_no_os.h . . . . .	829
osclconfig_proc.h (This file contains configuration information for the linux platform) . . . . .	830
osclconfig_proc_check.h . . . . .	831
osclconfig_proc_unix_android.h . . . . .	833
osclconfig_proc_unix_common.h . . . . .	835
osclconfig_time.h . . . . .	837
osclconfig_time_check.h . . . . .	838
osclconfig_unix_android.h . . . . .	839
osclconfig_unix_common.h . . . . .	843
osclconfig_util.h . . . . .	847
osclconfig_util_check.h . . . . .	848
pvlogger.h (This file contains basic logger interfaces for common use across platforms) . . . . .	849
pvlogger_accessories.h . . . . .	857
pvlogger_c.h (This file contains basic logger interfaces for common use across platforms. C-callable version) . . . . .	858
pvlogger_registry.h . . . . .	860

# **Chapter 5**

## **oscl Page Index**

### **5.1 oscl Related Pages**

Here is a list of all related documentation pages:

Todo List . . . . .	861
---------------------	-----

# 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\_SYMBIAN\_SUPPORT 0
- #define OSCL\_HAS\_SAVAJE\_SUPPORT 0
- #define OSCL\_HAS\_PV\_C\_OS\_SUPPORT 0
- #define OSCL\_HAS\_ANDROID\_SUPPORT 0
- #define OSCL\_HAS\_IPHONE\_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_ANDROID_SUPPORT 0`

#### 6.1.1.5 `#define OSCL_HAS_BERKELEY_SOCKETS 0`

#### 6.1.1.6 `#define OSCL_HAS_IPHONE_SUPPORT 0`

#### 6.1.1.7 `#define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0`

#### 6.1.1.8 `#define OSCL_HAS_MSWIN_SUPPORT 0`

#### 6.1.1.9 `#define OSCL_HAS_PRAGMA_PACK 0`

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

- 6.1.1.10 #define OSCL\_HAS\_PTHREAD\_SUPPORT 0
- 6.1.1.11 #define OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_FUNCS 0
- 6.1.1.12 #define OSCL\_HAS\_PV\_C\_OS\_SUPPORT 0
- 6.1.1.13 #define OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS 0
- 6.1.1.14 #define OSCL\_HAS\_SAVAJE\_IO\_SUPPORT 0
- 6.1.1.15 #define OSCL\_HAS\_SAVAJE\_SUPPORT 0
- 6.1.1.16 #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 0
- 6.1.1.17 #define OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION 0
- 6.1.1.18 #define OSCL\_HAS\_SYMBIAN\_DNS\_SERVER 0
- 6.1.1.19 #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0
- 6.1.1.20 #define OSCL\_HAS\_SYMBIAN\_MATH 0
- 6.1.1.21 #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0
- 6.1.1.22 #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- 6.1.1.23 #define OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER 0
- 6.1.1.24 #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- 6.1.1.25 #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- 6.1.1.26 #define OSCL\_HAS\_UNIX\_SUPPORT 0
- 6.1.1.27 #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 0
- 6.1.1.28 #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 two classes [NTPTime](#) and [TimeValue](#) for getting, manipulating, and formatting time values. The [TimeValue](#) class is based on the native system time format while [NTPTime](#) is used for the standard Network Time Protocol format.*

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

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

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

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

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

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

## Data Structures

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

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

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

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

- class [OsclExclusivePtr](#)

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

- class [OsclExclusivePtrA](#)

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

- class [OsclLockBase](#)
- struct [OsclMemoryFragment](#)
- class [OsclNullLock](#)
- class [OsclRefCounter](#)
- class [OsclRefCounterDA](#)

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

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

- class [OsclSharedPtr](#)

*A parameterized smart pointer class.*

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

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

- class [TLSStorageOps](#)

## Defines

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

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

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

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

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

*Type casting macros.*

- #define [OSCL\\_STATIC\\_CAST](#)(*type*, *exp*) ((*type*)(*exp*))
- #define [OSCL\\_REINTERPRET\\_CAST](#)(*type*, *exp*) ((*type*)(*exp*))
- #define [OSCL\\_DYNAMIC\\_CAST](#)(*type*, *exp*) ((*type*)(*exp*))
- #define [OSCL\\_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 [EPV\\_ARM\\_GNUC](#) 1
- #define [EPV\\_ARM\\_RVCT](#) 2
- #define [EPV\\_ARM\\_MSEVC](#) 3
- #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 **OsclAny** TOsclTlsKey
- typedef int **c\_bool**

*The c\_bool type is mapped to an integer to provide a bool type for C interfaces.*

- typedef void **OsclAny**

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

- typedef char **mbchar**

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

- typedef unsigned int **uint**

*The uint type is a convenient abbreviation for unsigned int.*

- typedef uint8 **octet**

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

- typedef float **OsclFloat**

*The Float type defined as OsclFloat.*

- typedef OSCL\_NATIVE\_INT64\_TYPE **int64**
- typedef OSCL\_NATIVE\_UINT64\_TYPE **uint64**
- typedef OSCL\_NATIVE\_WCHAR\_TYPE **oscl\_wchar**
- typedef **oscl\_wchar OSCL\_TCHAR**

*define OSCL\_TCHAR*

## Enumerations

- enum **TimeUnits** { **SECONDS** = 0, **MILLISECONDS** = 1, **MICROSECONDS** = 2 }

*The TimeUnits enum can be used when constructing a TimeValue class.*

## Functions

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

- OSCL\_IMPORT\_REF int32 `oscl_Clstrncmp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2, uint32 count)
- OSCL\_IMPORT\_REF char `oscl_tolower` (const char car)
- OSCL\_IMPORT\_REF `oscl_wchar` `oscl_tolower` (const `oscl_wchar` car)
- OSCL\_IMPORT\_REF bool `oscl_isLetter` (const char car)
- OSCL\_IMPORT\_REF const char \* `oscl_strstr` (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF char \* `oscl_strstr` (char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `oscl_strstr` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strstr` (`oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF char \* `oscl_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, `OsclRefCounter` \*in\_refcnt)

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

## Variables

- const int `CTIME_BUFFER_SIZE` = 26
- const int `PV8601TIME_BUFFER_SIZE` = 21
- const 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 EPV\_ARM\_GNUC 1**

**6.2.2.4 #define EPV\_ARM\_MSEVC 3**

**6.2.2.5 #define EPV\_ARM\_RVCT 2**

**6.2.2.6 #define NULL (0)**

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

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

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

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

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

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

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

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

Type casting macros.

### Parameters:

*type* Destination type of cast

*exp* Expression to cast

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

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

**6.2.2.15 #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()**

**6.2.2.16 #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()**

- 
- 6.2.2.17 #define OSCL\_DYNAMIC\_CAST(type, exp) ((type)(exp))
- 6.2.2.18 #define OSCL\_HAS\_SINGLETON\_SUPPORT 1
- 6.2.2.19 #define OSCL\_INLINE inline
- 6.2.2.20 #define OSCL\_MAX(a, b) ((a) > (b) ? (a) : (b))
- 6.2.2.21 #define OSCL\_MIN(a, b) ((a) < (b) ? (a) : (b))
- 6.2.2.22 #define OSCL\_PACKED\_VAR "error"
- 6.2.2.23 #define OSCL\_REINTERPRET\_CAST(type, exp) ((type)(exp))
- 6.2.2.24 #define OSCL\_STATIC\_CAST(type, exp) ((type)(exp))
- 6.2.2.25 #define OSCL\_TEMPLATED\_DESTRUCTOR\_CALL(type, simple\_type) type ::  
~simple\_type ()
- 6.2.2.26 #define OSCL\_TLS\_BASE\_SLOTS OSCL\_TLS\_ID\_BASE\_LAST +1
- 6.2.2.27 #define OSCL\_TLS\_EXTERNAL\_SLOTS 0
- 6.2.2.28 #define OSCL\_TLS\_MAX\_SLOTS ( OSCL\_TLS\_BASE\_SLOTS +  
OSCL\_TLS\_EXTERNAL\_SLOTS)
- 6.2.2.29 #define OSCL\_UNSIGNED\_CONST(x) x
- 6.2.2.30 #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.

- 6.2.2.31 #define OSCL\_UNUSED\_RETURN(value) return value

- 6.2.2.32 #define OSCL\_VIRTUAL\_BASE(type) type

- 6.2.2.33 #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 OsclAny TOsclTlsKey`****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 OsclSharedPtr< TheClass >::Bind (TheClass \* *ptr*, OsclRefCounter \* *in\_refcnt*) [inline, inherited]

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

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

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

### 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 OsclSharedPtr< TheClass >::operator== (const OsclSharedPtr< 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 oscl\_Clstrcmp (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 oscl\_CIstrcmp (const char \* str1, const char \* str2)**

Case in-sensitive string comparision.

**Parameters:**

*str1* string to compare

*str2* string to compare

**Returns:**

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

**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\_streat (oscl\_wchar \* *dest*, const oscl\_wchar \* *src*)**

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

**Parameters:**

*dest* null terminated destination string  
*src* source string  
*count* number of characters to append.

**Returns:**

dest

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

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

**Parameters:**

*dest* null terminated destination string  
*src* source string

**Returns:**

dest

**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\_strcmp (const oscl\_wchar \* str1, const oscl\_wchar \* str2, uint32 count)**

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

**Parameters:**

*str1* String to compare

***str2*** String to compare

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

**Returns:**

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

#### 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\_strrchr (oscl\_wchar \*str, int32 c)**

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

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

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

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

**Parameters:**

*str* null terminated source string

*c* character to search for

**Returns:**

**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 begining 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 begining 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\_PVERRORTRAP = 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 [OsclAuditCB](#)
- class [OsclMem](#)
- class [OsclMemAllocator](#)
- class [OsclMemAllocator](#)
- class [OsclMemAllocDestructDealloc](#)
- class [OsclMemAllocDestructDealloc](#)
- class [OsclMemAudit](#)
- class [OSCLMemAutoPtr](#)

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

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

## Defines

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

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

## Typedefs

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

## Functions

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

- OSCL\_COND\_IMPORT\_REF int [oscl\\_memcmp](#) (const void \*buf1, const void \*buf2, uint32 count)
- OSCL\_COND\_IMPORT\_REF [uint oscl\\_mem\\_aligned\\_size](#) (uint size)
- OSCL\_IMPORT\_REF void [OsclMemInit](#) (OsclAuditCB &auditCB)
- OSCL\_IMPORT\_REF void \* [\\_oscl\\_audit\\_malloc](#) (size\_t, OsclAuditCB &, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* [\\_oscl\\_audit\\_calloc](#) (size\_t, size\_t, OsclAuditCB &, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* [\\_oscl\\_audit\\_realloc](#) (void \*, size\_t, OsclAuditCB &, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* [\\_oscl\\_audit\\_new](#) (size\_t, OsclAuditCB &, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* [\\_oscl\\_default\\_audit\\_malloc](#) (size\_t, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* [\\_oscl\\_default\\_audit\\_calloc](#) (size\_t, size\_t, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* [\\_oscl\\_default\\_audit\\_realloc](#) (void \*, size\_t, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void \* [\\_oscl\\_default\\_audit\\_new](#) (size\_t, const char \*f=NULL, const int l=0)
- OSCL\_IMPORT\_REF void [\\_oscl\\_audit\\_free](#) (void \*)
- void \* [operator new](#) (size\_t aSize, const char \*aFile, int aLine)
- void \* [operator new](#) (size\_t)
- void [operator delete](#) (void \*)
- void \* [operator new\[\]](#) (size\_t aSize, const char \*aFile, int aLine)
- void \* [operator new\[\]](#) (size\_t aSize)
- void [operator delete\[\]](#) (void \*aPtr)

## Variables

- const uint32 [ALLOC\\_NODE\\_FLAG](#) = 0x80000000

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

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),audit-CB)) T params**

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

**Parameters:**

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

**Returns:**

pointer to the newly created object of type *T*

**Exceptions:**

*may* leave with code = bad alloc

**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 oscl\_calloc(a, b) OSCL\_CALLOC(a,b)****6.3.1.34 #define OSCL\_CALLOC(num, size) \_oscl\_default\_audit\_calloc(num,size)**

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

**Parameters:**

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

**Returns:**

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

**Exceptions:**

*none*

**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, Oscl_TAlloc<MM_AllocNode, OsclMemBasicAllocator> > MM_AllocNodeAutoPtr`

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

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

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

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

6.3.2.6 `typedef Oscl_TagTree<MM_StatsNodeTagTreeType, TagTree_Allocator> OsclTagTreeType`

6.3.2.7 `typedef Oscl_TAlloc<MM_StatsNodeTagTreeType, OsclMemBasicAllocator> TagTree_Allocator`

### 6.3.3 Function Documentation

6.3.3.1 `OSCL_IMPORT_REF void* _oscl_audit_calloc (size_t, size_t, OsclAuditCB &, 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, OsclAuditCB &, 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, OsclAuditCB &, const char \**f*=NULL, const int *l*=0)**
- 6.3.3.5 OSCL\_IMPORT\_REF void\* \_oscl\_audit\_realloc (void \*, size\_t, OsclAuditCB &, 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 aligned 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\\_pqueue.h](#)  
*Implements a priority queue data structure similar to STL.*
- file [oscl\\_rand.h](#)  
*Provides pseudo-random number generation.*
- file [oscl\\_registry\\_access\\_client.h](#)  
*Client-side implementation Registry Access implementation.*
- file [oscl\\_registry\\_client.h](#)  
*Client-side implementation of OsclRegistry.*
- file [oscl\\_registry\\_client\\_impl.h](#)  
*Client-side implementation of OsclRegistryInterface.*
- file [oscl\\_registry\\_serv\\_impl.h](#)  
*Server-side implementation of OsclRegistry interfaces.*
- file [oscl\\_registry\\_types.h](#)  
*Common types used in Oscl registry interfaces.*
- file [oscl\\_snprintf.h](#)  
*Provides a portable implementation of sprintf.*
- file [oscl\\_str\\_ptr\\_len.h](#)  
*Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.*
- file [oscl\\_string.h](#)  
*Provides a standardized set of string containers that can be used in place of character arrays.*
- file [oscl\\_string\\_containers.h](#)  
*Provides a standardized set of string containers that can be used in place of character arrays.*

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

## Data Structures

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

- class [OsclBinOStream](#)

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

- class [OsclBinOStreamBigEndian](#)

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

- class [OsclBinOStreamLittleEndian](#)

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

- class [OsclBinStream](#)

- class [OsclCompareLess](#)

- class [OsclComponentRegistry](#)

- class [OsclComponentRegistryData](#)

- class [OsclComponentRegistryElement](#)

- class [OsclPriorityQueue](#)

- class [OsclPriorityQueueBase](#)

- class [OsclRand](#)

- class [OsclRegistryAccessClient](#)

- class [OsclRegistryAccessClientImpl](#)

- class [OsclRegistryAccessClientTlsImpl](#)

- class [OsclRegistryAccessElement](#)

- class [OsclRegistryClient](#)

- class [OsclRegistryClientImpl](#)

- class [OsclRegistryClientTlsImpl](#)

- class [OsclRegistryServTlsImpl](#)

- class [OsclTickCount](#)

- struct [StrCSumPtrLen](#)

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

- struct [StrPtrLen](#)

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

- struct [WStrPtrLen](#)

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

## Defines

- #define [oscl\\_isdigit](#)(c) ((c) >= '0' && (c) <= '9')
- #define [OSCLTICKCOUNT\\_MAX\\_TICKS](#) 0xffffffff
- #define [MAX\\_NUMBER\\_OF\\_BYTE\\_PER\\_UTF8](#) 3

## Typedefs

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

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

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

## 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, [OsclFloat](#) &value)
- **OSCL\_IMPORT\_REF bool PV\_atof** (const char \*buf, int length, [OsclFloat](#) &value)
- **OSCL\_IMPORT\_REF int oscl\_abs** (int aVal)
- **OSCL\_COND\_IMPORT\_REF double oscl\_log** (double value)
- **OSCL\_COND\_IMPORT\_REF double oscl\_log10** (double value)
- **OSCL\_COND\_IMPORT\_REF double oscl\_sqrt** (double value)
- **OSCL\_COND\_IMPORT\_REF double oscl\_pow** (double x, double y)
- **OSCL\_COND\_IMPORT\_REF double oscl\_exp** (double value)
- **OSCL\_COND\_IMPORT\_REF double oscl\_sin** (double value)
- **OSCL\_COND\_IMPORT\_REF double oscl\_cos** (double value)
- **OSCL\_COND\_IMPORT\_REF double oscl\_tan** (double value)
- **OSCL\_COND\_IMPORT\_REF double oscl\_asin** (double value)
- **OSCL\_COND\_IMPORT\_REF double oscl\_atan** (double value)
- **OSCL\_COND\_IMPORT\_REF double oscl\_floor** (double value)
- **OSCL\_IMPORT\_REF int32 oscl\_snprintf** (char \*str, uint32 count, const char \*fmt,...)
- **OSCL\_IMPORT\_REF int32 oscl\_snprintf** ([oscl\\_wchar](#) \*str, uint32 count, const [oscl\\_wchar](#) \*fmt,...)
- **OSCL\_IMPORT\_REF int32 oscl\_vsnprintf** (char \*str, uint32 count, const char \*fmt, va\_list args)

- OSCL\_IMPORT\_REF int32 `oscl_vsnprintf` (oscl\_wchar \*str, uint32 count, const oscl\_wchar \*fmt, va\_list args)
- OSCL\_IMPORT\_REF bool `oscl_str_unescape_uri` (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes, uint32 &out\_buf\_len)
 

*unescape any of the special escape sequence in the uri string*
- OSCL\_IMPORT\_REF bool `oscl_str_unescape_uri` (const OSCL\_String &oscl\_str\_in, OSCL\_String &oscl\_str\_out, uint32 &out\_buf\_len)
 

*unescape any of the special escape sequence in the uri string*
- OSCL\_IMPORT\_REF bool `oscl_str_is_valid_utf8` (const uint8 \*str\_buf, uint32 &num\_valid\_characters, uint32 max\_bytes=0, uint32 max\_char\_2\_valid=0, uint32 \*num\_byte\_4\_char=NULL)
 

*Check if the input string contains any illegal UTF-8 character. The function scans the string and validate that each character is a valid utf-8. It stops at the first NULL character, invalid character or the max\_byte value. The string is valid if and only if every character is a valid utf-8 character and the scanning stopped on a character boundary.*
- OSCL\_IMPORT\_REF int32 `oscl_str_truncate_utf8` (uint8 \*str\_buf, uint32 max\_char, uint32 max\_bytes=0)
 

*Truncates the UTF-8 string upto the required size.*
- OSCL\_IMPORT\_REF bool `oscl_str_need_escape_xml` (const char \*str\_buf, uint32 &num\_escape\_bytes, uint32 max\_bytes=0)
 

*Check if the input string contains any special ASCII character like &, <, >, ', ". The function scans the string and check if each character is a special character. It stops at the first NULL character (if max\_bytes = 0), or the max\_byte value.*
- OSCL\_IMPORT\_REF int32 `oscl_str_escape_xml` (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes=0, uint32 \*num\_bytes\_written=NULL)
 

*Escape any of the following special characters in the string Special ASCII characters: &, <, >, ', ".*
- OSCL\_IMPORT\_REF int32 `oscl_UTF8ToUnicode` (const char \*input, int32 inLength, oscl\_wchar \*output, int32 outLength)
 

*Convert UTF8 byte sequence to Unicode string.*
- OSCL\_IMPORT\_REF int32 `oscl_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 \* BuffFragGroup< ChainClass, max\_frags >::GetBufferState (const int32 idx) [inline, inherited]**

**6.4.4.20 template<class ChainClass, uint32 max\_frags> BufferFragment \* BuffFragGroup< 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 sine 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\_outring 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 buferr is not enough for output. In case of (A), it adds a terminated '\0' at the end of the output UTF8 byte sequence, and returns length of the output UTF8 byte sequence (without counting terminated '\0'). In case of (B), it converts as much as possible to the output buffer and adds a terminated '\0' at the end of the output UTF8 byte sequence "(no '\0' added if outLength is less than or equal to 0, return 0)", and returns 0.

**Parameters:**

*input* Ptr to an input Unicode string. '\0' 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 buferr is not enough for output, or parse error. In case of (A), it adds a terminated '\0' at the end of the output Unicode string, and returns length of the output Unicode string (without counting terminated '\0'). In case of (B), it converts as much as possible to the output buffer and adds a terminated '\0' at the end of the output Unicode string "(no '\0' added if outLength is less than or equal to 0, return 0)", and returns 0.

**Parameters:**

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

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

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

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

**Returns:**

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



- 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*, *otype op*) [inherited]**

**6.4.4.85 template<uint32 MaxBufSize> void OSCL\_StackString< MaxBufSize >::set (const other\_chartype \* *buf*, *otype 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*, *otype op*) [inherited]**

**6.4.4.89 template<class Alloc> void OSCL\_HeapString< Alloc >::set (const other\_chartype \* *buf*, uint32 *length*, *otype 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*, *otype op*) [inherited]**

**6.4.4.91 template<class Alloc> void OSCL\_HeapString< Alloc >::set (const other\_chartype \* *buf*, *otype 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\\_namestring.h](#)  
*Name string class include file.*

### Data Structures

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

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

- class OsclErrorTrap
- class OsclErrorTrapImp
- class OsclException

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

- class OsclJump
- class OsclNameString
- class OsclTLSEx
- class OsclTLSRegistryEx
- class OsclTrapItem
- class OsclTrapStack
- class OsclTrapStackItem

## Defines

- #define OSCL\_TRAPSTACK\_PUSH(a) OsclError::PushL(a)
- #define OSCL\_TRAPSTACK\_POP() OsclError::Pop()
- #define OSCL\_TRAPSTACK\_POPDEALLOC() OsclError::PopDealloc()
- #define OsclErrNone 0
- #define OsclErrGeneral 100
- #define OsclErrNoMemory 101
- #define OsclErrCancelled 102
- #define OsclErrNotSupported 103
- #define OsclErrArgument 104
- #define OsclErrBadHandle 105
- #define OsclErrAlreadyExists 106
- #define OsclErrBusy 107
- #define OsclErrNotReady 108
- #define OsclErrCorrupt 109
- #define OsclErrTimeout 110
- #define OsclErrOverflow 111
- #define OsclErrUnderflow 112
- #define OsclErrInvalidState 113
- #define OsclErrNoResources 114
- #define OsclErrNotInstalled 115
- #define OsclErrAlreadyInstalled 116
- #define OsclErrSystemCallFailed 117
- #define OsclErrNoHandler 118
- #define OsclErrThreadContextIncorrect 119
- #define OSCL\_ERR\_NONE OsclErrNone
- #define OSCL\_BAD\_ALLOC\_EXCEPTION\_CODE OsclErrNoMemory
- #define OsclSuccess 0
- #define OsclPending 1
- #define OsclFailure -1
- #define PVERROR\_IMP\_JUMPS
- #define PVERROR\_DoLeave() internalLeave \_\_ilv; \_\_ilv.a=0; throw(\_\_ilv)
- #define \_PV\_TRAP(\_\_r, \_\_s)
- #define \_PV\_TRAP\_NO\_TLS(\_\_trapimp, \_\_r, \_\_s)
- #define OSCL\_JUMP\_MAX\_JUMP\_MARKS OSCL\_MAX\_TRAP\_LEVELS

- #define **internalLeave** (-1)
  - #define **OSCL\_MAX\_TRAP\_LEVELS** 20
  - #define **PVERRORTRAP\_REGISTRY\_ID** OSCL\_TLS\_ID\_PVERRORTRAP
  - #define **PVERRORTRAP\_REGISTRY** OsclTLSRegistry
  - #define **OSCL\_LEAVE**(\_leave\_status) OsclError::Leave(\_leave\_status)
- Use this macro to cause a Leave. It terminates the execution of the current active function.*
- #define **OSCL\_TRY**(\_leave\_status, \_statements) \_PV\_TRAP(\_leave\_status,\_statements)
- This macro will be used to set up a try block.*
- #define **OSCL\_TRY\_NO\_TLS**(\_trapimp, \_leave\_status, \_statements) \_PV\_TRAP\_NO\_TLS(\_-trapimp,\_leave\_status,\_statements)
  - #define **OSCL\_FIRST\_CATCH\_ANY**(\_leave\_status, \_statements) if (\_leave\_status!=OsclErrNone){ \_statements; }
- This section defines the macros to be used in the catch block following the try block. Use this macro to call a function that handles all exception types thrown in the preceding try block.*
- #define **OSCL\_FIRST\_CATCH**(\_leave\_status, \_catch\_value, \_statements) if (\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}
- Use this macro to define a block of code that catches the first exception type thrown in the preceding try block.*
- #define **OSCL\_CATCH**(\_leave\_status, \_catch\_value, \_statements) else if (\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}
- Use this macro to define a block of code for catching additional exception types.*
- #define **OSCL\_CATCH\_ANY**(\_leave\_status, \_statements) else if (\_leave\_status!=OsclErrNone){ \_-statements;}
- Use this macro to call a function that will catch all remaining exception types.*
- #define **OSCL\_LAST\_CATCH**(\_leave\_status) else if (\_leave\_status!=OsclErrNone){OSCL\_-LEAVE(\_leave\_status);}
- Use this macro if OSCL\_CATCH\_ANY has not been used. It will mark the end of the catch block.*

## Typedefs

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

## Functions

- OSCL\_IMPORT\_REF bool **OSCL\_IsErrnoSupported** ()  
*This function determines if a particular system saves the error number that occurs on a system call.*
- OSCL\_IMPORT\_REF int **OSCL\_GetLastError** ()  
*This function returns the value of the system's global error number variable.*
- OSCL\_IMPORT\_REF bool **OSCL\_SetLastError** (int newVal)

*This function sets the last error code for the system.*

- OSCL\_IMPORT\_REF char \* **OSCL\_StrError** (int errnum)

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

## 6.5.1 Define Documentation

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

**Value:**

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

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

**Value:**

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

### 6.5.1.3 #define internalLeave (-1)

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

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

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

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

**Parameters:**

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

*exceptiontype* is the exception handled by this catch block

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

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

**Parameters:**

*\_leave\_status*

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

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

For backward compatibility with old definitions

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

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

**Parameters:**

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

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

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

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

**Parameters:**

*\_leave\_status*

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

**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!=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.

**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) OsclError::Leave(\_leave\_status)**

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

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

**Parameters:**

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

**6.5.1.13 #define OSCL\_MAX\_TRAP\_LEVELS 20****6.5.1.14 #define OSCL\_TRAPSTACK\_POP() OsclError::Pop()****6.5.1.15 #define OSCL\_TRAPSTACK\_POPDEALLOC() OsclError::PopDealloc()****6.5.1.16 #define OSCL\_TRAPSTACK\_PUSH(a) OsclError::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 PVError_DoLeave() internalLeave __ilv;__ilv.a=0;throw(__ilv)

6.5.1.44 #define PVERROR_IMP_JUMPS
```

**6.5.1.45 #define PVERRORTRAP\_REGISTRY OsclTLSRegistry**

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

## 6.5.2 Typedef Documentation

**6.5.2.1 typedef int32 OsclLeaveCode**

Leave Codes

**6.5.2.2 typedef int32 OsclReturnCode**

Return Codes

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

**OsclTrapItem** 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 [OsclFileCache](#).*
- file [oscl\\_file\\_dir\\_utils.h](#)  
*The file [oscl\\_file\\_dir\\_utils.h](#) defines some unix-style directory ops.*
- file [oscl\\_file\\_find.h](#)  
*The file [oscl\\_file\\_find.h](#) defines the class [Oscl\\_FileFind](#).*
- file [oscl\\_file\\_handle.h](#)  
*The file [oscl\\_file\\_handle.h](#) defines the class [Oscl\\_FileHandle](#).*
- file [oscl\\_file\\_io.h](#)  
*The file [oscl\\_file\\_io.h](#) defines the class [Oscl\\_File](#). This is the public API to the basic file I/O operations.*
- file [oscl\\_file\\_manager.h](#)  
*File management class.*
- file [oscl\\_file\\_native.h](#)  
*The file [oscl\\_file\\_native.h](#) defines the class [OsclNativeFile](#). This is the porting layer for basic file I/O operations.*
- file [oscl\\_file\\_server.h](#)  
*The file [oscl\\_file\\_server.h](#) defines the class [Oscl\\_FileServer](#). This is the porting layer for file server implementations.*
- file [oscl\\_file\\_stats.h](#)  
*File stats class.*
- file [oscl\\_file\\_types.h](#)  
*The file [oscl\\_file\\_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.*
- file [oscl\\_socket.h](#)  
*The file [oscl\\_socket.h](#) defines the OSCL Socket APIs.*

### Data Structures

- class [Oscl\\_File](#)
- class [Oscl\\_FileFind](#)
- class [Oscl\\_FileServer](#)
- struct [oscl\\_fsstat](#)

- struct `oscl_stat_buf`
- class `OsclIDNS`
- class `OsclDNSObserver`
- class `OsclFileCache`
- class `OsclFileCacheBuffer`
- class `OsclFileHandle`
- class `OsclFileManager`
- class `OsclFileStats`
- class `OsclFileStatsItem`
- class `OsclNativeFile`
- class `OsclNativeFileParams`
- class `OsclSocketServ`
- class `OsclTCPSocket`
- class `OsclUDPSocket`

## Defines

- #define `TOsclFileOffsetInt32` int32
- #define `OSCL_FILE_STATS_LOGGER_NODE` "OsclFileStats"
- #define `OSCL_IO_FILENAME_MAXLEN` 512
- #define `OSCL_IO_EXTENSION_MAXLEN` 512
- #define `OSCL_FILE_WCHAR_PATH_DELIMITER` \_STRLIT("/")
- #define `OSCL_FILE_CHAR_PATH_DELIMITER` \_STRLIT\_CHAR("/")

## Typedefs

- typedef `oscl_fsstat` `OSCL_FSSTAT`
- typedef `oscl_stat_buf` `OSCL_STAT_BUF`
- typedef FILE \* `TOsclFileHandle`

## Enumerations

- enum `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 `TOsclFileOp` { `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` }

## Functions

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

### 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 "OsclFileStats"

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 TOsclFileOffsetInt32 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\* TOsclFileHandle

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

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

oscl\_stat function can be used to determine the 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 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_stat (const oscl\_wchar \* path, OSCL\_STAT\_BUF \*statbuf)

oscl\_stat function can be used to determine the 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 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_statsfs (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 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_statsfs (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](#)  
*Oscl Scheduler user execution object classes.*
- file [oscl\\_scheduler\\_aobase.h](#)  
*Oscl Scheduler internal active object classes.*
- file [oscl\\_scheduler\\_readyq.h](#)  
*ready q types for oscl scheduler*
- file [oscl\\_scheduler\\_threadcontext.h](#)  
*Thread context functions needed by oscl scheduler.*
- file [oscl\\_scheduler\\_tuneables.h](#)  
*Tuneable settings for Oscl Scheduler.*
- file [oscl\\_scheduler\\_types.h](#)  
*Scheduler common types include file.*

### Data Structures

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

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

## Defines

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

## Typedefs

- typedef PVActiveBase \* TOsclReady

## Enumerations

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

## Functions

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

## Variables

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

## 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 PVEEXECNAMELEN 30**

**6.7.1.11 #define PVSCEDNAMELEN 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 [OsclInit](#)
- class [OsclSelect](#)

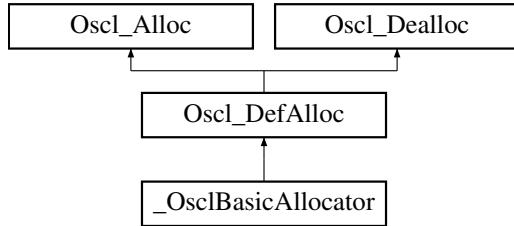
# Chapter 7

## oscl Data Structure Documentation

### 7.1 \_OsclBasicAllocator Class Reference

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

Inheritance diagram for \_OsclBasicAllocator::



#### Public Methods

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

#### 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 Oscl code only. Higher level code should use [OsclMemAllocator](#) defined in "[oscl\\_mem.h](#)".

## 7.1.2 Constructor & Destructor Documentation

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

## 7.1.3 Member Function Documentation

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

Implements [Oscl\\_DefAlloc](#).

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

Implements [Oscl\\_DefAlloc](#).

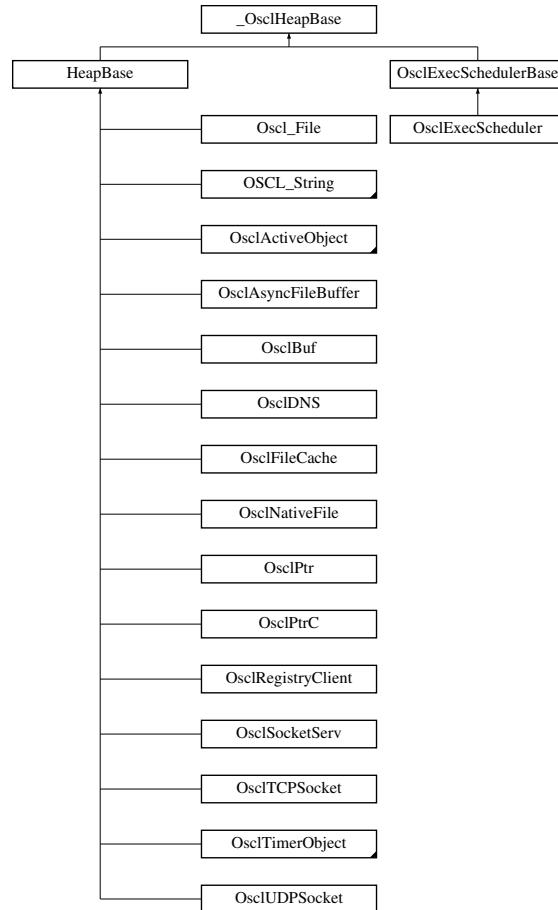
The documentation for this class was generated from the following file:

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

## 7.2 \_OsclHeapBase Class Reference

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

Inheritance diagram for \_OsclHeapBase::



### Public Methods

- virtual ~\_OsclHeapBase ()

### Protected Methods

- \_OsclHeapBase ()
- \_OsclHeapBase (const \_OsclHeapBase &)

### Friends

- class PVCleanupStack

### 7.2.1 Detailed Description

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

### 7.2.2 Constructor & Destructor Documentation

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

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

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

### 7.2.3 Friends And Related Function Documentation

7.2.3.1 `friend class PVCleanupStack [friend]`

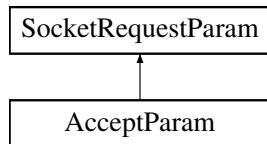
The documentation for this class was generated from the following file:

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

## 7.3 AcceptParam Class Reference

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

Inheritance diagram for AcceptParam::



### Public Methods

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

### Data Fields

- [OsclSocketI \\* iBlankSocket](#)

#### 7.3.1 Constructor & Destructor Documentation

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

#### 7.3.2 Field Documentation

7.3.2.1 [OsclSocketI\\* 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.

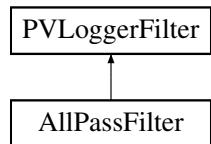
The documentation for this class was generated from the following file:

- [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 FilterOpaqueMessge \(char \\*tag, message\\_id\\_type msgID, log\\_level\\_type level\)](#)

#### 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](#).

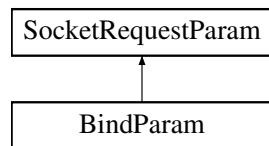
The documentation for this class was generated from the following file:

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

## 7.6 BindParam Class Reference

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

Inheritance diagram for BindParam::



### Public Methods

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

### Data Fields

- [OsclNetworkAddress iAddr](#)

#### 7.6.1 Constructor & Destructor Documentation

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

#### 7.6.2 Field Documentation

7.6.2.1 [OsclNetworkAddress BindParam::iAddr](#)

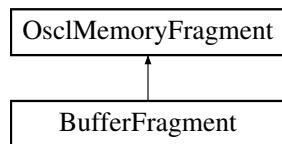
The documentation for this class was generated from the following file:

- [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 getRefCount ()`
- `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::getRefCount ()`** [inline]

**7.9.2.8 `void BufferState::increment_refcnt ()`** [inline]

**7.9.2.9 `void BufferState::reset ()`** [inline]

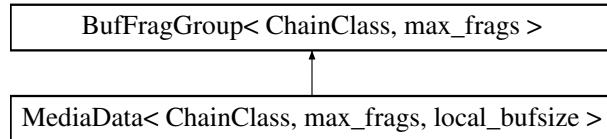
The documentation for this class was generated from the following file:

- `oscl_media_data.h`

## 7.10 BufFragGroup< ChainClass, max\_frags > Class Template Reference

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

Inheritance diagram for BufFragGroup< ChainClass, max\_frags >::



### Public Methods

- [BufFragGroup \(\)](#)
- virtual [~BufFragGroup \(\)](#)
- int32 [GetMaxFrags \(\) const](#)
- int32 [GetNumFrags \(\) const](#)
- uint32 [GetLength \(\) const](#)
- [BufferFragment \\* GetFragment \(const int32 idx\)](#)
- [BufferState \\* GetBufferState \(const int32 idx\)](#)
- void [AppendNext \(ChainClass \\*next\\_ptr\)](#)
- ChainClass \* [GetNext \(\) const](#)

### Protected Methods

- virtual void [Clear \(\)](#)
- [BufFragStatusClass::status\\_t AddFragment \(const BufferFragment &frag, BufferState \\*in\\_buffer\\_state, int32 location\\_offset=max\\_frags\)](#)

### Protected Attributes

- [BufferFragment fragments \[max\\_frags\]](#)
- [BufferState \\* buffer\\_states \[max\\_frags\]](#)
- [ChainClass \\* next](#)
- uint32 [num.fragments](#)
- uint32 [length](#)

```
template<class ChainClass, uint32 max_frags> class BufFragGroup< ChainClass, max_frags >
```

### 7.10.1 Constructor & Destructor Documentation

**7.10.1.1** `template<class ChainClass, uint32 max_frags> BufFragGroup< ChainClass, max_frags >::BufFragGroup () [inline]`

**7.10.1.2** `template<class ChainClass, uint32 max_frags> virtual BufFragGroup< ChainClass, max_frags >::~BufFragGroup () [inline, virtual]`

### 7.10.2 Member Function Documentation

**7.10.2.1** `template<class ChainClass, uint32 max_frags> BufFragStatusClass::status\_t BufFragGroup< ChainClass, max_frags >::AddFragment (const BufferFragment & frag, BufferState * in_buffer_state, int32 location_offset = max_frags) [inline, protected]`

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

**7.10.2.3** `template<class ChainClass, uint32 max_frags> virtual void BufFragGroup< ChainClass, max_frags >::Clear () [inline, protected, virtual]`

Reimplemented in [MediaData< ChainClass, max\\_frags, local\\_bufsize >](#).

**7.10.2.4** template<class ChainClass, uint32 max\_frags> uint32 BufFragGroup< ChainClass, max\_frags >::GetLength () const [inline]

**7.10.2.5** template<class ChainClass, uint32 max\_frags> int32 BufFragGroup< ChainClass, max\_frags >::GetMaxFrags () const [inline]

**7.10.2.6** template<class ChainClass, uint32 max\_frags> ChainClass\* BufFragGroup< ChainClass, max\_frags >::GetNext () const [inline]

**7.10.2.7** template<class ChainClass, uint32 max\_frags> int32 BufFragGroup< ChainClass, max\_frags >::GetNumFrags () const [inline]

### 7.10.3 Field Documentation

**7.10.3.1** template<class ChainClass, uint32 max\_frags> [BufferState\\*](#) BufFragGroup< ChainClass, max\_frags >::buffer\_states[max\_frags] [protected]

**7.10.3.2** template<class ChainClass, uint32 max\_frags> [BufferFragment](#) BufFragGroup< ChainClass, max\_frags >::fragments[max\_frags] [protected]

**7.10.3.3** template<class ChainClass, uint32 max\_frags> uint32 BufFragGroup< ChainClass, max\_frags >::length [protected]

**7.10.3.4** template<class ChainClass, uint32 max\_frags> ChainClass\* BufFragGroup< ChainClass, max\_frags >::next [protected]

**7.10.3.5** template<class ChainClass, uint32 max\_frags> uint32 BufFragGroup< ChainClass, max\_frags >::num\_frags [protected]

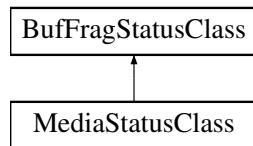
The documentation for this class was generated from the following file:

- [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`

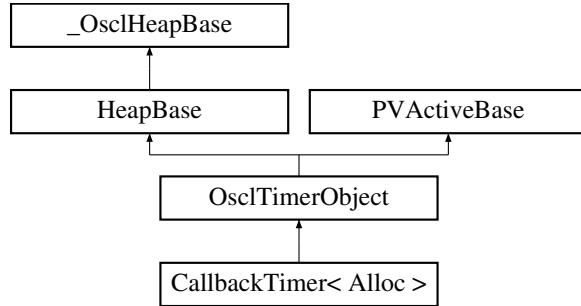
The documentation for this class was generated from the following file:

- [oscl\\_media\\_status.h](#)

## 7.12 CallbackTimer< Alloc > Class Template Reference

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

Inheritance diagram for CallbackTimer< Alloc >::



### Public Methods

- [CallbackTimer \(CallbackTimerObserver &aContainer, const char \\*name, int32 aPriority=OsclActiveObject::EPriorityNominal\)](#)
- [~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 = OsclActiveObject::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](#).

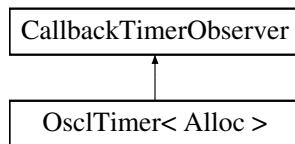
The documentation for this class was generated from the following file:

- [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 [OsclTimer< 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](#) (CHheapRep \*&, [Oscl\\_DefAlloc](#) &, const char \*, uint32)
- OSCL\_IMPORT\_REF void [set\\_rep](#) (CHheapRep \*&, [Oscl\\_DefAlloc](#) &, const [oscl\\_wchar](#) \*, uint32)
- OSCL\_IMPORT\_REF void [append\\_rep](#) (CHheapRep \*&, [Oscl\\_DefAlloc](#) &, const char \*, uint32)
- OSCL\_IMPORT\_REF void [append\\_rep](#) (CHheapRep \*&, [Oscl\\_DefAlloc](#) &, const [oscl\\_wchar](#) \*, uint32)
- OSCL\_IMPORT\_REF void [assign](#) (CHheapRep \*&, CHheapRep \*, [Oscl\\_DefAlloc](#) &)

### Data Fields

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

#### 7.15.1 Detailed Description

For internal use only– heap string representation

## 7.15.2 Constructor & Destructor Documentation

7.15.2.1 `OSCL_IMPORT_REF void 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`

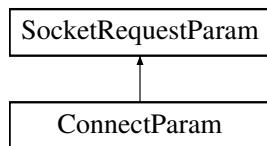
The documentation for this class was generated from the following file:

- `oscl_string_rep.h`

## 7.16 ConnectParam Class Reference

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

Inheritance diagram for ConnectParam::



### Public Methods

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

### Data Fields

- [OsclNetworkAddress iAddr](#)

#### 7.16.1 Constructor & Destructor Documentation

**7.16.1.1 ConnectParam::ConnectParam (OsclNetworkAddress & *anAddr*) [inline]**

#### 7.16.2 Field Documentation

**7.16.2.1 OsclNetworkAddress 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

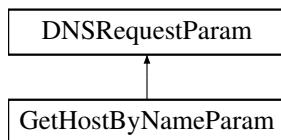
The documentation for this class was generated from the following file:

- [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

- TPVDNSFx<sub>n</sub> iFx<sub>n</sub>
- OsclDNSRequest \* iDNSRequest

### Protected Methods

- DNSRequestParam (TPVDNSFx<sub>n</sub> aFx<sub>n</sub>)

### Protected Attributes

- uint32 iRefCount

#### 7.18.1 Constructor & Destructor Documentation

**7.18.1.1** virtual DNSRequestParam::~DNSRequestParam () [inline, virtual]

**7.18.1.2** DNSRequestParam::DNSRequestParam (TPVDNSFx<sub>n</sub> aFx<sub>n</sub>) [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 OsclDNSRequest\* DNSRequestParam::iDNSRequest**

**7.18.3.2 TPVDNSFxn DNSRequestParam::iFxn**

**7.18.3.3 uint32 DNSRequestParam::iRefCount [protected]**

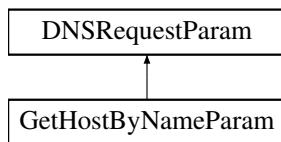
The documentation for this class was generated from the following file:

- [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 `OsclNetworkAddress` &`addr`)
- bool `canPersistMoreHostAddresses` ()

### Static Public Methods

- `GetHostByNameParam * Create` (const char \*`name`, `OsclNetworkAddress` \*&`addr`, `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >` \*`aAddressList`)

### Data Fields

- `char * iName`
- `OsclNetworkAddress * iAddr`
- `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * 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, OsclNetworkAddress *& addr, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aAddressList) [static]`

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

Implements [DNSRequestParam](#).

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

## 7.19.4 Field Documentation

7.19.4.1 `OsclNetworkAddress* GetHostByNameParam::iAddr`

7.19.4.2 `Oscl_Vector<OsclNetworkAddress, OsclMemAllocator>* GetHostByNameParam::i-AddressList`

7.19.4.3 `char* GetHostByNameParam::iName`

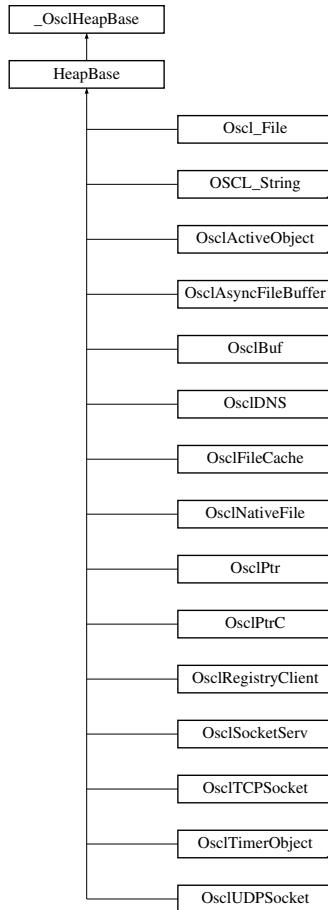
The documentation for this class was generated from the following file:

- [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 [\\_OsclHeapBase](#) providing CBase\* alike pointer and virtual destructor for cleanupstack to Push and Pop for cleanup when leave occurs.

HeapBase has a virtual destructor which calls the destructor of all the derived classes.

## 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
```

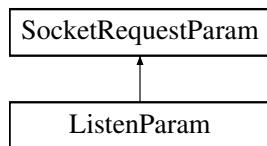
The documentation for this class was generated from the following file:

- [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](#)

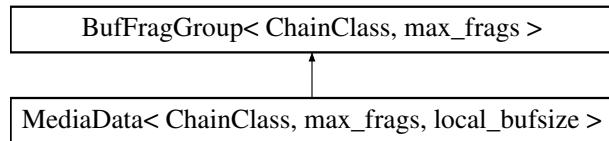
The documentation for this class was generated from the following file:

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

## 7.24 MediaData< ChainClass, max\_frags, local\_bufsize > Class Template Reference

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

Inheritance diagram for MediaData< ChainClass, max\_frags, local\_bufsize >::



### Public Methods

- [MediaData \(\)](#)
- virtual [~MediaData \(\)](#)
- uint32 [GetLocalBufsize \(\) const](#)
- [MediaTimestamp GetTimestamp \(\) const](#)
- void [SetTimestamp \(MediaTimestamp in\\_timestamp\)](#)
- uint32 [GetAvailableBufferSize \(\) const](#)
- [MediaStatusClass::status\\_t GetLocalFragment \(BufferFragment &fragment\)](#)
- virtual void [Clear \(\)](#)
- bool [IsLocalData \(const OsclMemoryFragment &frag\) const](#)
- int [GetMediaSize \(\) const](#)
- [BufferFragment \\* GetMediaFragment \(const uint32 idx\)](#)
- uint32 [GetNumMediaFrags \(const uint32 idx\) const](#)

### Protected Methods

- [MediaStatusClass::status\\_t AddLocalFragment \(const BufferFragment &frag, int32 location\\_offset\)](#)

### Protected Attributes

- [MediaTimestamp timestamp](#)
- uint8 [localbuf \[local\\_bufsize\]](#)
- uint32 [available\\_localbuf](#)
- int [num\\_reserved\\_fragments](#)

template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> class MediaData< ChainClass, max\_frags, local\_bufsize >

### 7.24.1 Constructor & Destructor Documentation

- 7.24.1.1 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> MediaData< ChainClass, max\_frags, local\_bufsize >::MediaData () [inline]
- 7.24.1.2 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> virtual MediaData< ChainClass, max\_frags, local\_bufsize >::~MediaData () [inline, virtual]

### 7.24.2 Member Function Documentation

- 7.24.2.1 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> MediaStatusClass::status\_t MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment (const BufferFragment &frag, int32 location\_offset) [inline, protected]
- 7.24.2.2 template<class ChainClass, uint32 max\_frags, uint32 local\_bufsize> virtual void MediaData< ChainClass, max\_frags, local\_bufsize >::Clear () [inline, virtual]

Reimplemented from [BufFragGroup< ChainClass, max\\_frags >](#).

- 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 >::GetNumMediaFrags (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 **OsclMemoryFragment** & *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]

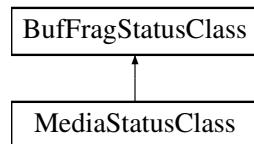
The documentation for this class was generated from the following file:

- [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 \(oscl\\_memsize\\_t size\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size, MM\\_AllocInfo \\*ptr\)](#)
- [void operator delete \(void \\*ptr\) throw \(\)](#)

### Data Fields

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

## 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 \(\)](#)
- [OSCL\\_IMPORT\\_REF void \\* MM\\_allocate \(const OsclMemStatsNode \\*statsNode, uint32 sizeIn, const char \\*pFileName, uint32 lineNumber, bool allocNodeTracking=false\)](#)
- [OSCL\\_IMPORT\\_REF bool MM\\_deallocate \(void \\*pMemBlockIn\)](#)
- [OSCL\\_IMPORT\\_REF MM\\_Stats\\_t \\* MM\\_GetStats \(const char \\*const tagIn\)](#)
- [OSCL\\_IMPORT\\_REF uint32 MM\\_GetStatsInDepth \(const char \\*tagIn, MM\\_Stats\\_CB \\*array\\_ptr, uint32 max\\_nodes\)](#)
- [OSCL\\_IMPORT\\_REF uint32 MM\\_GetTreeNodes \(const char \\*tagIn\)](#)
- [OSCL\\_IMPORT\\_REF bool MM\\_AddTag \(const char \\*tagIn\)](#)
- [OSCL\\_IMPORT\\_REF const OsclMemStatsNode \\* MM\\_GetTagName \(const char \\*tagIn\)](#)
- [OSCL\\_IMPORT\\_REF const OsclMemStatsNode \\* MM\\_GetExistingTag \(const char \\*tagIn\)](#)
- [OSCL\\_IMPORT\\_REF const OsclMemStatsNode \\* MM\\_GetRootNode \(\)](#)
- [OSCL\\_IMPORT\\_REF MM\\_AllocQueryInfo \\* MM\\_CreateAllocNodeInfo \(uint32 max\\_array\\_size\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_ReleaseAllocNodeInfo \(MM\\_AllocQueryInfo \\*info\)](#)
- [OSCL\\_IMPORT\\_REF uint32 MM\\_GetAllocNodeInfo \(MM\\_AllocQueryInfo \\*output\\_array, uint32 max\\_array\\_size, uint32 offset\)](#)
- [OSCL\\_IMPORT\\_REF bool MM\\_Validate \(const void \\*ptrIn\)](#)
- [uint32 MM\\_GetAllocNo \(void\)](#)
- [void MM\\_GetOverheadStats \(MM\\_AuditOverheadStats &stats\)](#)
- [uint32 MM\\_GetNumAllocNodes \(\)](#)
- [uint32 MM\\_GetMode \(void\)](#)
- [uint8 MM\\_GetPrefillPattern \(void\)](#)
- [uint32 MM\\_GetPostfillPattern \(void\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_SetMode \(uint32 inMode\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_SetPrefillPattern \(uint8 pattern\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_SetPostfillPattern \(uint8 pattern\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_SetTagLevel \(uint32 level\)](#)
- [OSCL\\_IMPORT\\_REF bool MM\\_SetFailurePoint \(const char \\*tagIn, uint32 alloc\\_number\)](#)
- [OSCL\\_IMPORT\\_REF void MM\\_UnsetFailurePoint \(const char \\*tagIn\)](#)
- [MM\\_AllocNode \\* addAllocNode \(void \\*pMem, uint32 sizeIn, OsclMemStatsNode \\*pStatsNode, const char \\*pFileName, uint32 lineNumber\)](#)
- [OsclMemStatsNode \\* removeAllocNode \(void \\*pMemBlockIn, uint32 &size\)](#)
- [void removeALLAllocNodes \(\)](#)
- [OsclMemStatsNode \\* createStatsNode \(const char \\*tagIn\)](#)
- [bool updateStatsNode \(OsclMemStatsNode \\*pCurrStatsNode, const MM\\_Stats\\_t &pDelta, bool bAdd\)](#)
- [bool updateStatsNodeInFailure \(const char \\*tagIn\)](#)
- [bool updateStatsNodeInFailure \(OsclMemStatsNode \\*pStatsNode\)](#)
- [bool pruneSubtree \(OsclMemStatsNode \\*pNode\)](#)
- [bool pruneSubtree \(const char \\*tagIn\)](#)
- [void retrieveParentTag \(char \\*tag\)](#)
- [int32 retrieveParentTagLength \(const char \\*tag, int32 bound\)](#)
- [void makeValidTag \(const char \\*tagIn, MMAuditCharAutoPtr &autoPtr\)](#)

- uint32 [getTagActualSize](#) (const char \*tagIn)
- bool [isSetFailure](#) (const char \*tagIn)
- bool [isSetFailure](#) (OsclMemStatsNode \*statsNode)
- bool [validate\\_all\\_heap](#) ()

## Static Public Methods

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

### 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, OsclMemStatsNode \* pStatsNode, const char \* pFileName, uint32 lineNumber)

##### Returns:

true if operation succeeds;

#### 7.32.2.2 OsclMemStatsNode\* MM\_Audit\_Imp::createStatsNode (const char \* tagIn)

##### Returns:

true if operation succeeds;

#### 7.32.2.3 OsclMemAudit\* 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 (OsclMemStatsNode \* 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 OSCL\_IMPORT\_REF bool MM\_Audit\_Imp::MM\_AddTag (const char \* tagIn) [inline]**

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

true if operation succeeds;

**7.32.2.10 OSCL\_IMPORT\_REF void\* MM\_Audit\_Imp::MM\_allocate (const OsclMemStatsNode \* statsNode, uint32 sizeIn, const char \* pFileName, uint32 lineNumber, bool allocNodeTracking = false)**

The following are APIs t \_\_nothrow\_ / const \_\_nothrow\_

**Returns:**

the memory pointer if operation succeeds.

**7.32.2.11 OSCL\_IMPORT\_REF MM\_AllocQueryInfo\* MM\_Audit\_Imp::MM\_CreateAllocNode-Info (uint32 max\_array\_size)**

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

**7.32.2.12 OSCL\_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 OsclMemStatsNode\* MM\_Audit\_Imp::MM\_GetExisting-  
Tag (const char \* tagIn)**

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

true if operation succeeds;

**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 OsclMemStatsNode\* 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 OsclMemStatsNode\* MM\_Audit\_Imp::MM\_GetTagName (const char \* tagIn)**

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

pointer to [OsclMemStatsNode](#) which should be passed to MM\_allocate

**7.32.2.25 OSCL\_IMPORT\_REF uint32 MM\_Audit\_Imp::MM\_GetTreeNodes (const char \* tagIn)**

API to get the number of tree nodes including the tag node and its subtree

**Parameters:**

*tagIn* input tag

**Returns:**

the number of tree nodes ; 0 means no tag node

**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 (OsclMemStatsNode * pNode)`

**Returns:**

true if operation succeeds;

**7.32.2.36** `void MM_Audit_Imp::removeALLAllocNodes ()`

**7.32.2.37** `OsclMemStatsNode* 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 (OsclMemStatsNode * pCurrStatsNode, const MM_Stats_t & pDelta, bool bAdd)`

**Returns:**

true if operation succeeds;

**7.32.2.41** `bool MM_Audit_Imp::updateStatsNodeInFailure (OsclMemStatsNode * 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 &npt) const**  
*The - operator allows subtraction of one NTPTime value from another. This is useful to measure an interval.*
- **void set\_from\_system\_time (const uint32 systemtime)**  
*This method converts a 32-bit system time to NTP time.*
- **OSCL\_COND\_IMPORT\_REF uint32 get\_middle32 () const**  
*Grab the middle 32 bits of the 64 bit 32.32 representation.*
- **OSCL\_COND\_IMPORT\_REF uint32 get\_upper32 () const**  
*This method returns the upper 32 bits of the 32.32 representation.*
- **OSCL\_COND\_IMPORT\_REF uint32 get\_lower32 () const**  
*This method returns the lower 32 bits of the 32.32 representation.*

- int32 [to\\_system\\_time \(\) const](#)

*This method converts the ntp time value to system time.*

- OSCL\_COND\_IMPORT\_REF [uint64 get\\_value \(\) const](#)

*This method returns the 32.32 ntp representation.*

- OSCL\_IMPORT\_REF int [set\\_to\\_current\\_time \(\)](#)

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

### 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 & npt)  
const**

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

**Parameters:**

*npt* A reference to the NTPTime object to be subtracted from this one.

**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.

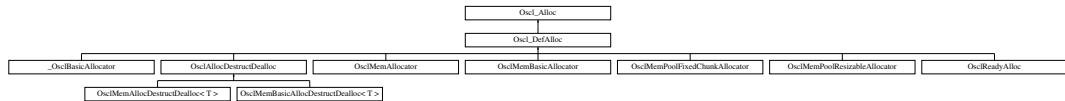
The documentation for this class was generated from the following file:

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

## 7.38 Oscl\_Alloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oscl\_Alloc::



### Public Methods

- virtual `~Oscl_Alloc ()`
- virtual `OsclAny * allocate (const uint32 size)=0`
- virtual `OsclAny * allocate_fl (const uint32 size, const char *file_name, const int line_num)`

#### 7.38.1 Constructor & Destructor Documentation

**7.38.1.1 virtual Oscl\_Alloc::~Oscl\_Alloc () [inline, virtual]**

#### 7.38.2 Member Function Documentation

**7.38.2.1 virtual OsclAny\* Oscl\_Alloc::allocate (const uint32 size) [pure virtual]**

Implemented in `_OsclBasicAllocator`, `Oscl_DefAlloc`, `OsclMemAllocator`, `OsclMemBasicAllocator`, `OsclMemAllocDestructDealloc< T >`, `OsclMemBasicAllocDestructDealloc< T >`, `OsclMemPoolFixedChunkAllocator`, `OsclMemPoolResizableAllocator`, and `OsclReadyAlloc`.

**7.38.2.2 virtual OsclAny\* Oscl\_Alloc::allocate\_fl (const uint32 size, const char \*file\_name, const int line\_num) [inline, virtual]**

Reimplemented in `Oscl_DefAlloc`, `OsclMemAllocator`, `OsclMemAllocDestructDealloc< T >`, and `OsclReadyAlloc`.

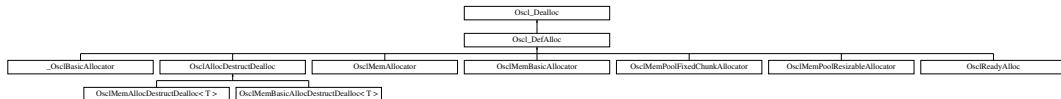
The documentation for this class was generated from the following file:

- `oscl_defalloc.h`

## 7.39 Oscl\_Dealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oscl\_Dealloc::



### Public Methods

- virtual void [deallocate \(OsclAny \\*p\)=0](#)
- virtual [~Oscl\\_Dealloc \(\)](#)

#### 7.39.1 Constructor & Destructor Documentation

**7.39.1.1 virtual Oscl\_Dealloc::~Oscl\_Dealloc () [inline, virtual]**

#### 7.39.2 Member Function Documentation

**7.39.2.1 virtual void Oscl\_Dealloc::deallocate (OsclAny \*p) [pure virtual]**

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

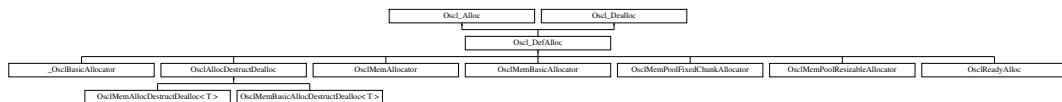
The documentation for this class was generated from the following file:

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

## 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](#).

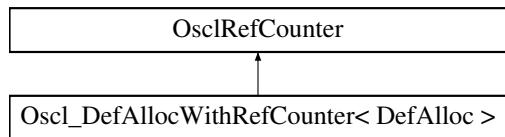
The documentation for this class was generated from the following file:

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

## 7.41 Oscl\_DefAllocWithRefCounter< DefAlloc > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for Oscl\_DefAllocWithRefCounter< DefAlloc >::



### Public Methods

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

### Static Public Methods

- Oscl\_DefAllocWithRefCounter \* [New](#) ()

#### 7.41.1 Detailed Description

**template<class DefAlloc> class Oscl\_DefAllocWithRefCounter< DefAlloc >**

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

#### 7.41.2 Member Function Documentation

**7.41.2.1 template<class DefAlloc> void Oscl\_DefAllocWithRefCounter< DefAlloc >::addRef () [inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**7.41.2.2 template<class DefAlloc> void Oscl\_DefAllocWithRefCounter< DefAlloc >::Delete () [inline]**

Delete object

**7.41.2.3 template<class DefAlloc> uint32 Oscl\_DefAllocWithRefCounter< DefAlloc >::getCount () [inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**7.41.2.4 template<class DefAlloc> Oscl\_DefAllocWithRefCounter\*<br/>Oscl\_DefAllocWithRefCounter< DefAlloc >::New () [inline, static]**

Create object

**7.41.2.5 template<class DefAlloc> void Oscl\_DefAllocWithRefCounter< DefAlloc >::removeRef<br/>() [inline, virtual]**

Delete from reference count

Implements [OsclRefCounter](#).

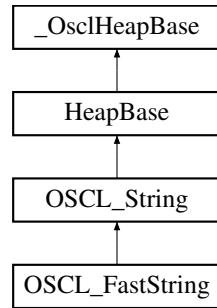
The documentation for this class was generated from the following file:

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

## 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 chartype\* 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 chartype\* OSCL\_FastString::get\_str () [virtual]**

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

Implements [OSCL\\_String](#).

**7.42.4.5 OSCL\_IMPORT\_REF OSCL\_FastString& OSCL\_FastString::operator= (const chartype \* 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\_chartype \* 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]

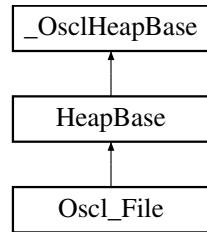
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 7.43 Oscl\_File Class Reference

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

Inheritance diagram for Oscl\_File::



### Public Types

- enum `seek_type` { `SEEKSET`, `SEEKCUR`, `SEEKEND` }
- enum `mode_type` { `MODE_READ` = 0x0001, `MODE_READWRITE` = 0x0002, `MODE_APPEND` = 0x0004, `MODE_BINARY` = 0x0008, `MODE_TEXT` = 0x0010, `MODE_READ_PLUS` = 0x0020 }
- enum `TSymbianAccessMode` { `ESymbianAccessMode_Rfile` = 0, `ESymbianAccessMode_RfileBuf` = 1 }

### Public Methods

- `OSCL_IMPORT_REF Oscl_File()`
- `OSCL_IMPORT_REF Oscl_File(uint32 aCacheSize)`
- `OSCL_IMPORT_REF Oscl_File(uint32 aCacheSize, OsclFileHandle *aFileHandle)`
- `OSCL_IMPORT_REF ~Oscl_File()`
- `OSCL_IMPORT_REF void SetPVCacheSize(uint32 aSize)`
- `void AddFixedCache(const OsclFixedCacheParam &aParam)`
- `void RemoveFixedCache(const TOsclFileOffset &aPos)`
- `void SetCacheObserver(OsclCacheObserver *aObs)`
- `OSCL_IMPORT_REF void SetNativeAccessMode(uint32 aMode)`
- `OSCL_IMPORT_REF void SetNativeBufferSize(int32 aSize)`
- `OSCL_IMPORT_REF void SetAsyncReadBufferSize(uint32 aSize)`
- `OSCL_IMPORT_REF int32 SetFileHandle(OsclFileHandle *aHandle)`
- `OSCL_IMPORT_REF int32 Open(const char *filename, uint32 mode, Oscl_FileServer &fileserv)`
- `OSCL_IMPORT_REF int32 Open(const oscl_wchar *filename, uint32 mode, Oscl_FileServer &fileserv)`
- `OSCL_IMPORT_REF uint32 Read(OsclAny *buffer, uint32 size, uint32 numelements)`
- `OSCL_IMPORT_REF uint32 Write(const OsclAny *buffer, uint32 size, uint32 numelements)`
- `OSCL_IMPORT_REF int32 Seek(TOscOfFileOffset offset, seek_type origin)`
- `OSCL_IMPORT_REF TOscOfFileOffset Tell()`
- `OSCL_IMPORT_REF int32 Close()`
- `OSCL_IMPORT_REF int32 Flush()`
- `OSCL_IMPORT_REF int32 SetSize(uint32 size)`
- `OSCL_IMPORT_REF int32 EndOfFile()`
- `OSCL_IMPORT_REF int32 GetError()`

- [OsclFileHandle \\* Handle \(\)](#)
- [OSCL\\_IMPORT\\_REF TOsclFileOffset Size \(\)](#)
- [OSCL\\_IMPORT\\_REF void SetLoggingEnable \(bool aEnable\)](#)
- [OSCL\\_IMPORT\\_REF void SetSummaryStatsLoggingEnable \(bool aEnable\)](#)

## Friends

- class [OsclFileCache](#)
- class [OsclFileCacheBuffer](#)
- class [asyncfilereadwrite\\_test](#)
- class [largeasyncfilereadwrite\\_test](#)
- class [asyncfilereadcancel\\_test](#)

### 7.43.1 Member Enumeration Documentation

#### 7.43.1.1 enum Oscl\_File::mode\_type

**Enumeration values:**

**MODE\_READ** Opens a file for reading. The file must exist.

**MODE\_READWRITE** Opens the file for reading and writing. If the file exists, its contents will be overwritten unless APPEND mode is specified. If the file does not exist, it will be created.

**MODE\_APPEND** Specifies all write operations to occur at the end of the file. The file pointer can be moved with the Seek command, but will always be moved to the end of the file for write commands.

**MODE\_BINARY** Opens the file in 'binary' mode. This is the default.

**MODE\_TEXT** Opens the file in 'text' mode. The default mode is 'binary'.

**MODE\_READ\_PLUS** Open a file for reading and writing. The file must exist. The default mode is 'binary'.

#### 7.43.1.2 enum Oscl\_File::seek\_type

**Enumeration values:**

**SEEKSET** Beginning of file

**SEEKCUR** Current position of file pointer

**SEEKEND** End of file

#### 7.43.1.3 enum Oscl\_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 Oscl\_File::Oscl\_File ()

Constructor

### 7.43.2.2 OSCL\_IMPORT\_REF Oscl\_File::Oscl\_File (uint32 *aCacheSize*)

Deprecated Constructor, present for back-compatibility.

**Parameters:**

*aCacheSize*: sets native buffer size, and when pv cache is enabled, also sets pv cache size.

### 7.43.2.3 OSCL\_IMPORT\_REF Oscl\_File::Oscl\_File (uint32 *aCacheSize*, **OsclFileHandle** \* *aFileHandle*)

Deprecated Constructor, present for back-compatibility.

**Parameters:**

*aCacheSize*: sets native buffer size, and when pv cache is enabled, also sets pv cache size.

*aFileHandle*: open file handle.

### 7.43.2.4 OSCL\_IMPORT\_REF Oscl\_File::~Oscl\_File ()

Destructor

## 7.43.3 Member Function Documentation

### 7.43.3.1 void Oscl\_File::AddFixedCache (const **OsclFixedCacheParam** & *aParam*) [inline]

AddFixedCache adds a fixed cache. The fixed cache will be used on the next opportunity. The fixed cache must not overlap with any other fixed cache.

**Parameters:**

*aParam*: Cache location and size.

### 7.43.3.2 OSCL\_IMPORT\_REF int32 Oscl\_File::Close ()

The File Close operation Closes the file after flushing any remaining data in the buffers.

Note: If the file object was opened with an external file handle, then Close will simply flush the file. The file will remain open.

**Returns:**

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

**7.43.3.3 OSCL\_IMPORT\_REF int32 Oscl\_File::EndOfFile ()**

The File EOF(end of file) operation returns a nonzero value after the first read operation that attempts to read past the end of the file

**Returns:****7.43.3.4 OSCL\_IMPORT\_REF int32 Oscl\_File::Flush ()**

The File Flush operation On an output stream OSCL\_FileFlush causes any buffered but unwritten data to be written to the file. Flush is meant for writable files. The behavior when calling it on read-only files is OS-dependent.

**Returns:**

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

**7.43.3.5 OSCL\_IMPORT\_REF int32 Oscl\_File::GetError ()**

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

**Returns:****7.43.3.6 OsclFileHandle\* Oscl\_File::Handle () [inline]**

Retrieve the file handle.

**Returns:**

file handle

**7.43.3.7 OSCL\_IMPORT\_REF int32 Oscl\_File::Open (const oscl\_wchar \*filename, uint32 mode, Oscl\_FileServer &fileserv)**

Opens a file.

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

**Parameters:**

*filename* name of file to open (Unicode)

*mode* combination of open mode flags

*fileserv* fileserv to use

**Returns:**

returns 0 if successful and a non-zero value otherwise

#### 7.43.3.8 OSCL\_IMPORT\_REF int32 Oscl\_File::Open (const char \*filename, uint32 mode, Oscl\_FileServer &fileserv)

Opens a file.

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

**Parameters:**

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

**Returns:**

returns 0 if successful and a non-zero value otherwise

#### 7.43.3.9 OSCL\_IMPORT\_REF uint32 Oscl\_File::Read (OsclAny \*buffer, uint32 size, uint32 numelements)

The File Read operation Reads from the file into the buffer a maximum of 'numelements' of size 'size'.

**Parameters:**

*buffer* pointer to buffer of type void  
*size* element size in bytes  
*numelements* max number of elements to read

**Returns:**

returns the number of full elements actually read, which may be less than count if an error occurs or if the end of the file is encountered before reaching count. Use the CheckEndOfFile or GetError function to distinguish a read error from an end-of-file condition.

#### 7.43.3.10 void Oscl\_File::RemoveFixedCache (const TOsclFileOffset &aPos) [ inline ]

RemoveFixedCache removes a fixed cache.

**Parameters:**

*aPos*: Cache location and size.

#### 7.43.3.11 OSCL\_IMPORT\_REF int32 Oscl\_File::Seek (TOsclFileOffset offset, seek\_type origin)

The File Seek operation Sets the position for file pointer

**Parameters:**

*offset* offset from the specified origin.  
*origin* starting point

**Returns:**

returns 0 on success, and a non-zero value otherwise

#### 7.43.3.12 OSCL\_IMPORT\_REF void Oscl\_File::SetAsyncReadBufferSize (uint32 *aSize*)

SetAsyncReadBufferSize configures the asynchronous background read function. May not be available on all platforms.

This should be called before opening the file. If used when the file is open, the option will not take effect until the next Open.

Note: if asynchronous read is not available on the platform, this call will have no effect.

**Parameters:**

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

#### 7.43.3.13 void Oscl\_File::SetCacheObserver ([OsclCacheObserver](#) \* *aObs*) [inline]

#### 7.43.3.14 OSCL\_IMPORT\_REF int32 Oscl\_File::SetFileHandle ([OsclFileHandle](#) \* *aHandle*)

SetFileHandle adds an open file handle to the Oscl\_File object. The Oscl\_File object will use that handle to access the file.

This call is not available when the Oscl\_File object is already open.

Note: This feature is used in Symbian with the MMF framework. The MMF framework provides an open RFile handle to access content. When using RFileBuf access mode with an RFile handle, the RFileBuf will be attached to the open RFile handle.

To use the external file handle, the caller starts with a native file handle to an open file. The caller must wrap the native file handle in an [OsclFileHandle](#) object, pass the [OsclFileHandle](#) pointer to SetFileHandle, call [Oscl\\_File::Open](#), then proceed to use the Oscl\_File object, finally calling [Oscl\\_File::Close](#). In this usage mode, [Oscl\\_File::Open](#) and [Oscl\\_File::Close](#) do not actually call native file open and close. It is assumed that the caller will close the original native file handle after usage is complete.

**Parameters:**

*aHandle*: container for an open file handle.

**Returns:**

returns 0 if successful, non-zero if error.

#### 7.43.3.15 OSCL\_IMPORT\_REF void Oscl\_File::SetLoggingEnable (bool *aEnable*)

SetLoggingEnable configures the [PVLogger](#) output for this file. This will enable full logging of each API entry and exit using the logger object "Oscl\_File", plus full logging of native operation entry & exit using logger object "OsclNativeFile".

**Parameters:**

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

#### 7.43.3.16 OSCL\_IMPORT\_REF void Oscl\_File::SetNativeAccessMode (uint32 *aMode*)

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

Note: for Symbian, use the TSymbianAccessMode values to choose the mode. If multiple access modes are not available on the platform, this call will have no effect.

**Parameters:**

*aMode*: access mode.

**7.43.3.17 OSCL\_IMPORT\_REF void Oscl\_File::SetNativeBufferSize (int32 *aSize*)**

SetNativeBufferSize configures the native file buffering feature, when available.

This should be called before opening the file. If used when the file is open, the option will not take effect until the next Open.

Note: For Symbian, this sets the RFileBuf cache size. If native buffering is not available on the platform, this call will have no effect.

**Parameters:**

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

**7.43.3.18 OSCL\_IMPORT\_REF void Oscl\_File::SetPVCacheSize (uint32 *aSize*)**

SetPVCacheSize configures the read/write cache.

This should be called before opening the file. If used when the file is open, the option will not take effect until the next Open.

**Parameters:**

*aSize*: cache size in bytes. Zero disables the cache.

**7.43.3.19 OSCL\_IMPORT\_REF int32 Oscl\_File::SetSize (uint32 *size*)**

The File SetSize operation If the file has been opened for writing this will set the size of the file. The file pointer position is undefined after calling SetSize. If file size is increased the contents of the new section are undefined.

**Returns:**

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

**7.43.3.20 OSCL\_IMPORT\_REF void Oscl\_File::SetSummaryStatsLoggingEnable (bool *aEnable*)**

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

**Parameters:**

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

**7.43.3.21 OSCL\_IMPORT\_REF [TOsclFileOffset](#) Oscl\_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 [TOsclFileOffset](#) Oscl\_File::Tell ()

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

#### 7.43.3.23 OSCL\_IMPORT\_REF uint32 Oscl\_File::Write (const [OsclAny](#) \* *buffer*, uint32 *size*, uint32 *numelements*)

The File Write operation Writes from the buffer '*numelements*' objects of size '*size*'

**Parameters:**

*buffer* pointer to buffer of type void

*size* element size in bytes

*numelements* number of elements to write

**Returns:**

The number of elements written

### 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 [largeasynccfilereadwrite\\_test](#) [friend]

#### 7.43.4.4 friend class [OsclFileCache](#) [friend]

#### 7.43.4.5 friend class [OsclFileCacheBuffer](#) [friend]

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

- [oscl\\_file\\_io.h](#)

## 7.44 Oscl\_File::OsclCacheObserver Class Reference

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

### Public Methods

- virtual ~OsclCacheObserver ()
- virtual OsclFileCacheBuffer \* ChooseCurCache (OsclFileCache &aContext, TOsclFileOffset 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 Constructor & Destructor Documentation

**7.44.2.1 virtual Oscl\_File::OsclCacheObserver::~OsclCacheObserver () [inline, virtual]**

#### 7.44.3 Member Function Documentation

**7.44.3.1 virtual OsclFileCacheBuffer\* Oscl\_File::OsclCacheObserver::ChooseCurCache (OsclFileCache & aContext, TOsclFileOffset aPos) [pure virtual]**

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

- [oscl\\_file\\_io.h](#)

## 7.45 Oscl\_File::OsclFixedCacheParam Class Reference

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

### Public Methods

- bool [Contains \(TOsclFileOffset pos\) const](#)

### Data Fields

- [TOsclFileOffset 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 Oscl\_File::OsclFixedCacheParam::Contains (TOsclFileOffset *pos*) const  
[inline]**

#### 7.45.3 Field Documentation

**7.45.3.1 TOsclFileOffset Oscl\_File::OsclFixedCacheParam::iFilePosition**

**7.45.3.2 uint32 Oscl\_File::OsclFixedCacheParam::iSize**

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

- [oscl\\_file\\_io.h](#)

## 7.46 Oscl\_FileFind Class Reference

```
#include <oscl_file_find.h>
```

### Public Types

- enum `error_type` { `E_OK` = 0, `E_INVALID_STATE`, `E_INVALID_ARG`, `E_PATH_TOO_LONG`, `E_PATH_NOT_FOUND`, `E_NO_MATCH`, `E_BUFFER_TOO_SMALL`, `E_NOT_IMPLEMENTED`, `E_MEMORY_ERROR`, `E_OTHER` }
- enum `element_type` { `FILE_TYPE` = 0, `DIR_TYPE`, `INVALID_TYPE` }

### Public Methods

- OSCL\_IMPORT\_REF const char \* `FindFirst` (const char \*directory, const char \*pattern, char \*buf, uint32 buflen)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `FindFirst` (const `oscl_wchar` \*directory, const `oscl_wchar` \*pattern, `oscl_wchar` \*buf, uint32 buflen)
- OSCL\_IMPORT\_REF char \* `FindNext` (char \*buf, uint32 buflen)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `FindNext` (`oscl_wchar` \*buf, uint32 buflen)
- OSCL\_IMPORT\_REF void `Close` ()
- OSCL\_IMPORT\_REF `element_type` `GetElementType` ()
- OSCL\_IMPORT\_REF `error_type` `GetLastError` ()
- OSCL\_IMPORT\_REF `Oscl_FileFind` ()
- OSCL\_IMPORT\_REF `~Oscl_FileFind` ()

### 7.46.1 Detailed Description

`Oscl_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 Oscl\_FileFind::element\_type

Enumeration values:

`FILE_TYPE`

`DIR_TYPE`

`INVALID_TYPE`

#### 7.46.2.2 enum Oscl\_FileFind::error\_type

Enumeration values:

`E_OK`

`E_INVALID_STATE`

`E_INVALID_ARG`

`E_PATH_TOO_LONG`

**E\_PATH\_NOT\_FOUND**  
**E\_NO\_MATCH**  
**E\_BUFFER\_TOO\_SMALL**  
**E\_NOT\_IMPLEMENTED**  
**E\_MEMORY\_ERROR**  
**E\_OTHER**

### 7.46.3 Constructor & Destructor Documentation

#### 7.46.3.1 OSCL\_IMPORT\_REF Oscl\_FileFind::Oscl\_FileFind ()

constructor.

**Returns:**

none

#### 7.46.3.2 OSCL\_IMPORT\_REF Oscl\_FileFind::~Oscl\_FileFind ()

destructor. will deallocate open handles if necessary

**Returns:**

none

### 7.46.4 Member Function Documentation

#### 7.46.4.1 OSCL\_IMPORT\_REF void Oscl\_FileFind::Close ()

closes the handle to directory.

**Returns:**

none

#### 7.46.4.2 OSCL\_IMPORT\_REF const oscl\_wchar\* Oscl\_FileFind::FindFirst (const oscl\_wchar \* directory, const oscl\_wchar \* pattern, oscl\_wchar \* buf, uint32 buflen)

Opens a directory for reading.

**Parameters:**

*directory* directory to search (utf16).

*pattern* wildcard pattern filter (utf16). passing NULL, results in a universal match.

*buf* buffer for returned pathname (utf16).

*buflen* size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E\_BUFFER\_TOO\_SMALL.

**Returns:**

returns a pointer to buffer supplied, which contains the pathname of the first found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

#### 7.46.4.3 OSCL\_IMPORT\_REF const char\* Oscl\_FileFind::FindFirst (const char \* *directory*, const char \* *pattern*, char \* *buf*, uint32 *buflen*)

Finds first element matching the pattern.

**Parameters:**

*directory* directory to search (utf8).

*pattern* wildcard pattern filter (utf8). passing NULL, results in a universal match.

*buf* buffer for returned pathname (utf8).

*buflen* size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E\_BUFFER\_TOO\_SMALL.

**Returns:**

returns a pointer to buffer supplied, which contains the pathname of the first found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

#### 7.46.4.4 OSCL\_IMPORT\_REF oscl\_wchar\* Oscl\_FileFind::FindNext (oscl\_wchar \* *buf*, uint32 *buflen*)

Reads the next element in a directory. Note: the pointer returned by this function is not persistent and should be stored. Its scope is limited to the lifetime of the class.

**Parameters:**

*buf* buffer to hold directory name(utf16)

*buflen* size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E\_BUFFER\_TOO\_SMALL.

**Returns:**

returns a pointer to buffer supplied, which contains the pathname of the next found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

#### 7.46.4.5 OSCL\_IMPORT\_REF char\* Oscl\_FileFind::FindNext (char \* *buf*, uint32 *buflen*)

Reads the next element in the directory. Note: the pointer returned by this function is not persistent and should be stored. Its scope is limited to the lifetime of the class.

**Parameters:**

*buf* buffer to hold directory name(utf8)

*buflen* size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E\_BUFFER\_TOO\_SMALL.

**Returns:**

returns a pointer to buffer supplied, which contains the pathname of the next found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

#### 7.46.4.6 OSCL\_IMPORT\_REF *element\_type* Oscl\_FileFind::GetElementType ()

Returns the element type for the last element returned

**Returns:**

see enumeration above for more info.

#### 7.46.4.7 OSCL\_IMPORT\_REF [error\\_type](#) Oscl\_FileFind::GetLastError ()

Returns the error code for the last operation.

**Returns:**

see enumeration above for more info.

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

- [oscl\\_file\\_find.h](#)

## 7.47 Oscl\_FileServer Class Reference

```
#include <oscl_file_server.h>
```

### Public Methods

- OSCL\_IMPORT\_REF [Oscl\\_FileServer \(\)](#)
- OSCL\_IMPORT\_REF [~Oscl\\_FileServer \(\)](#)
- OSCL\_IMPORT\_REF int32 [Connect \(bool aShareSession=false\)](#)
- OSCL\_IMPORT\_REF int32 [Close \(\)](#)
- OSCL\_IMPORT\_REF int32 [Oscl\\_DeleteFile \(const char \\*filename\)](#)
- OSCL\_IMPORT\_REF int32 [Oscl\\_DeleteFile \(const oscl\\_wchar \\*filename\)](#)

### Friends

- class [Oscl\\_File](#)
- class [OsclNativeFile](#)

#### 7.47.1 Constructor & Destructor Documentation

##### 7.47.1.1 OSCL\_IMPORT\_REF Oscl\_FileServer::Oscl\_FileServer ()

Constructor

##### 7.47.1.2 OSCL\_IMPORT\_REF Oscl\_FileServer::~Oscl\_FileServer ()

Destructor

#### 7.47.2 Member Function Documentation

##### 7.47.2.1 OSCL\_IMPORT\_REF int32 Oscl\_FileServer::Close ()

Closes a file server.

**Returns:**

returns 0 on success and a non-zero value otherwise

##### 7.47.2.2 OSCL\_IMPORT\_REF int32 Oscl\_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 Oscl\_FileServer::Oscl\_DeleteFile (const oscl\_wchar \*  
*filename*)**

Deletes a file from the filesystem

**Parameters:**

*filename* name of the file to delete (Unicode)

**Returns:**

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

**7.47.2.4 OSCL\_IMPORT\_REF int32 Oscl\_FileServer::Oscl\_DeleteFile (const char \**filename*)**

Deletes a file from the filesystem \*

**Parameters:**

*filename* name of the file to delete (Utf8)

**Returns:**

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

**7.47.3 Friends And Related Function Documentation****7.47.3.1 friend class Oscl\_File [friend]****7.47.3.2 friend class OsclNativeFile [friend]**

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

- [oscl\\_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](#)

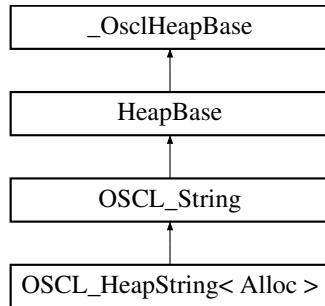
The documentation for this struct was generated from the following file:

- [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 otype`
- `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, otype op)`
- `void set(const other_chartype *buf, uint32 length, otype 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]**

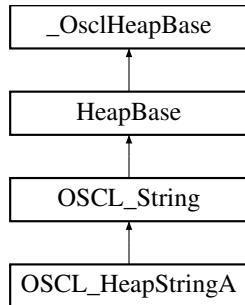
The documentation for this class was generated from the following file:

- [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 otype`
- `typedef OSCL_wString::chartype other_chartype`

### Public Methods

- `OSCL_IMPORT_REF OSCL_HeapStringA()`
- `OSCL_IMPORT_REF OSCL_HeapStringA(Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA(const OSCL_HeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA(const OSCL_HeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA(const OSCL_String &src, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA(const chartype *cstr, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA(const chartype *buf, uint32 length, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF ~OSCL_HeapStringA()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator=(const OSCL_HeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator=(const OSCL_String &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(const chartype *buf, uint32 length)`
- `OSCL_IMPORT_REF void set(const other_chartype *buf, otype op)`
- `OSCL_IMPORT_REF void set(const other_chartype *buf, uint32 length, otype 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 [OsclMemAllocator](#) will be used.

The string array is variable length, is allocated from the heap, and is modifiable. A copy-on-write mechanism is used to minimize unnecessary copying when multiple instances of a string are created for reading. Allocated memory is automatically freed by the class destructor when the last string referencing the memory is destroyed.

The class HAS NO thread synchronization built-in, so it is NOT MT-SAFE. External locks should be used if the class is to be shared across threads.

### 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 TOSCL_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 object will use an [OsclMemAllocator](#).

#### 7.50.3.2 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (Oscl_DefAlloc *alloc, OsclRefCounter *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 object will use an [OsclMemAllocator](#).

**7.50.3.4 OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const OSCL\_HeapStringA & src, Oscl\_DefAlloc \* alloc, OsclRefCounter \* ref = NULL)**

**7.50.3.5 OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const OSCL\_String & src, Oscl\_DefAlloc \* alloc = NULL, OsclRefCounter \* ref = NULL)**

**7.50.3.6 OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const chartype \* cstr, Oscl\_DefAlloc \* alloc = NULL, OsclRefCounter \* ref = NULL)**

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

**Parameters:**

*cp*: null-terminated string.

**am: (optional) allocator or reference-counted allocator.**

**am: (optional) reference counter associated with allocator object.**

If no allocator is provided, this object will use an [OsclMemAllocator](#).

**7.50.3.7 OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const chartype \* buf, uint32 length, Oscl\_DefAlloc \* alloc = NULL, OsclRefCounter \* ref = NULL)**

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

**Parameters:**

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

*length*: number of characters to copy.

**am: (optional) allocator or reference-counted allocator.**

**am: (optional) reference counter associated with allocator object.**

If no allocator is provided, this object will use an [OsclMemAllocator](#).

**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 chartype\* 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 Oscl\_Int64\_Utils Class Reference

The Oscl\_Int64\_Utils class provides a wrapper for commonly used int64/uint64 operations.

```
#include <oscl_int64_utils.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void `set_int64` (int64 &input\_value, const int32 upper, const 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 Oscl\_Int64\_Utils class provides a wrapper for commonly used int64/uint64 operations.

The Oscl\_Int64\_Utils class:

Provides a wrapper for commonly used operations to mask the differences between OSes that have an int64/uint64 class instead of a 64-bit integer.

## 7.51.2 Member Function Documentation

- 7.51.2.1 **OSCL\_IMPORT\_REF** **uint32** **Oscl\_Int64\_Utils::get\_int64\_lower32** (**const int64 & input\_value**) [static]
- 7.51.2.2 **OSCL\_IMPORT\_REF** **uint32** **Oscl\_Int64\_Utils::get\_int64\_middle32** (**const int64 & input\_value**) [static]
- 7.51.2.3 **OSCL\_IMPORT\_REF** **int32** **Oscl\_Int64\_Utils::get\_int64\_upper32** (**const int64 & input\_value**) [static]
- 7.51.2.4 **OSCL\_IMPORT\_REF** **uint32** **Oscl\_Int64\_Utils::get\_uint64\_lower32** (**const uint64 & input\_value**) [static]
- 7.51.2.5 **OSCL\_IMPORT\_REF** **uint32** **Oscl\_Int64\_Utils::get\_uint64\_middle32** (**const uint64 & input\_value**) [static]
- 7.51.2.6 **OSCL\_IMPORT\_REF** **uint32** **Oscl\_Int64\_Utils::get\_uint64\_upper32** (**const uint64 & input\_value**) [static]
- 7.51.2.7 **OSCL\_IMPORT\_REF** **void** **Oscl\_Int64\_Utils::set\_int64** (**int64 & input\_value, const int32 upper, const uint32 lower**) [static]
- 7.51.2.8 **OSCL\_IMPORT\_REF** **void** **Oscl\_Int64\_Utils::set\_uint64** (**uint64 & input\_value, const uint32 upper, const uint32 lower**) [static]

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

- [oscl\\_int64\\_utils.h](#)

## 7.52 Oscl\_Less< T > Struct Template Reference

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

### Public Methods

- bool [operator\(\)](#) (const T &x, const T &y) const

```
template<class T> struct Oscl_Less< T >
```

#### 7.52.1 Member Function Documentation

**7.52.1.1 template<class T> bool Oscl\_Less< T >::operator() (const T & x, const T & y) const [inline]**

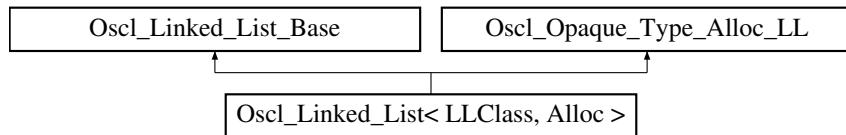
The documentation for this struct was generated from the following file:

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

## 7.53 Oscl\_Linked\_List< LLClass, Alloc > Class Template Reference

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

Inheritance diagram for Oscl\_Linked\_List< LLClass, Alloc >::



### Public Methods

- [Oscl\\_Linked\\_List \(\)](#)
- [~Oscl\\_Linked\\_List \(\)](#)
- 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 Oscl_Linked_List< LLClass, Alloc >`

Oscl Linked List Class

### 7.53.2 Constructor & Destructor Documentation

**7.53.2.1 `template<class LLClass, class Alloc> Oscl_Linked_List< LLClass, Alloc >::Oscl_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 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.

**7.53.3.2 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::add\_to\_front (const LLClass & new\_element) [inline]**

Adds new element at the start of the list.if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

**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 node 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 Oscl\_Linked\_List< LLClass, Alloc >::get\_first (LLClass & ele) [inline]**

Return the first element of list in passed parameter,

**Parameters:**

*ele* return the value of first element of list in this parameter

**Returns:**

32-bit interger,If first element found, it returns 1 otherwise it returns 0

**7.53.3.8 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_index (const LLClass & data) [inline]**

Returns the index for requested element.

**Parameters:**

*data* the element for which index to be return.

**Returns:**

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

**7.53.3.9 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_next (LLClass & ele) [inline]**

Return the next element of list in passed parameter,

**Parameters:**

*ele* return the value of next element of list in this parameter

**Returns:**

32-bit interger ,if next element is found in list,it returns 1 otherwise it returns 0

**7.53.3.10 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_num\_elements () [inline]**

Get number of elements in the list.

**Returns:**

32-bit integer, number of elements in list.

### 7.53.3.11 template<class LLClass, class Alloc> int32 Oscl\_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 Oscl\_Linked\_List< LLClass, Alloc >::move\_to\_end (const LLClass & *data\_to\_move*) [inline]

Moves the element to end of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

### 7.53.3.13 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::move\_to\_front (const LLClass & *data\_to\_move*) [inline]

Moves the element to front of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

### 7.53.3.14 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::remove\_element (const int32 *index\_to\_remove*) [inline]

Removes the element for requested index.

**Parameters:**

*index\_to\_remove*

**Returns:**

on success return 1 otherwise return 0.

Reimplemented from [Oscl\\_Linked\\_List\\_Base](#).

**7.53.3.15 template<class LLClass, class Alloc> int32 Oscl\_Linked\_List< LLClass, Alloc >::remove\_element (const LLClass & *data\_to\_remove*) [inline]**

Removes the element from the list.

**Parameters:**

*data\_to\_remove*

**Returns:**

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

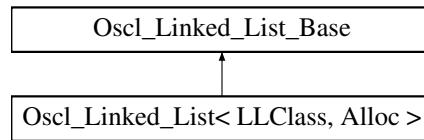
The documentation for this class was generated from the following file:

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

## 7.54 Oscl\_Linked\_List\_Base Class Reference

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

Inheritance diagram for Oscl\_Linked\_List\_Base::



### Protected Methods

- virtual ~Oscl\_Linked\_List\_Base ()
- OSCL\_IMPORT\_REF void construct (Oscl\_Opaque\_Type\_Alloc\_LL \*op)
- OSCL\_IMPORT\_REF void destroy ()
- OSCL\_IMPORT\_REF int32 get\_first (OsclAny \*ele)
- OSCL\_IMPORT\_REF int32 get\_next (OsclAny \*ele)
- OSCL\_IMPORT\_REF int32 check\_list ()
- OSCL\_IMPORT\_REF int32 add\_element (const OsclAny \*new\_element)
- OSCL\_IMPORT\_REF int32 add\_to\_front (const OsclAny \*new\_element)
- OSCL\_IMPORT\_REF int32 insert\_element (const OsclAny \*new\_element, int index)
- OSCL\_IMPORT\_REF int32 get\_element (int32 index, OsclAny \*element)
- OSCL\_IMPORT\_REF int32 remove\_element (const OsclAny \*data\_to\_remove)
- OSCL\_IMPORT\_REF int32 get\_index (const OsclAny \*data)
- OSCL\_IMPORT\_REF int32 remove\_element (const int32 index\_to\_remove)
- OSCL\_IMPORT\_REF int32 move\_to\_end (const OsclAny \*data\_to\_move)
- OSCL\_IMPORT\_REF int32 move\_to\_front (const OsclAny \*data\_to\_move)

### Protected Attributes

- OsclAny \* head
- OsclAny \* tail
- OsclAny \* iterator
- int32 num\_elements
- uint32 sizeof\_T

#### 7.54.1 Detailed Description

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

## 7.54.2 Constructor & Destructor Documentation

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

## 7.54.3 Member Function Documentation

**7.54.3.1** `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::add_element (const OsclAny * new_element)` [protected]

Adds new element to the list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

**7.54.3.2** `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::add_to_front (const OsclAny * new_element)` [protected]

Adds new element at the start of the list.if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

**7.54.3.3** `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::check_list ()` [protected]

Debug routine: Checks the list for elements.

**Returns:**

32-bit integer, if node count is equal to number of node added to the list then returns 1 otherwise returns 0.

Reimplemented in [Oscl\\_Linked\\_List< LLClass, Alloc >](#).

**7.54.3.4** `OSCL_IMPORT_REF void Oscl_Linked_List_Base::construct (Oscl_Opaque_Type_Alloc_LL * op)` [protected]

**7.54.3.5** `OSCL_IMPORT_REF void Oscl_Linked_List_Base::destroy ()` [protected]

**7.54.3.6** `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::get_element (int32 index, OsclAny * element)` [protected]

Search and returns the element in the list for passed index.

**Parameters:**

*index, element* The index is the count for the node.

**Returns:**

32-bit integer on success returns 1 otherwise returns 0.

**7.54.3.7 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::get\_first (OsclAny \* *ele*) [protected]**

Return the first element of list in passed parameter,

**Parameters:**

*ele* return the value of first element of list in this parameter

**Returns:**

32-bit interger,If first element found, it returns 1 otherwise it returns 0

**7.54.3.8 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::get\_index (const OsclAny \* *data*) [protected]**

Returns the index for requested element.

**Parameters:**

*data* the element for which index to be return.

**Returns:**

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

**7.54.3.9 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::get\_next (OsclAny \* *ele*) [protected]**

Return the next element of list in passed parameter,

**Parameters:**

*ele* return the value of next element of list in this parameter

**Returns:**

32-bit interger ,if next element is found in list,it returns 1 otherwise it returns 0

**7.54.3.10 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::insert\_element (const OsclAny \* *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 Oscl\_Linked\_List\_Base::move\_to\_end (const OsclAny \*  
*data\_to\_move*) [protected]**

Moves the element to end of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**7.54.3.12 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::move\_to\_front (const OsclAny \*  
*data\_to\_move*) [protected]**

Moves the element to front of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**7.54.3.13 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::remove\_element (const int32  
*index\_to\_remove*) [protected]**

Removes the element for requested index.

**Parameters:**

*index\_to\_remove*

**Returns:**

on success return 1 otherwise return 0.

Reimplemented in [Oscl\\_Linked\\_List< LLClass, Alloc >](#).

**7.54.3.14 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::remove\_element (const OsclAny \*  
*data\_to\_remove*) [protected]**

Removes the element from the list.

**Parameters:**

*data\_to\_remove*

**Returns:**

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

## 7.54.4 Field Documentation

7.54.4.1 **OsclAny\* Oscl\_Linked\_List\_Base::head** [protected]

7.54.4.2 **OsclAny\* Oscl\_Linked\_List\_Base::iterator** [protected]

7.54.4.3 **int32 Oscl\_Linked\_List\_Base::num\_elements** [protected]

7.54.4.4 **uint32 Oscl\_Linked\_List\_Base::sizeof\_T** [protected]

7.54.4.5 **OsclAny\* Oscl\_Linked\_List\_Base::tail** [protected]

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

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

## 7.55 Oscl\_Map< Key, T, Alloc, Compare > Class Template Reference

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

### Public Types

- `typedef Key key_type`
- `typedef Compare key_compare`
- `typedef Oscl_Pair< const Key, T > value_type`
- `typedef Oscl_Map< Key, T, Alloc, Compare > self`
- `typedef rep_type::pointer pointer`
- `typedef rep_type::reference reference`
- `typedef rep_type::const_reference const_reference`
- `typedef rep_type::iterator iterator`
- `typedef rep_type::const_iterator const_iterator`
- `typedef rep_type::size_type size_type`
- `typedef Oscl_Pair< iterator, bool > pair_iterator_bool`
- `typedef Oscl_Pair< iterator, iterator > pair_iterator_iterator`
- `typedef Oscl_Pair< const_iterator, const_iterator > pair_citerator_citerator`

### Public Methods

- `Oscl_Map (const Compare &comp=Compare())`
- `Oscl_Map (const self &x)`
- `self & operator= (const self &x)`
- `key_compare key_comp () const`
- `value_compare value_comp () const`
- `iterator begin ()`
- `const_iterator begin () const`
- `iterator end ()`
- `const_iterator end () const`
- `bool empty () const`
- `size_type size () const`
- `size_type max_size () const`
- `T & operator[ ] (const key_type &k)`
- `pair_iterator_bool insert (const value_type &x)`
- `iterator insert (iterator position, const value_type &x)`
- `void insert (const value_type *first, const value_type *last)`
- `void erase (iterator position)`
- `size_type erase (const key_type &x)`
- `void erase (iterator first, iterator last)`
- `void clear ()`
- `iterator find (const key_type &x)`
- `const_iterator find (const key_type &x) const`
- `size_type count (const key_type &x) const`
- `iterator lower_bound (const key_type &x)`
- `const_iterator lower_bound (const key_type &x) const`
- `iterator upper_bound (const key_type &x)`

- 
- `const_iterator upper_bound (const key_type &x) const`
  - `pair_iterator iterator equal_range (const key_type &x)`
  - `pair_citerator citerator equal_range (const key_type &x) const`

### 7.55.1 Detailed Description

**template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> class Oscl\_Map< Key, T, Alloc, Compare >**

Oscl\_Map Class. A subset of STL::Map methods. Oscl\_Map is a sorted associative container that associates objects of type Key with objects of type T. It is also a unique associative container, meaning that no two elements have the same key. Oscl\_Map uses the key to speed lookup, insertion, and deletion of elements. It is often superior to all other containers when you need to lookup an element by key value. Due to the underlying tree structure, inserts and erases can be performed in logarithmic time, where a vector would take linear time in some cases.

## 7.55.2 Member Typedef Documentation

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

## 7.55.3 Constructor & Destructor Documentation

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

Creates an empty map using comp as the key compare object

**7.55.3.2 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> Oscl\_Map< Key, T, Alloc, Compare >::Oscl\_Map (const **self** & *x*) [inline]**

Oscl\_Map copy constructor

#### 7.55.4 Member Function Documentation

**7.55.4.1 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::begin () const [inline]**

Returns a const iterator pointing to the beginning of the map

**7.55.4.2 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator Oscl\_Map< Key, T, Alloc, Compare >::begin () [inline]**

Returns an iterator pointing to the beginning of the map

**7.55.4.3 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void Oscl\_Map< Key, T, Alloc, Compare >::clear () [inline]**

Erases all elements

**7.55.4.4 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> size\_type Oscl\_Map< Key, T, Alloc, Compare >::count (const **key\_type** & *x*) const [inline]**

Returns the number of elements with key *x*. For map this will either be 0 or 1.

**7.55.4.5 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> bool Oscl\_Map< Key, T, Alloc, Compare >::empty () const [inline]**

Returns true if map size is 0

**7.55.4.6 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::end () const [inline]**

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

**7.55.4.7 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator Oscl\_Map< Key, T, Alloc, Compare >::end () [inline]**

Returns an iterator pointing to the end of the map.

**7.55.4.8 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> pair\_citerator\_citerator Oscl\_Map< Key, T, Alloc, Compare >::equal\_range (const **key\_type** & *x*) const [inline]**

Finds a range containing all elements whose key is *x*

**7.55.4.9 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
 pair\_iterator iterator Oscl\_Map< Key, T, Alloc, Compare >::equal\_range (const  
 key\_type & x) [inline]**

Finds a range containing all elements whose key is x

**7.55.4.10 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void  
 Oscl\_Map< Key, T, Alloc, Compare >::erase (iterator first, iterator last) [inline]**

Erases all elements in the range [first,last)

**7.55.4.11 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> size\_type  
 Oscl\_Map< Key, T, Alloc, Compare >::erase (const key\_type & x) [inline]**

Erases the element with key x

**7.55.4.12 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void  
 Oscl\_Map< Key, T, Alloc, Compare >::erase (iterator position) [inline]**

Erases the element pointed to by position

**7.55.4.13 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
 const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::find (const key\_type & x) const  
 [inline]**

Finds an element whose key is x

**7.55.4.14 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator  
 Oscl\_Map< Key, T, Alloc, Compare >::find (const key\_type & x) [inline]**

Finds an element whose key is x

**7.55.4.15 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void  
 Oscl\_Map< Key, T, Alloc, Compare >::insert (const value\_type \*first, const value\_type  
 \*last) [inline]**

Inserts the range [first,last) into the map

**7.55.4.16 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator  
 Oscl\_Map< Key, T, Alloc, Compare >::insert (iterator position, const value\_type & x)  
 [inline]**

Inserts x into the map using position as a hint as to where it should be inserted

**7.55.4.17 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
`pair_iterator_bool` Oscl\_Map< Key, T, Alloc, Compare >::insert (const `value_type` & x) [inline]**

Inserts x into the map

**7.55.4.18 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
`key_compare` Oscl\_Map< Key, T, Alloc, Compare >::key\_comp () const [inline]**

Returns the key compare object used by the map

**7.55.4.19 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
`const_iterator` Oscl\_Map< Key, T, Alloc, Compare >::lower\_bound (const `key_type` & x) const [inline]**

Finds the first element whose key is not less than x

**7.55.4.20 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator  
`Oscl_Map`< Key, T, Alloc, Compare >::lower\_bound (const `key_type` & x) [inline]**

Finds the first element whose key is not less than x

**7.55.4.21 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> size\_type  
`Oscl_Map`< Key, T, Alloc, Compare >::max\_size () const [inline]**

Returns the maximum possible size of the map

**7.55.4.22 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> self&  
`Oscl_Map`< Key, T, Alloc, Compare >::operator= (const `self` & x) [inline]**

Oscl\_Map assignment operator

**7.55.4.23 ]**

template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> T& Oscl\_Map< Key, T, Alloc, Compare >::operator[] (const `key_type` & k) [inline]

Returns a reference to the object that is associated with a particular key. If the map does not already contain such an object, operator[] inserts the default object `value_type()`.

**7.55.4.24 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> size\_type  
`Oscl_Map`< Key, T, Alloc, Compare >::size () const [inline]**

Returns the size of the map

**7.55.4.25 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::upper\_bound (const key\_type &  
x) const [inline]**

Finds the first element whose key is not greater than x

**7.55.4.26 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator  
Oscl\_Map< Key, T, Alloc, Compare >::upper\_bound (const key\_type & x) [inline]**

Finds the first element whose key is not greater than x

**7.55.4.27 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
value\_compare Oscl\_Map< Key, T, Alloc, Compare >::value\_comp () const  
[inline]**

Returns the value compare object used by the map

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

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

## 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:

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

## 7.57 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock > Class Template Reference

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

### Public Methods

- [Oscl\\_MTLinked\\_List \(\)](#)
- [~Oscl\\_MTLinked\\_List \(\)](#)
- int32 [dequeue\\_element \(LLClass &element\)](#)
- int32 [add\\_element \(LLClass &new\\_element\)](#)
- int32 [add\\_to\\_front \(LLClass &new\\_element\)](#)
- uint32 [get\\_element \(int32 index, LLClass &element\)](#)
- int32 [remove\\_element \(const LLClass &data\\_to\\_remove\)](#)
- int32 [get\\_index \(const LLClass &data\)](#)
- int32 [remove\\_element \(const int32 index\\_to\\_remove\)](#)
- int32 [move\\_to\\_end \(const LLClass &data\\_to\\_move\)](#)
- int32 [move\\_to\\_front \(const LLClass &data\\_to\\_move\)](#)

### Protected Attributes

- [Oscl\\_Linked\\_List< LLClass, Alloc > the\\_list](#)

#### 7.57.1 Detailed Description

**template<class LLClass, class Alloc, class TheLock> class Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >**

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

#### 7.57.2 Constructor & Destructor Documentation

**7.57.2.1 template<class LLClass, class Alloc, class TheLock> Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::Oscl\_MTLinked\_List () [inline]**

Constructor for Oscl\_MTLinked\_List

**7.57.2.2 template<class LLClass, class Alloc, class TheLock> Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::~Oscl\_MTLinked\_List () [inline]**

Destructor for Oscl\_MTLinked\_List

### 7.57.3 Member Function Documentation

#### 7.57.3.1 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::add\_element (LLClass & *new\_element*) [inline]

Adds new element to the Multi Threaded Linked list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

#### 7.57.3.2 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::add\_to\_front (LLClass & *new\_element*) [inline]

Adds new element at the start of the Multi Threaded Linked list. if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

#### 7.57.3.3 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::dequeue\_element (LLClass & *element*) [inline]

#### 7.57.3.4 template<class LLClass, class Alloc, class TheLock> uint32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::get\_element (int32 *index*, LLClass & *element*) [inline]

Search and returns the element in the Multi Threaded Linked List for passed index.

**Parameters:**

*index, element* The index is the count for the node.

**Returns:**

32-bit integer on success returns 1 otherwise returns 0.

#### 7.57.3.5 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::get\_index (const LLClass & *data*) [inline]

Returns the index for requested element.

**Parameters:**

*data* the element for which index to be return.

**Returns:**

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

**7.57.3.6 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::move\_to\_end (const LLClass & *data\_to\_move*) [inline]**

Moves the element to end of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**7.57.3.7 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::move\_to\_front (const LLClass & *data\_to\_move*) [inline]**

Moves the element to front of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**7.57.3.8 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::remove\_element (const int32 *index\_to\_remove*) [inline]**

Removes the element for requested index.

**Parameters:**

*index\_to\_remove*

**Returns:**

on success return 1 otherwise return 0.

**7.57.3.9 template<class LLClass, class Alloc, class TheLock> int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::remove\_element (const LLClass & *data\_to\_remove*) [inline]**

Removes the element from the list.

**Parameters:**

*data\_to\_remove*

**Returns:**

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

## 7.57.4 Field Documentation

**7.57.4.1 template<class LLClass, class Alloc, class TheLock> Oscl\_Linked\_List<LLClass, Alloc> Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list [protected]**

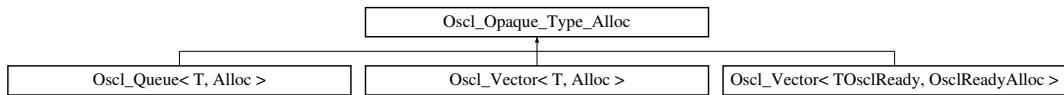
The documentation for this class was generated from the following file:

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

## 7.58 Oscl\_Opaque\_Type\_Alloc Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl\_Opaque\_Type\_Alloc::



### Public Methods

- virtual ~Oscl\_Opaque\_Type\_Alloc ()
- virtual void construct (OsclAny \*p, const OsclAny \*init\_val)=0
- virtual void destroy (OsclAny \*p)=0
- virtual OsclAny \* allocate (const uint32 size)=0
- virtual void deallocate (OsclAny \*p)=0

#### 7.58.1 Detailed Description

This class combines opaque type operations with memory allocation operations.

#### 7.58.2 Constructor & Destructor Documentation

**7.58.2.1 virtual Oscl\_Opaque\_Type\_Alloc::~Oscl\_Opaque\_Type\_Alloc () [inline, virtual]**

#### 7.58.3 Member Function Documentation

**7.58.3.1 virtual OsclAny\* Oscl\_Opaque\_Type\_Alloc::allocate (const uint32 size) [pure virtual]**

Allocate "size" bytes

**7.58.3.2 virtual void Oscl\_Opaque\_Type\_Alloc::construct (OsclAny \* p, const OsclAny \* init\_val) [pure virtual]**

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

**7.58.3.3 virtual void Oscl\_Opaque\_Type\_Alloc::deallocate (OsclAny \* p) [pure virtual]**

Deallocate memory previously allocated with "allocate"

**7.58.3.4 virtual void Oscl\_Opaque\_Type\_Alloc::destroy (OsclAny \* p) [pure virtual]**

Destroy element at p.

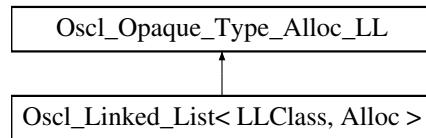
The documentation for this class was generated from the following file:

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

## 7.59 Oscl\_Opaque\_Type\_Alloc\_LL Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl\_Opaque\_Type\_Alloc\_LL::



### Public Methods

- virtual ~Oscl\_Opaque\_Type\_Alloc\_LL ()
- virtual void construct (OsclAny \*p, const OsclAny \*init\_val)=0
- virtual void destroy (OsclAny \*p)=0
- virtual OsclAny \* allocate (const uint32 size)=0
- virtual void deallocate (OsclAny \*p)=0
- virtual OsclAny \* get\_next (const OsclAny \*elem) const=0
- virtual void set\_next (OsclAny \*elem, const OsclAny \*nextelem)=0
- virtual void get\_data (OsclAny \*elem, OsclAny \*data\_val)=0
- virtual bool compare\_data (const OsclAny \*elem, const OsclAny \*data\_val) const=0

#### 7.59.1 Detailed Description

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

#### 7.59.2 Constructor & Destructor Documentation

**7.59.2.1 virtual Oscl\_Opaque\_Type\_Alloc\_LL::~Oscl\_Opaque\_Type\_Alloc\_LL () [inline, virtual]**

#### 7.59.3 Member Function Documentation

**7.59.3.1 virtual OsclAny\* Oscl\_Opaque\_Type\_Alloc\_LL::allocate (const uint32 size) [pure virtual]**

Allocate "size" bytes

**7.59.3.2 virtual bool Oscl\_Opaque\_Type\_Alloc\_LL::compare\_data (const OsclAny \* elem, const OsclAny \* data\_val) const [pure virtual]**

Compare data.

**7.59.3.3 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::construct (OsclAny \* p, const OsclAny \* init\_val) [pure virtual]**

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

**7.59.3.4 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::deallocate (OsclAny \* *p*) [pure virtual]**

Deallocate memory previously allocated with "allocate"

**7.59.3.5 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::destroy (OsclAny \* *p*) [pure virtual]**

Destroy element at p.

**7.59.3.6 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::get\_data (OsclAny \* *elem*, OsclAny \* *data\_val*) [pure virtual]**

Get data

**7.59.3.7 virtual OsclAny\* Oscl\_Opaque\_Type\_Alloc\_LL::get\_next (const OsclAny \* *elem*) const [pure virtual]**

Get next element in linked list.

**7.59.3.8 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::set\_next (OsclAny \* *elem*, const OsclAny \* *nextelem*) [pure virtual]**

Set next element in linked list.

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

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

## 7.60 Oscl\_Opaque\_Type\_Compare Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl\_Opaque\_Type\_Compare::



### Public Methods

- virtual ~Oscl\_Opaque\_Type\_Compare ()
- virtual void swap (OsclAny \*a, const OsclAny \*b)=0
- virtual int compare\_LT (OsclAny \*a, OsclAny \*b) const=0
- virtual int compare\_EQ (const OsclAny \*a, const OsclAny \*b) const=0

#### 7.60.1 Detailed Description

Opaque type operations with swap & comparisons.

#### 7.60.2 Constructor & Destructor Documentation

**7.60.2.1 virtual Oscl\_Opaque\_Type\_Compare::~Oscl\_Opaque\_Type\_Compare () [inline, virtual]**

#### 7.60.3 Member Function Documentation

**7.60.3.1 virtual int Oscl\_Opaque\_Type\_Compare::compare\_EQ (const OsclAny \* a, const OsclAny \* b) const [pure virtual]**

Return a==b.

Implemented in [OsclPriorityQueue< Qelem, Alloc, Container, Compare >](#), [OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\\_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >](#), and [OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >](#).

**7.60.3.2 virtual int Oscl\_Opaque\_Type\_Compare::compare\_LT (OsclAny \* a, OsclAny \* b) const [pure virtual]**

Return a<b.

Implemented in [OsclPriorityQueue< Qelem, Alloc, Container, Compare >](#), [OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\\_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >](#), and [OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >](#).

**7.60.3.3 virtual void Oscl\_Opaque\_Type\_Compare::swap (OsclAny \* a, const OsclAny \* b)  
[pure virtual]**

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

Implemented in [OsclPriorityQueue< Qelem, Alloc, Container, Compare >](#), [OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\\_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >](#), and [OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >](#).

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

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

## 7.61 Oscl\_Pair< T1, T2 > Struct Template Reference

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

### Public Methods

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

### Data Fields

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

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

#### 7.61.1 Constructor & Destructor Documentation

**7.61.1.1 template<class T1, class T2> Oscl\_Pair< T1, T2 >::Oscl\_Pair () [inline]**

**7.61.1.2 template<class T1, class T2> Oscl\_Pair< T1, T2 >::Oscl\_Pair (const T1 & a, const T2 & b) [inline]**

#### 7.61.2 Field Documentation

**7.61.2.1 template<class T1, class T2> T1 Oscl\_Pair< T1, T2 >::first**

**7.61.2.2 template<class T1, class T2> T2 Oscl\_Pair< T1, T2 >::second**

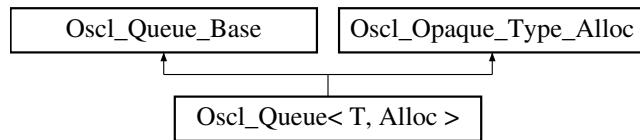
The documentation for this struct was generated from the following file:

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

## 7.62 Oscl\_Queue< T, Alloc > Class Template Reference

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

Inheritance diagram for Oscl\_Queue< T, Alloc >::



### Public Types

- typedef T [value\\_type](#)
- typedef T \* [pointer](#)
- typedef T & [reference](#)
- typedef const T & [const\\_reference](#)
- typedef uint32 [size\\_type](#)

### Public Methods

- [Oscl\\_Queue \(\)](#)
- [Oscl\\_Queue \(uint32 n\)](#)
- virtual [~Oscl\\_Queue \(\)](#)
- void [push \(const T &x\)](#)
- [reference front \(\)](#)
- [const\\_reference front \(\) const](#)
- void [pop \(\)](#)
- [reference back \(\)](#)
- [const\\_reference back \(\) const](#)
- void [clear \(\)](#)

#### 7.62.1 Detailed Description

**template<class T, class Alloc> class Oscl\_Queue< T, Alloc >**

Oscl\_Queue Class. A subset of STL::Queue methods. Oscl\_Queue supports constant time insertion (at the end) and removal of elements at the front of the queue. It does not support insertion or removal of elements at the other ends or middle of the queue, nor random access to elements. \* No iteration capability is [currently] supplied. \* No assignment or copy capability is [currently] supplied. The number of elements in a queue can vary dynamically, and memory management is performed automatically.

## 7.62.2 Member Typedef Documentation

- 7.62.2.1 `template<class T, class Alloc> typedef const T& Oscl_Queue< T, Alloc >::const_reference`
- 7.62.2.2 `template<class T, class Alloc> typedef T* Oscl_Queue< T, Alloc >::pointer`
- 7.62.2.3 `template<class T, class Alloc> typedef T& Oscl_Queue< T, Alloc >::reference`
- 7.62.2.4 `template<class T, class Alloc> typedef uint32 Oscl_Queue< T, Alloc >::size_type`
- 7.62.2.5 `template<class T, class Alloc> typedef T Oscl_Queue< T, Alloc >::value_type`

## 7.62.3 Constructor & Destructor Documentation

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

Creates an empty queue.

- 7.62.3.2 `template<class T, class Alloc> Oscl_Queue< T, Alloc >::Oscl_Queue (uint32 n) [inline]`

Creates an empty queue with capacity n.

### Parameters:

*n* creates a queue with n elements. The main reason for specifying n is efficiency. If you know the capacity to which your queue must grow, then it is more efficient to allocate the queue all at once rather than rely on the automatic reallocation scheme.

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

The destructor.

## 7.62.4 Member Function Documentation

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

Returns the last element (const)

- 7.62.4.2 `template<class T, class Alloc> reference Oscl_Queue< T, Alloc >::back () [inline]`

Returns the last element: "back" (generally not too useful, but some debugging aids might want to find out what was just added)

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

Removes all elements.

Reimplemented from [Oscl\\_Queue\\_Base](#).

**7.62.4.4 template<class T, class Alloc> const\_reference Oscl\_Queue< T, Alloc >::front () const [inline]**

Returns the first element (const)

**7.62.4.5 template<class T, class Alloc> reference Oscl\_Queue< T, Alloc >::front () [inline]**

Returns the first element.

Reimplemented from [Oscl\\_Queue\\_Base](#).

**7.62.4.6 template<class T, class Alloc> void Oscl\_Queue< T, Alloc >::pop () [inline]**

Removes the first element

Reimplemented from [Oscl\\_Queue\\_Base](#).

**7.62.4.7 template<class T, class Alloc> void Oscl\_Queue< T, Alloc >::push (const T & x) [inline]**

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL\_LEAVE will occur

**Parameters:**

*x* new element

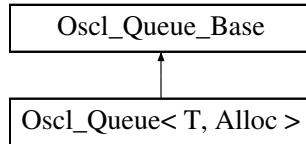
The documentation for this class was generated from the following file:

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

## 7.63 Oscl\_Queue\_Base Class Reference

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

Inheritance diagram for Oscl\_Queue\_Base::



### Public Methods

- uint32 `size () const`
- uint32 `capacity () const`
- bool `empty () const`
- OSCL\_IMPORT\_REF void `reserve (uint32 n)`

### Protected Methods

- OSCL\_IMPORT\_REF void `construct (Oscl_Opaque_Type_Alloc *aType)`
- OSCL\_IMPORT\_REF void `construct (Oscl_Opaque_Type_Alloc *aType, uint32 n)`
- virtual `~Oscl_Queue_Base ()`
- OSCL\_IMPORT\_REF void `destroy ()`
- OSCL\_IMPORT\_REF void `push (const OsclAny *x)`
- OSCL\_IMPORT\_REF void `pop ()`
- OSCL\_IMPORT\_REF void `clear ()`

### Protected Attributes

- uint32 `numelems`
- uint32 `bufsize`
- `OsclAny * elems`
- uint32 `sizeof_T`
- uint32 `ifront`
- uint32 `irear`

#### 7.63.1 Detailed Description

`Oscl_Queue_Base` is a non-templatized base class for [Oscl\\_Queue](#). The purpose of this base class is to avoid large inline routines in the [Oscl\\_Queue](#) implementation. This class is not intended for direct instantiation except by [Oscl\\_Queue](#).

#### 7.63.2 Constructor & Destructor Documentation

**7.63.2.1 virtual Oscl\_Queue\_Base::~Oscl\_Queue\_Base () [inline, protected, virtual]**

The destructor.

### 7.63.3 Member Function Documentation

#### 7.63.3.1 **uint32 Oscl\_Queue\_Base::capacity () const [inline]**

Returns the allocated memory of the queue.

#### 7.63.3.2 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::clear () [protected]**

Removes all elements.

Reimplemented in [Oscl\\_Queue< T, Alloc >](#).

#### 7.63.3.3 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType, uint32 n) [protected]**

#### 7.63.3.4 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType) [protected]**

#### 7.63.3.5 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::destroy () [protected]**

Like an explicit destructor call.

#### 7.63.3.6 **bool Oscl\_Queue\_Base::empty () const [inline]**

True if there are no elements in the queue

#### 7.63.3.7 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::pop () [protected]**

Removes the first element

Reimplemented in [Oscl\\_Queue< T, Alloc >](#).

#### 7.63.3.8 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::push (const OsclAny \* x) [protected]**

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL\_LEAVE will occur

**Parameters:**

*x* new element

#### 7.63.3.9 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::reserve (uint32 n)**

Reallocates memory if necessary to a capacity of *n* elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

**Parameters:**

*n* size of vector

**7.63.3.10 uint32 Oscl\_Queue\_Base::size () const [inline]**

Returns the size of the queue.

#### 7.63.4 Field Documentation

**7.63.4.1 uint32 Oscl\_Queue\_Base::bufsize [protected]****7.63.4.2 OsclAny\* Oscl\_Queue\_Base::elems [protected]****7.63.4.3 uint32 Oscl\_Queue\_Base::ifront [protected]****7.63.4.4 uint32 Oscl\_Queue\_Base::irear [protected]****7.63.4.5 uint32 Oscl\_Queue\_Base::numelems [protected]****7.63.4.6 uint32 Oscl\_Queue\_Base::sizeof\_T [protected]**

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

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

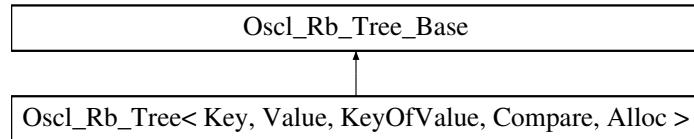


## 7.64 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

### 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](#)



## 7.64 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

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]



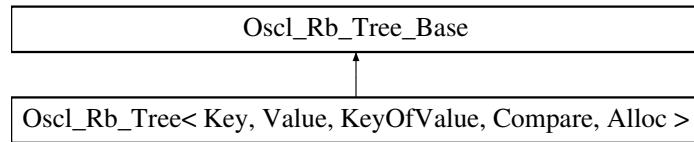
## 7.64 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

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

## 7.65 Oscl\_Rb\_Tree\_Base Class Reference

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

Inheritance diagram for Oscl\_Rb\_Tree\_Base::



### Public Types

- `typedef Oscl_Rb_Tree_Node_Base::base_link_type base_link_type`

### Public Methods

- `OSCL_IMPORT_REF void rotate_left (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF void rotate_right (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF void rebalance (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF base_link_type rebalance_for_erase (base_link_type z, base_link_type &root, base_link_type &leftmost, base_link_type &rightmost)`

#### 7.65.1 Member Typedef Documentation

##### 7.65.1.1 `typedef Oscl_Rb_Tree_Node_Base::base_link_type Oscl_Rb_Tree_Base::base_link_type`

#### 7.65.2 Member Function Documentation

##### 7.65.2.1 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rebalance (base_link_type x, base_link_type &root)`

##### 7.65.2.2 `OSCL_IMPORT_REF base_link_type Oscl_Rb_Tree_Base::rebalance_for_erase (base_link_type z, base_link_type &root, base_link_type &leftmost, base_link_type &rightmost)`

##### 7.65.2.3 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rotate_left (base_link_type x, base_link_type &root)`

##### 7.65.2.4 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rotate_right (base_link_type x, base_link_type &root)`

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

- `oscl_tree.h`

## 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 → () const`
- `bool operator==(const self &x)`
- `bool operator!=(const self &x)`
- `self & operator++()`
- `self operator++(int)`
- `self & operator--()`
- `self operator--(int)`

### Data Fields

- `base_link_type node`

```
template<class Value> struct Oscl_Rb_Tree_Const_Iterator< Value >
```

### 7.66.1 Member Typedef Documentation

- 7.66.1.1 `template<class Value> typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Const_Iterator< Value >::base_link_type`
- 7.66.1.2 `template<class Value> typedef Oscl_Rb_Tree_Const_Iterator<Value> Oscl_Rb_Tree_Const_Iterator< Value >::const_iterator`
- 7.66.1.3 `template<class Value> typedef Oscl_Rb_Tree_Node<Value>* Oscl_Rb_Tree_Const_Iterator< Value >::link_type`
- 7.66.1.4 `template<class Value> typedef const value_type* Oscl_Rb_Tree_Const_Iterator< Value >::pointer`
- 7.66.1.5 `template<class Value> typedef const value_type& Oscl_Rb_Tree_Const_Iterator< Value >::reference`
- 7.66.1.6 `template<class Value> typedef Oscl_Rb_Tree_Const_Iterator<Value> Oscl_Rb_Tree_Const_Iterator< Value >::self`
- 7.66.1.7 `template<class Value> typedef Value Oscl_Rb_Tree_Const_Iterator< Value >::value_type`

### 7.66.2 Constructor & Destructor Documentation

- 7.66.2.1 `template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator () [inline]`
- 7.66.2.2 `template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator (link_type x) [inline]`
- 7.66.2.3 `template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator (const const_iterator & it) [inline]`

### 7.66.3 Member Function Documentation

- 7.66.3.1 `template<class Value> reference Oscl_Rb_Tree_Const_Iterator< Value >::operator * () const [inline]`
- 7.66.3.2 `template<class Value> bool Oscl_Rb_Tree_Const_Iterator< Value >::operator!= (const self & x) [inline]`
- 7.66.3.3 `template<class Value> self Oscl_Rb_Tree_Const_Iterator< Value >::operator++ (int) [inline]`
- 7.66.3.4 `template<class Value> self& Oscl_Rb_Tree_Const_Iterator< Value >::operator++ () [inline]`
- 7.66.3.5 `template<class Value> self Oscl_Rb_Tree_Const_Iterator< Value >::operator- (int) [inline]`
- 7.66.3.6 `template<class Value> self& Oscl_Rb_Tree_Const_Iterator< Value >::operator- () [inline]`

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

## 7.67 Oscl\_Rb\_Tree\_Iterator< Value > Struct Template Reference

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

### Public Types

- `typedef Value value_type`
- `typedef value_type & reference`
- `typedef value_type * pointer`
- `typedef Oscl_Rb_Tree_Iterator< Value > iterator`
- `typedef Oscl_Rb_Tree_Iterator< Value > self`
- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef Oscl_Rb_Tree_Node< Value > * link_type`

### Public Methods

- `Oscl_Rb_Tree_Iterator ()`
- `Oscl_Rb_Tree_Iterator (link_type x)`
- `Oscl_Rb_Tree_Iterator (const iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator== (const self &x)`
- `bool operator!= (const self &x)`
- `self & operator++ ()`
- `self operator++ (int)`
- `self & operator– ()`
- `self operator– (int)`

### Data Fields

- `base_link_type node`

---

```
template<class Value> struct Oscl_Rb_Tree_Iterator< Value >
```

### 7.67.1 Member Typedef Documentation

- 7.67.1.1 template<class Value> typedef Oscl\_Rb\_Tree\_Node\_Base\* Oscl\_Rb\_Tree\_Iterator< Value >::base\_link\_type
- 7.67.1.2 template<class Value> typedef Oscl\_Rb\_Tree\_Iterator<Value> Oscl\_Rb\_Tree\_Iterator< Value >::iterator
- 7.67.1.3 template<class Value> typedef Oscl\_Rb\_Tree\_Node<Value>\* Oscl\_Rb\_Tree\_Iterator< Value >::link\_type
- 7.67.1.4 template<class Value> typedef value\_type\* Oscl\_Rb\_Tree\_Iterator< Value >::pointer
- 7.67.1.5 template<class Value> typedef value\_type& Oscl\_Rb\_Tree\_Iterator< Value >::reference
- 7.67.1.6 template<class Value> typedef Oscl\_Rb\_Tree\_Iterator<Value> Oscl\_Rb\_Tree\_Iterator< Value >::self
- 7.67.1.7 template<class Value> typedef Value Oscl\_Rb\_Tree\_Iterator< Value >::value\_type

### 7.67.2 Constructor & Destructor Documentation

- 7.67.2.1 template<class Value> Oscl\_Rb\_Tree\_Iterator< Value >::Oscl\_Rb\_Tree\_Iterator () [inline]
- 7.67.2.2 template<class Value> Oscl\_Rb\_Tree\_Iterator< Value >::Oscl\_Rb\_Tree\_Iterator (link\_type x) [inline]
- 7.67.2.3 template<class Value> Oscl\_Rb\_Tree\_Iterator< Value >::Oscl\_Rb\_Tree\_Iterator (const iterator & it) [inline]

### 7.67.3 Member Function Documentation

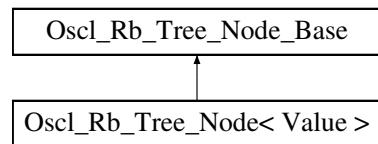
- 7.67.3.1 template<class Value> reference Oscl\_Rb\_Tree\_Iterator< Value >::operator \* () const [inline]
- 7.67.3.2 template<class Value> bool Oscl\_Rb\_Tree\_Iterator< Value >::operator!= (const self & x) [inline]
- 7.67.3.3 template<class Value> self Oscl\_Rb\_Tree\_Iterator< Value >::operator++ (int) [inline]
- 7.67.3.4 template<class Value> self& Oscl\_Rb\_Tree\_Iterator< Value >::operator++ () [inline]
- 7.67.3.5 template<class Value> self Oscl\_Rb\_Tree\_Iterator< Value >::operator- (int) [inline]
- 7.67.3.6 template<class Value> self& Oscl\_Rb\_Tree\_Iterator< Value >::operator- () [inline]
- 7.67.3.7 template<class Value> pointer Oscl\_Rb\_Tree\_Iterator< Value >::operator -> () const [inline]

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

## 7.68 Oscl\_Rb\_Tree\_Node< Value > Struct Template Reference

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

Inheritance diagram for Oscl\_Rb\_Tree\_Node< Value >::



### Public Types

- `typedef Value value_type`
- `typedef Oscl_Rb_Tree_Node< Value > * link_type`

### Data Fields

- `value_type value`

```
template<class Value> struct Oscl_Rb_Tree_Node< Value >
```

#### 7.68.1 Member Typedef Documentation

**7.68.1.1 template<class Value> typedef Oscl\_Rb\_Tree\_Node<Value>\* Oscl\_Rb\_Tree\_Node< Value >::link\_type**

**7.68.1.2 template<class Value> typedef Value Oscl\_Rb\_Tree\_Node< Value >::value\_type**

#### 7.68.2 Field Documentation

**7.68.2.1 template<class Value> value\_type Oscl\_Rb\_Tree\_Node< Value >::value**

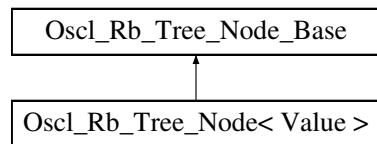
The documentation for this struct was generated from the following file:

- `oscl_tree.h`

## 7.69 Oscl\_Rb\_Tree\_Node\_Base Struct Reference

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

Inheritance diagram for Oscl\_Rb\_Tree\_Node\_Base::



### Public Types

- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef enum Oscl_Rb_Tree_Node_Base::RedBl color_type`
- `enum RedBl { red, black }`

### Static Public Methods

- `base_link_type minimum (base_link_type x)`
- `base_link_type maximum (base_link_type x)`

### Data Fields

- `color_type color`
- `base_link_type parent`
- `base_link_type left`
- `base_link_type right`

#### 7.69.1 Member Typedef Documentation

7.69.1.1 `typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Node_Base::base_link_type`

7.69.1.2 `typedef enum Oscl_Rb_Tree_Node_Base::RedBl Oscl_Rb_Tree_Node_Base::color_type`

#### 7.69.2 Member Enumeration Documentation

7.69.2.1 `enum Oscl_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 Oscl\_Select1st< V, U > Struct Template Reference

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

### Public Methods

- const U & [operator\(\)](#) (const V &x) const

```
template<class V, class U> struct Oscl_Select1st< V, U >
```

#### 7.70.1 Member Function Documentation

**7.70.1.1 template<class V, class U> const U& Oscl\_Select1st< V, U >::operator() (const V & x)  
const [inline]**

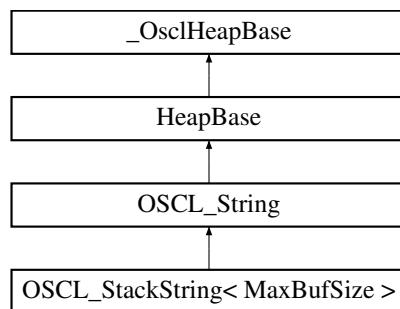
The documentation for this struct was generated from the following file:

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

## 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 TOSCL\_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

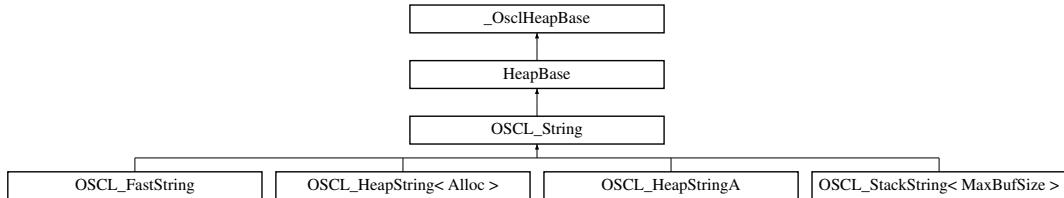
The documentation for this struct was generated from the following file:

- [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< OsclMemAllocator >](#).

### 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< OsclMemAllocator >](#).

**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< OsclMemAllocator >](#).

**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< OsclMemAllocator >](#).

**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< OsclMemAllocator >](#).

**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< OsclMemAllocator >](#).

**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< OsclMemAllocator >](#).

**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.

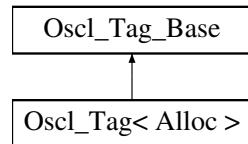
The documentation for this class was generated from the following file:

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

## 7.74 Oscl\_Tag< Alloc > Struct Template Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oscl\_Tag< Alloc >::



### Public Methods

- [Oscl\\_Tag](#) (const Oscl\_Tag< Alloc > &t)
- [Oscl\\_Tag](#) (const [tag\\_base\\_type](#) &t)
- [~Oscl\\_Tag](#) ()
- bool [operator<](#) (const Oscl\_Tag< Alloc > &x) const

### Data Fields

- [Oscl\\_TAlloc< tag\\_base\\_unit, Alloc > tagAllocator](#)
- [tag\\_base\\_type tag](#)

```
template<class Alloc> struct Oscl_Tag< Alloc >
```

#### 7.74.1 Constructor & Destructor Documentation

**7.74.1.1 template<class Alloc> Oscl\_Tag< Alloc >::Oscl\_Tag (const Oscl\_Tag< Alloc > & t)**  
`[inline]`

**7.74.1.2 template<class Alloc> Oscl\_Tag< Alloc >::Oscl\_Tag (const tag\_base\_type & t)**  
`[inline]`

**7.74.1.3 template<class Alloc> Oscl\_Tag< Alloc >::~Oscl\_Tag ()** `[inline]`

#### 7.74.2 Member Function Documentation

**7.74.2.1 template<class Alloc> bool Oscl\_Tag< Alloc >::operator< (const Oscl\_Tag< Alloc > & x) const** `[inline]`

#### 7.74.3 Field Documentation

**7.74.3.1 template<class Alloc> tag\_base\_type Oscl\_Tag< Alloc >::tag**

**7.74.3.2 template<class Alloc> Oscl\_TAlloc<tag\_base\_unit, Alloc> Oscl\_Tag< Alloc >::tagAllocator**

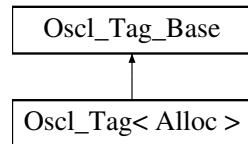
The documentation for this struct was generated from the following file:

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

## 7.75 Oscl\_Tag\_Base Struct Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oscl\_Tag\_Base::



### Public Types

- `typedef char tag_base_unit`
- `typedef tag_base_unit * tag_base_type`
- `typedef uint32 size_type`

### Public Methods

- `bool operator() (const tag_base_type &x, const tag_base_type &y) const`
- `size_type tag_len (const tag_base_type &t) const`
- `tag_base_type tag_copy (tag_base_type &dest, const tag_base_type &src) const`
- `int32 tag_cmp (const tag_base_type &x, const tag_base_type &y) const`
- `OSCL_IMPORT_REF tag_base_type tag_ancestor (tag_base_type &dest, const tag_base_type &src) const`
- `OSCL_IMPORT_REF size_type tag_depth (const tag_base_type &t) const`

### 7.75.1 Member Typedef Documentation

- 7.75.1.1 `typedef uint32 Oscl_Tag_Base::size_type`
- 7.75.1.2 `typedef tag_base_unit* Oscl_Tag_Base::tag_base_type`
- 7.75.1.3 `typedef char Oscl_Tag_Base::tag_base_unit`

### 7.75.2 Member Function Documentation

- 7.75.2.1 `bool Oscl_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 Oscl_Tag_Base::tag_ancestor (tag_base_type & dest, const tag_base_type & src) const`
- 7.75.2.3 `int32 Oscl_Tag_Base::tag_cmp (const tag_base_type & x, const tag_base_type & y) const [inline]`
- 7.75.2.4 `tag_base_type Oscl_Tag_Base::tag_copy (tag_base_type & dest, const tag_base_type & src) const [inline]`
- 7.75.2.5 `OSCL_IMPORT_REF size_type Oscl_Tag_Base::tag_depth (const tag_base_type & t) const`
- 7.75.2.6 `size_type Oscl_Tag_Base::tag_len (const tag_base_type & t) const [inline]`

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

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

## 7.76 Oscl\_TagTree< T, Alloc > Class Template Reference

```
#include <oscl_tagtree.h>
```

### Public Types

- `typedef Oscl_Tag< Alloc > tag_type`
- `typedef tag_type::tag_base_type tag_base_type`
- `typedef Oscl_Vector< Node *, Alloc > children_type`
- `typedef Node node_type`
- `typedef node_type * node_ptr`
- `typedef Oscl_Map< const tag_base_type, node_ptr, Alloc, Oscl_Tag_Base > map_type`
- `typedef map_type::size_type size_type`
- `typedef map_type::value_type value_type`
- `typedef Oscl_Pair< iterator, bool > pair_iterator_bool`

### Public Methods

- `Oscl_TagTree (size_type max_depth=0)`
- `Oscl_TagTree (const Oscl_TagTree< T, Alloc > &x)`
- `Oscl_TagTree< T, Alloc > & operator= (const Oscl_TagTree< T, Alloc > &x)`
- `~Oscl_TagTree ()`
- `iterator begin ()`
- `const_iterator begin () const`
- `iterator end ()`
- `const_iterator end () const`
- `bool empty () const`
- `size_type size () const`
- `T & operator[ ] (const tag_base_type &t)`
- `pair_iterator_bool insert (const tag_base_type &t, const T &x)`
- `void erase (iterator position)`
- `size_type erase (const tag_base_type &x)`
- `void clear ()`
- `iterator find (const tag_base_type &x)`
- `size_type count (const tag_base_type &x) const`

### 7.76.1 Detailed Description

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

Oscl\_TagTree Class.

## 7.76.2 Member Typedef Documentation

- 7.76.2.1 template<class T, class Alloc> typedef **Oscl\_Vector<Node\*, Alloc>** Oscl\_TagTree< T, Alloc >::children\_type
- 7.76.2.2 template<class T, class Alloc> typedef **Oscl\_Map<const tag\_base\_type, node\_ptr, Alloc, Oscl\_Tag\_Base>** Oscl\_TagTree< T, Alloc >::map\_type
- 7.76.2.3 template<class T, class Alloc> typedef **node\_type\*** Oscl\_TagTree< T, Alloc >::node\_ptr
- 7.76.2.4 template<class T, class Alloc> typedef **Node** Oscl\_TagTree< T, Alloc >::node\_type
- 7.76.2.5 template<class T, class Alloc> typedef **Oscl\_Pair<iterator, bool>** Oscl\_TagTree< T, Alloc >::pair\_iterator\_bool
- 7.76.2.6 template<class T, class Alloc> typedef map\_type::size\_type Oscl\_TagTree< T, Alloc >::size\_type
- 7.76.2.7 template<class T, class Alloc> typedef tag\_type::tag\_base\_type Oscl\_TagTree< T, Alloc >::tag\_base\_type
- 7.76.2.8 template<class T, class Alloc> typedef **Oscl\_Tag<Alloc>** Oscl\_TagTree< T, Alloc >::tag\_type
- 7.76.2.9 template<class T, class Alloc> typedef map\_type::value\_type Oscl\_TagTree< T, Alloc >::value\_type

## 7.76.3 Constructor & Destructor Documentation

- 7.76.3.1 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::Oscl\_TagTree (**size\_type max\_depth = 0**) [inline]

Creates a tag tree with only a root node with tag ""

- 7.76.3.2 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::Oscl\_TagTree (const Oscl\_TagTree< T, Alloc > & x) [inline]

Copy constructor

- 7.76.3.3 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >::~Oscl\_TagTree () [inline]

Destructor

## 7.76.4 Member Function Documentation

- 7.76.4.1 template<class T, class Alloc> const\_iterator Oscl\_TagTree< T, Alloc >::begin () const [inline]

Returns an iterator pointing to the first node in the tree.

**7.76.4.2 template<class T, class Alloc> iterator Oscl\_TagTree< T, Alloc >::begin () [inline]**

Returns an iterator pointing to the first node in the tree.

**7.76.4.3 template<class T, class Alloc> void Oscl\_TagTree< T, Alloc >::clear () [inline]**

Erases the entire tag tree.

**7.76.4.4 template<class T, class Alloc> size\_type Oscl\_TagTree< T, Alloc >::count (const tag\_base\_type & x) const [inline]**

Returns the number of elements with key x. This can only be 0 or 1..

**7.76.4.5 template<class T, class Alloc> bool Oscl\_TagTree< T, Alloc >::empty () const [inline]**

Returns true if tree size is 0

**7.76.4.6 template<class T, class Alloc> const\_iterator Oscl\_TagTree< T, Alloc >::end () const [inline]**

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

**7.76.4.7 template<class T, class Alloc> iterator Oscl\_TagTree< T, Alloc >::end () [inline]**

Returns an iterator pointing to the end of the tree.

**7.76.4.8 template<class T, class Alloc> size\_type Oscl\_TagTree< T, Alloc >::erase (const tag\_base\_type & x) [inline]**

Erases the node with tag x. If the node has children, then the node will not be erased from the tree. It will be replaced with the default node value

**Parameters:**

*x* Tag of node to erase

**Returns:**

Returns the number of nodes erased. Since one-to-one mapping between nodes and tags, this will be either 0 or 1

**7.76.4.9 template<class T, class Alloc> void Oscl\_TagTree< T, Alloc >::erase (iterator position) [inline]**

Erases the element pointed to by the iterator. If the node has children, then the node will not be erased from the tree. It will be replaced with the default node value.

**Parameters:**

*position* Iterator pointing to the node to be erased

**7.76.4.10 template<class T, class Alloc> iterator Oscl\_TagTree< T, Alloc >::find (const tag\_base\_type & x) [inline]**

Finds an element whose key is x

**Returns:**

returns an iterator to the element with key x. If no element is found, returns [end\(\)](#)

**7.76.4.11 template<class T, class Alloc> pair\_iterator\_bool Oscl\_TagTree< T, Alloc >::insert (const tag\_base\_type & t, const T & x) [inline]**

Inserts x into the tree and associates it with tag t. If the tag already exists x will not be inserted, and an iterator pointing to the existing node with tag t is returned.

**Parameters:**

*t* tag to use

*x* element to insert

**Returns:**

Returns a pair of parameters, iterator and bool. The iterator points to the inserted node containing x. If the tag t already existed, then the iterator points to the node associated with tag t. The bool is true if x was inserted and false if it was not inserted due to an existing node with tag t.

**7.76.4.12 template<class T, class Alloc> Oscl\_TagTree<T, Alloc>& Oscl\_TagTree< T, Alloc >::operator= (const Oscl\_TagTree< T, Alloc > & x) [inline]**

Assignment operator

**7.76.4.13 ]**

template<class T, class Alloc> T& Oscl\_TagTree< T, Alloc >::operator[] (const tag\_base\_type & t) [inline]

Returns a reference to the object that is associated with a particular tag. If the map does not already contain such an object, operator[] inserts the default object T().

**7.76.4.14 template<class T, class Alloc> size\_type Oscl\_TagTree< T, Alloc >::size () const [inline]**

Returns the number of nodes stored in the tree

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

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

## 7.77 Oscl\_TagTree< T, Alloc >::const\_iterator Struct Reference

```
#include <oscl_tagtree.h>
```

### Public Types

- `typedef const node_type & reference`
- `typedef const node_type * pointer`
- `typedef map_type::const_iterator mapiter`
- `typedef const_iterator self`

### Public Methods

- `const_iterator()`
- `const_iterator(mapiter x)`
- `const_iterator(const const_iterator &it)`
- `reference operator*() const`
- `pointer operator->() const`
- `bool operator==(const self &x)`
- `bool operator!=(const self &x)`
- `self & operator++()`
- `self operator++(int)`
- `self & operator--()`
- `self operator--(int)`

### Data Fields

- `mapiter mapit`

```
template<class T, class Alloc> struct Oscl_TagTree< T, Alloc >::const_iterator
```

### 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> mapiter Oscl_TagTree< T, Alloc >::const_iterator::mapiter`

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

## 7.78 Oscl\_TagTree< T, Alloc >::iterator Struct Reference

```
#include <oscl_tagtree.h>
```

### Public Types

- `typedef node_type & reference`
- `typedef node_type * pointer`
- `typedef map_type::iterator mapiter`
- `typedef iterator self`

### Public Methods

- `iterator ()`
- `iterator (mapiter x)`
- `iterator (const iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator== (const self &x)`
- `bool operator!= (const self &x)`
- `self & operator++ ()`
- `self operator++ (int)`
- `self & operator– ()`
- `self operator– (int)`

### Data Fields

- `mapiter mapit`

template<class T, class Alloc> struct Oscl\_TagTree< T, Alloc >::iterator

### 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 Oscl\_TagTree< T, Alloc >::Node Struct Reference

```
#include <oscl_tagtree.h>
```

### Public Types

- `typedef Oscl_Vector< Node *, Alloc > children_type`

### Public Methods

- `Node()`
- `void sort_children()`
- `tag_type::size_type depth()`

### Data Fields

- `tag_type tag`
- `T value`
- `Node * parent`
- `children_type children`

template<class T, class Alloc> struct Oscl\_TagTree< T, Alloc >::Node

### 7.79.1 Member Typedef Documentation

7.79.1.1 template<class T, class Alloc> typedef Oscl\_Vector<Node\*, Alloc> Oscl\_TagTree< T, Alloc >::Node::children\_type

### 7.79.2 Constructor & Destructor Documentation

7.79.2.1 template<class T, class Alloc> Oscl\_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 Oscl\_TagTree< T, Alloc >::Node::depth () [inline]

7.79.3.2 template<class T, class Alloc> void Oscl\_TagTree< T, Alloc >::Node::sort\_children () [inline]

### 7.79.4 Field Documentation

7.79.4.1 template<class T, class Alloc> children\_type Oscl\_TagTree< T, Alloc >::Node::children

7.79.4.2 template<class T, class Alloc> Node\* Oscl\_TagTree< T, Alloc >::Node::parent

7.79.4.3 template<class T, class Alloc> tag\_type Oscl\_TagTree< T, Alloc >::Node::tag

7.79.4.4 template<class T, class Alloc> T Oscl\_TagTree< T, Alloc >::Node::value

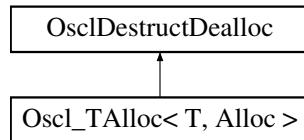
The documentation for this struct was generated from the following file:

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

## 7.80 Oscl\_TAlloc< T, Alloc > Class Template Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oscl\_TAlloc< T, Alloc >::



### Public Types

- typedef T [value\\_type](#)
- typedef T \* [pointer](#)
- typedef const T \* [const\\_pointer](#)
- typedef uint32 [size\\_type](#)
- typedef T & [reference](#)
- typedef const T & [const\\_reference](#)

### Public Methods

- virtual [~Oscl\\_TAlloc \(\)](#)
- [pointer allocate\\_fl \(uint32 size, const char \\*file\\_name, const int line\\_num\)](#)
- [pointer allocate \(uint32 size\)](#)
- [pointer alloc\\_and\\_construct\\_fl \(const\\_reference val, const char \\*file\\_name, const int line\\_num\)](#)
- [pointer alloc\\_and\\_construct \(const\\_reference val\)](#)
- void [deallocate \(OsclAny \\*p\)](#)
- void [deallocate \(OsclAny \\*p, size\\_type n\)](#)
- void [destruct\\_and\\_dealloc \(OsclAny \\*p\)](#)
- [pointer address \(reference r\)](#)
- [const\\_pointer address \(const\\_reference r\) const](#)
- void [construct \(pointer p, const\\_reference val\)](#)
- void [destroy \(pointer p\)](#)

template<class T, class Alloc> class Oscl\_TAlloc< T, Alloc >

### 7.80.1 Member Typedef Documentation

7.80.1.1 template<class T, class Alloc> **typedef const T\* Oscl\_TAlloc< T, Alloc >::const\_pointer**

7.80.1.2 template<class T, class Alloc> **typedef const T& Oscl\_TAlloc< T, Alloc >::const\_reference**

7.80.1.3 template<class T, class Alloc> **typedef T\* Oscl\_TAlloc< T, Alloc >::pointer**

7.80.1.4 template<class T, class Alloc> **typedef T& Oscl\_TAlloc< T, Alloc >::reference**

7.80.1.5 template<class T, class Alloc> **typedef uint32 Oscl\_TAlloc< T, Alloc >::size\_type**

7.80.1.6 template<class T, class Alloc> **typedef T Oscl\_TAlloc< T, Alloc >::value\_type**

### 7.80.2 Constructor & Destructor Documentation

7.80.2.1 template<class T, class Alloc> **virtual Oscl\_TAlloc< T, Alloc >::~Oscl\_TAlloc ()**  
 [inline, virtual]

### 7.80.3 Member Function Documentation

7.80.3.1 template<class T, class Alloc> **const\_pointer Oscl\_TAlloc< T, Alloc >::address (const\_reference r) const** [inline]

7.80.3.2 template<class T, class Alloc> **pointer Oscl\_TAlloc< T, Alloc >::address (reference r)**  
 [inline]

7.80.3.3 template<class T, class Alloc> **pointer Oscl\_TAlloc< T, Alloc >::alloc\_and\_construct (const\_reference val)** [inline]

7.80.3.4 template<class T, class Alloc> **pointer Oscl\_TAlloc< T, Alloc >::alloc\_and\_construct\_fl (const\_reference val, const char \*file\_name, const int line\_num)** [inline]

7.80.3.5 template<class T, class Alloc> **pointer Oscl\_TAlloc< T, Alloc >::allocate (uint32 size)**  
 [inline]

7.80.3.6 template<class T, class Alloc> **pointer Oscl\_TAlloc< T, Alloc >::allocate\_fl (uint32 size, const char \*file\_name, const int line\_num)** [inline]

7.80.3.7 template<class T, class Alloc> **void Oscl\_TAlloc< T, Alloc >::construct (pointer p, const\_reference val)** [inline]

7.80.3.8 template<class T, class Alloc> **void Oscl\_TAlloc< T, Alloc >::deallocate (OsclAny \*p, size\_type n)** [inline]

7.80.3.9 template<class T, class Alloc> **void Oscl\_TAlloc< T, Alloc >::deallocate (OsclAny \*p)**  
 [inline]

7.80.3.10 template<class T, class Alloc> **void Oscl\_TAlloc< T, Alloc >::destroy (pointer p)**  
 [inline]

7.80.3.11 template<class T, class Alloc> **void Oscl\_TAlloc< T, Alloc >::destruct\_and\_dealloc (OsclAny \*p)** [inline, virtual]

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

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

## 7.81 Oscl\_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference

```
#include <oscl_defalloc.h>
```

### Public Types

- `typedef Oscl_TAlloc< U, V > other`

```
template<class T, class Alloc>template<class U, class V> struct Oscl_TAlloc< T, Alloc >::rebind< U, V >
```

#### 7.81.1 Member Typedef Documentation

```
7.81.1.1 template<class T, class Alloc> template<class U, class V> typedef Oscl_TAlloc<U, V>  
Oscl_TAlloc< T, Alloc >::rebind< U, V >::other
```

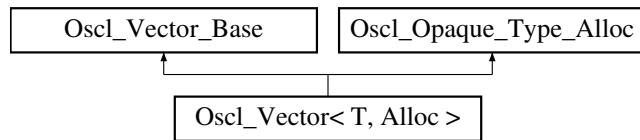
The documentation for this struct was generated from the following file:

- `oscl_defalloc.h`

## 7.82 Oscl\_Vector< T, Alloc > Class Template Reference

```
#include <oscl_vector.h>
```

Inheritance diagram for Oscl\_Vector< T, Alloc >::



### Public Types

- `typedef T value_type`
- `typedef T * pointer`
- `typedef T & reference`
- `typedef const T & const_reference`
- `typedef T * iterator`
- `typedef const T * const_iterator`

### Public Methods

- `Oscl_Vector()`
- `Oscl_Vector(uint32 n)`
- `Oscl_Vector(const Oscl_Vector< T, Alloc > &x)`
- `virtual ~Oscl_Vector()`
- `Oscl_Vector< T, Alloc > & operator=(const Oscl_Vector< T, Alloc > &x)`
- `void push_back(const T &x)`
- `void push_front(const T &x)`
- `iterator insert(iterator pos, const T &x)`
- `T & operator[](uint32 n)`
- `const T & operator[](uint32 n) const`
- `T & front()`
- `const T & front() const`
- `T & back()`
- `const T & back() const`
- `void pop_back()`
- `void clear()`
- `void destroy()`
- `iterator begin() const`
- `iterator end() const`
- `iterator erase(iterator pos)`
- `iterator erase(iterator first, iterator last)`

### 7.82.1 Detailed Description

**template<class T, class Alloc> class Oscl\_Vector< T, Alloc >**

Oscl\_Vector Class. A subset of STL::Vector methods. Oscl\_Vector supports random access to elements, constant time insertion and removal of elements at the end of the vector, and linear time insertion and removal of elements at the beginning or middle of the vector. The number of elements in a vector can vary dynamically, and memory management is performed automatically.

### 7.82.2 Member Typedef Documentation

**7.82.2.1 template<class T, class Alloc> typedef const T\* Oscl\_Vector< T, Alloc >::const\_iterator**

**7.82.2.2 template<class T, class Alloc> typedef const T& Oscl\_Vector< T, Alloc >::const\_reference**

**7.82.2.3 template<class T, class Alloc> typedef T\* Oscl\_Vector< T, Alloc >::iterator**

**7.82.2.4 template<class T, class Alloc> typedef T\* Oscl\_Vector< T, Alloc >::pointer**

**7.82.2.5 template<class T, class Alloc> typedef T& Oscl\_Vector< T, Alloc >::reference**

**7.82.2.6 template<class T, class Alloc> typedef T Oscl\_Vector< T, Alloc >::value\_type**

### 7.82.3 Constructor & Destructor Documentation

**7.82.3.1 template<class T, class Alloc> Oscl\_Vector< T, Alloc >::Oscl\_Vector () [inline]**

Creates an empty vector.

**7.82.3.2 template<class T, class Alloc> Oscl\_Vector< T, Alloc >::Oscl\_Vector (uint32 n) [inline]**

Creates an empty vector with capacity n.

**Parameters:**

*n* creates a vector with n elements. The main reason for specifying n is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

**7.82.3.3 template<class T, class Alloc> Oscl\_Vector< T, Alloc >::Oscl\_Vector (const Oscl\_Vector< T, Alloc > & x) [inline]**

Copy Constructor.

**Parameters:**

*x* vector class to copy.

**7.82.3.4 template<class T, class Alloc> virtual Oscl\_Vector< T, Alloc >::~Oscl\_Vector ()  
[inline, virtual]**

The destructor.

#### 7.82.4 Member Function Documentation

**7.82.4.1 template<class T, class Alloc> const T& Oscl\_Vector< T, Alloc >::back () const  
[inline]**

Returns the last element.

**7.82.4.2 template<class T, class Alloc> T& Oscl\_Vector< T, Alloc >::back () [inline]**

Returns the last element.

**7.82.4.3 template<class T, class Alloc> iterator Oscl\_Vector< T, Alloc >::begin () const  
[inline]**

Returns an iterator pointing to the beginning of the vector.

Reimplemented from [Oscl\\_Vector\\_Base](#).

**7.82.4.4 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::clear () [inline]**

Removes all elements.

**7.82.4.5 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::destroy () [inline]**

Destroy – this is like an explicit destructor call.

Reimplemented from [Oscl\\_Vector\\_Base](#).

**7.82.4.6 template<class T, class Alloc> iterator Oscl\_Vector< T, Alloc >::end () const  
[inline]**

Returns an iterator pointing to the end of the vector..

Reimplemented from [Oscl\\_Vector\\_Base](#).

**7.82.4.7 template<class T, class Alloc> iterator Oscl\_Vector< T, Alloc >::erase (iterator *first*,  
iterator *last*) [inline]**

Erases elements in range [*first*, *last*). Erasing an element invalidates all iterators pointing to elements following the deletion point.

##### Parameters:

*first* starting position

*last* ending position, this position is not erased

**7.82.4.8 template<class T, class Alloc> iterator Oscl\_Vector< T, Alloc >::erase (iterator pos) [inline]**

Erases the element pointed to by iterator pos. Erasing an element invalidates all iterators pointing to elements following the deletion point.

**Parameters:**

*pos* iterator at erase position

**7.82.4.9 template<class T, class Alloc> const T& Oscl\_Vector< T, Alloc >::front () const [inline]**

Returns the first element.

**7.82.4.10 template<class T, class Alloc> T& Oscl\_Vector< T, Alloc >::front () [inline]**

Returns the first element.

**7.82.4.11 template<class T, class Alloc> iterator Oscl\_Vector< T, Alloc >::insert (iterator pos, const T & x) [inline]**

Inserts a new element before the one at pos.

**Parameters:**

*pos* position at which to insert the new element.

*x* new element

**7.82.4.12 template<class T, class Alloc> Oscl\_Vector<T, Alloc>& Oscl\_Vector< T, Alloc >::operator= (const Oscl\_Vector< T, Alloc > & x) [inline]**

The assignment operator

**7.82.4.13 ]**

template<class T, class Alloc> const T& Oscl\_Vector< T, Alloc >::operator[] (uint32 n) const [inline]

Returns the n'th element.

**Parameters:**

*n* element position to return

**7.82.4.14 ]**

template<class T, class Alloc> T& Oscl\_Vector< T, Alloc >::operator[] (uint32 n) [inline]

Returns the n'th element.

**Parameters:**

*n* element position to return

**7.82.4.15 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::pop\_back () [inline]**

Removes the last element.

Reimplemented from [Oscl\\_Vector\\_Base](#).

**7.82.4.16 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::push\_back (const T & x) [inline]**

Inserts a new element at the end. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* new element

**7.82.4.17 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::push\_front (const T & x) [inline]**

Inserts a new element at the front. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* new element

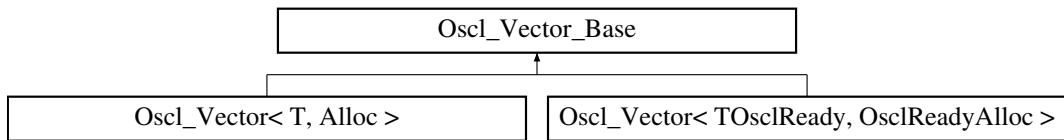
The documentation for this class was generated from the following file:

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

## 7.83 Oscl\_Vector\_Base Class Reference

```
#include <oscl_vector.h>
```

Inheritance diagram for Oscl\_Vector\_Base::



### Public Methods

- uint32 [size \(\) const](#)
- uint32 [capacity \(\) const](#)
- bool [empty \(\) const](#)
- OSCL\_IMPORT\_REF void [reserve \(uint32 n\)](#)

### Protected Methods

- OSCL\_IMPORT\_REF void [construct \(Oscl\\_Opaque\\_Type\\_Alloc \\*aType\)](#)
- OSCL\_IMPORT\_REF void [construct \(Oscl\\_Opaque\\_Type\\_Alloc \\*aType, uint32 n\)](#)
- OSCL\_IMPORT\_REF void [construct \(Oscl\\_Opaque\\_Type\\_Alloc \\*aType, const Oscl\\_Vector\\_Base &x\)](#)
- virtual [~Oscl\\_Vector\\_Base \(\)](#)
- OSCL\_IMPORT\_REF void [push\\_back \(const OsclAny \\*x\)](#)
- OSCL\_IMPORT\_REF void [pop\\_back \(\)](#)
- OSCL\_IMPORT\_REF void [push\\_front \(const OsclAny \\*x\)](#)
- OSCL\_IMPORT\_REF OsclAny \* [insert \(OsclAny \\*pos, const OsclAny \\*x\)](#)
- OSCL\_IMPORT\_REF OsclAny \* [erase \(OsclAny \\*pos\)](#)
- OSCL\_IMPORT\_REF OsclAny \* [erase \(OsclAny \\*first, OsclAny \\*last\)](#)
- OSCL\_IMPORT\_REF void [assign\\_vector \(const Oscl\\_Vector\\_Base &x\)](#)
- OSCL\_IMPORT\_REF void [destroy \(\)](#)

### Protected Attributes

- uint32 [numelems](#)
- uint32 [bufsize](#)
- [OsclAny \\* elems](#)
- uint32 [sizeof\\_T](#)

### Friends

- class [OsclPriorityQueueBase](#)

### 7.83.1 Detailed Description

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

### 7.83.2 Constructor & Destructor Documentation

**7.83.2.1 virtual Oscl\_Vector\_Base::~Oscl\_Vector\_Base () [inline, protected, virtual]**

The destructor.

### 7.83.3 Member Function Documentation

**7.83.3.1 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::assign\_vector (const Oscl\_Vector\_Base & x) [protected]**

**7.83.3.2 uint32 Oscl\_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 Oscl\_Vector\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType, const Oscl\_Vector\_Base & x) [protected]**

**7.83.3.4 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType, uint32 n) [protected]**

**7.83.3.5 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType) [protected]**

**7.83.3.6 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::destroy () [protected]**

Reimplemented in [Oscl\\_Vector< T, Alloc >](#), [Oscl\\_Vector< OsclComponentRegistryElement, OsclMemAllocator >](#), [Oscl\\_Vector< uint32, OsclMemAllocator >](#), [Oscl\\_Vector< OsclSocketServRequestQElem, OsclMemAllocator >](#), [Oscl\\_Vector< Node \\*, Alloc >](#), [Oscl\\_Vector< OsclFixedCacheParam, OsclMemAllocator >](#), [Oscl\\_Vector< OsclSocketRequest \\*, OsclMemAllocator >](#), [Oscl\\_Vector< entry\\_type \\*, Alloc >](#), [Oscl\\_Vector< OSCL\\_HeapString< OsclMemAllocator >, OsclMemAllocator >](#), [Oscl\\_Vector< OsclAsyncFileBuffer \\*, OsclMemAllocator >](#), [Oscl\\_Vector< TOsclFileOffset, OsclMemAllocator >](#), [Oscl\\_Vector< MemPoolBufferInfo \\*, OsclMemAllocator >](#), [Oscl\\_Vector< OsclSharedPtr< PVLoggerFilter >, alloc\\_type >](#), [Oscl\\_Vector< TOsclReady, OsclReadyAlloc >](#), [Oscl\\_Vector< OsclFileCacheBuffer, OsclMemAllocator >](#), [Oscl\\_Vector< OsclSharedPtr< PVLoggerAppender >, alloc\\_type >](#), [Oscl\\_Vector< OsclAny \\*, OsclMemAllocator >](#), and [Oscl\\_Vector< OsclNetworkAddress, OsclMemAllocator >](#).

**7.83.3.7 bool Oscl\_Vector\_Base::empty () const [inline]**

True if the vector's size is 0.

**7.83.3.8 OSCL\_IMPORT\_REF `OsclAny*` Oscl\_Vector\_Base::erase (`OsclAny *first, OsclAny *last`) [protected]**

Erases elements in range [first, last). Erasing an element invalidates all iterators pointing to elements following the deletion point.

**Parameters:**

*first* starting position

*last* ending position, this position is not erased

**7.83.3.9 OSCL\_IMPORT\_REF `OsclAny*` Oscl\_Vector\_Base::erase (`OsclAny *pos`) [protected]**

Erases the element pointed to by iterator pos. Erasing an element invalidates all iterators pointing to elements following the deletion point.

**Parameters:**

*pos* iterator at erase position

**7.83.3.10 OSCL\_IMPORT\_REF `OsclAny*` Oscl\_Vector\_Base::insert (`OsclAny *pos, const OsclAny *x`) [protected]**

Inserts a new element at a specific position.

**Parameters:**

*pos* iterator at insert position.

*x* pointer to new element

**7.83.3.11 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::pop\_back () [protected]**

Removes the last element.

Reimplemented in `Oscl_Vector< T, Alloc >`, `Oscl_Vector< OsclComponentRegistryElement, OsclMemAllocator >`, `Oscl_Vector< uint32, OsclMemAllocator >`, `Oscl_Vector< OsclSocketServRequestQELEM, OsclMemAllocator >`, `Oscl_Vector< Node *, Alloc >`, `Oscl_Vector< OsclFixedCacheParam, OsclMemAllocator >`, `Oscl_Vector< OsclSocketRequest *, OsclMemAllocator >`, `Oscl_Vector< entry_type *, Alloc >`, `Oscl_Vector< OSCL_HeapString< OsclMemAllocator >, OsclMemAllocator >`, `Oscl_Vector< OsclAsyncFileBuffer *, OsclMemAllocator >`, `Oscl_Vector< TOscFileOffset, OsclMemAllocator >`, `Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator >`, `Oscl_Vector< OsclSharedPtr< PVLoggerFilter >, alloc_type >`, `Oscl_Vector< TOscReady, OsclReadyAlloc >`, `Oscl_Vector< OsclFileCacheBuffer, OsclMemAllocator >`, `Oscl_Vector< OsclSharedPtr< PVLoggerAppender >, alloc_type >`, `Oscl_Vector< OsclAny *, OsclMemAllocator >`, and `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >`.

**7.83.3.12 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::push\_back (const `OsclAny *x`) [protected]**

Inserts a new element at the end. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* pointer to the new element

**7.83.3.13 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::push\_front (const OsclAny \* *x*) [protected]**

Inserts a new element at the front. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* pointer to new element

**7.83.3.14 OSCL\_IMPORT\_REF void Oscl\_Vector\_Base::reserve (uint32 *n*)**

Reallocates memory if necessary to a capacity of *n* elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

**Parameters:**

*n* size of vector

**7.83.3.15 uint32 Oscl\_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 OsclPriorityQueueBase [friend]**

## 7.83.5 Field Documentation

**7.83.5.1 uint32 Oscl\_Vector\_Base::bufsize [protected]****7.83.5.2 OsclAny\* Oscl\_Vector\_Base::elems [protected]****7.83.5.3 uint32 Oscl\_Vector\_Base::numelems [protected]****7.83.5.4 uint32 Oscl\_Vector\_Base::sizeof\_T [protected]**

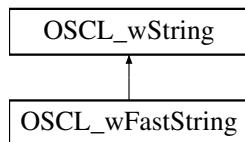
The documentation for this class was generated from the following file:

- [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 \* cstr)**

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 \* cstr)**

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 \* cstr, 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]**

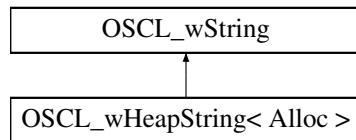
The documentation for this class was generated from the following file:

- [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 otype`
- `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, otype op)`
- `void set(const other_chartype *buf, uint32 length, otype 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]**

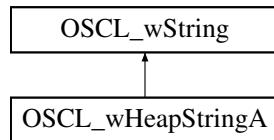
The documentation for this class was generated from the following file:

- [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(Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wHeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wHeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wString &src, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const chartype *cstr, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const chartype *buf, uint32 length, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF ~OSCL_wHeapStringA()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const OSCL_wHeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(const chartype *buf, uint32 length)`
- `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, OsclRefCounter *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, OsclRefCounter *ref = NULL)`

#### 7.86.3.5 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA(const OSCL_wString &src, Oscl_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

#### 7.86.3.6 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA(const chartype *cstr, Oscl_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

#### 7.86.3.7 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA(const chartype *buf, uint32 length, Oscl_DefAlloc *alloc = NULL, OsclRefCounter *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]**

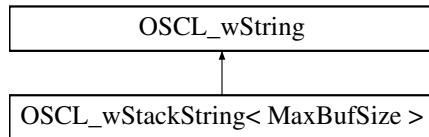
The documentation for this class was generated from the following file:

- [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]**

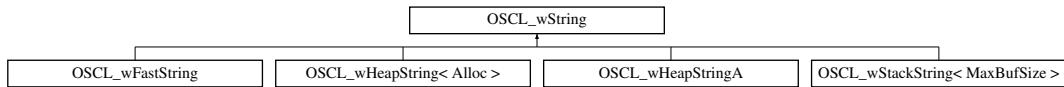
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 7.88 OSCL\_wString Class Reference

```
#include <oscl_string.h>
```

Inheritance diagram for OSCL\_wString::



### Public Types

- `typedef oscl_wchar 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_wString & operator=(const OSCL_wString &src)`
- `OSCL_wString & operator=(const chartype *cstr)`
- `OSCL_wString & operator+=(const OSCL_wString &src)`
- `OSCL_wString & operator+=(const chartype *cstr)`
- `OSCL_wString & operator+=(const chartype 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 chartype *cstr) const`
- `chartype operator[](uint32 index) const`
- `virtual chartype 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, chartype c)`
- `virtual void write(uint32 offset, uint32 length, const chartype *buf)`

### Protected Methods

- `OSCL_wString()`
- `virtual ~OSCL_wString()`
- `virtual void set_rep(const chartype *cstr)=0`
- `virtual void append_rep(const chartype *cstr)=0`
- `virtual void set_rep(const OSCL_wString &src)=0`
- `virtual void append_rep(const OSCL_wString &src)=0`
- `virtual void set_len(uint32 len)=0`

### 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 chartype** 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 chartype \* cstr**) [protected, pure virtual]
- 7.88.4.26 **virtual uint32** OSCL\_wString::setrep\_to\_wide\_char (**const char \* src**, **uint32 len**, **TOSCL\_wStringOp op**, **OscI\_DefAlloc \* aAlloc**) [virtual]
- 7.88.4.27 **virtual void** OSCL\_wString::write (**uint32 offset**, **uint32 length**, **const chartype \* buf**) [virtual]
- 7.88.4.28 **virtual void** OSCL\_wString::write (**uint32 index**, **chartype c**) [virtual]

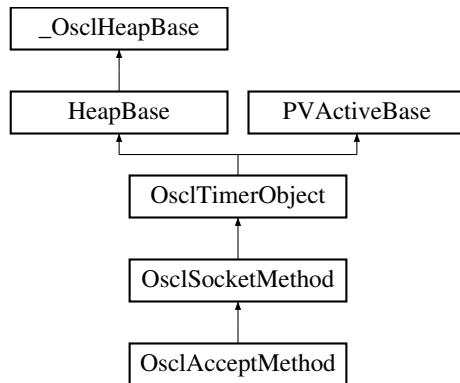
The documentation for this class was generated from the following file:

- [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]

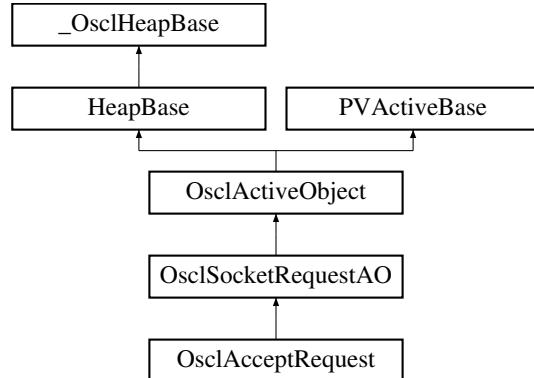
The documentation for this class was generated from the following file:

- [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](#))

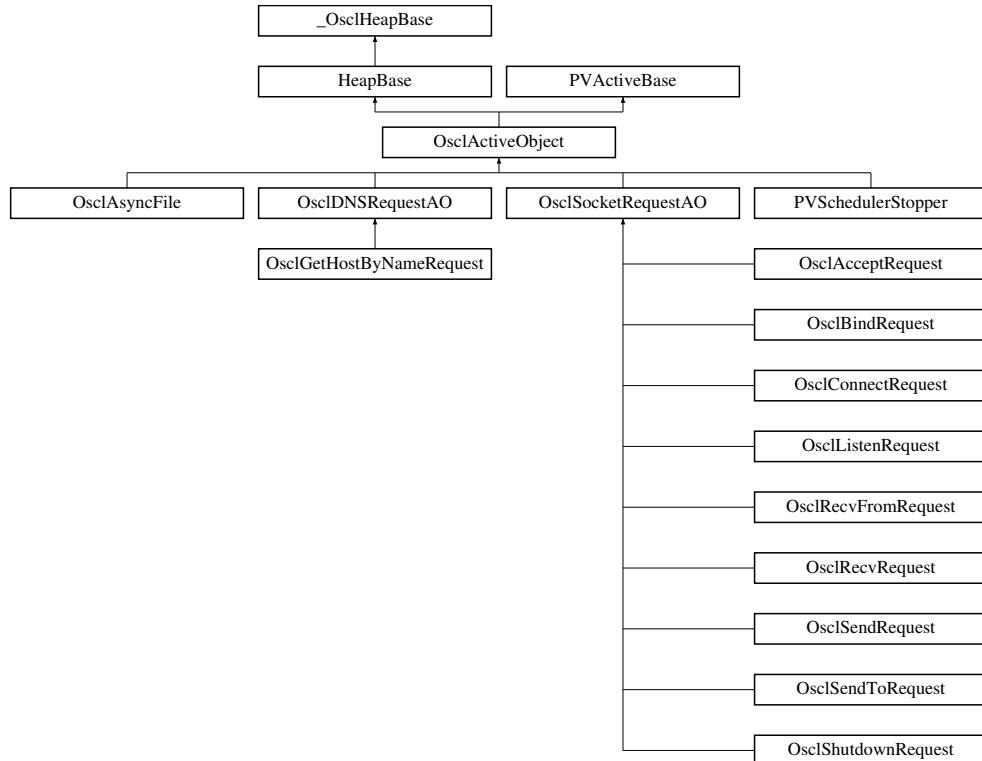
The documentation for this class was generated from the following file:

- [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 [PVActiveBase](#).

#### 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 [PVActiveBase](#).

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 [PVActiveBase](#).

**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 OsclAOStatus& OsclActiveObject::StatusRef ()**

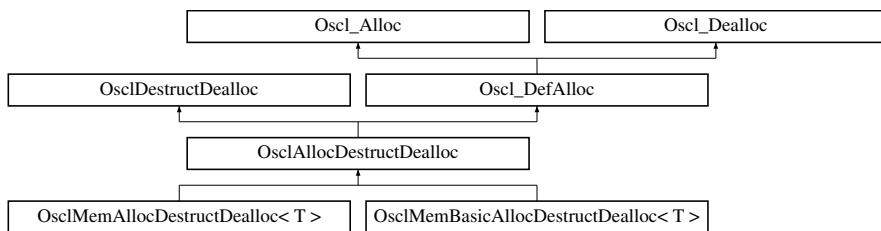
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_ao.h](#)

## 7.92 OsclAllocDestructDealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for OsclAllocDestructDealloc::



### Public Methods

- virtual ~OsclAllocDestructDealloc ()

#### 7.92.1 Constructor & Destructor Documentation

**7.92.1.1 virtual OsclAllocDestructDealloc::~OsclAllocDestructDealloc () [inline, virtual]**

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

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

## 7.93 OsclAOStatus Class Reference

```
#include <oscl_aostatus.h>
```

### Public Methods

- OSCL\_INLINE OsclAOStatus ()
- OSCL\_INLINE OsclAOStatus (int32 aStatus)
- OSCL\_INLINE int32 operator= (int32 aStatus)
- OSCL\_INLINE int32 operator== (int32 aStatus) const
- OSCL\_INLINE int32 operator!= (int32 aStatus) const
- OSCL\_INLINE int32 operator>= (int32 aStatus) const
- OSCL\_INLINE int32 operator<= (int32 aStatus) const
- OSCL\_INLINE int32 operator> (int32 aStatus) const
- OSCL\_INLINE int32 operator< (int32 aStatus) const
- OSCL\_INLINE int32 Value () const

#### 7.93.1 Constructor & Destructor Documentation

7.93.1.1 OSCL\_INLINE OsclAOStatus::OsclAOStatus ()

7.93.1.2 OSCL\_INLINE OsclAOStatus::OsclAOStatus (int32 *aStatus*)

#### 7.93.2 Member Function Documentation

7.93.2.1 OSCL\_INLINE int32 OsclAOStatus::operator!= (int32 *aStatus*) const

7.93.2.2 OSCL\_INLINE int32 OsclAOStatus::operator< (int32 *aStatus*) const

7.93.2.3 OSCL\_INLINE int32 OsclAOStatus::operator<= (int32 *aStatus*) const

7.93.2.4 OSCL\_INLINE int32 OsclAOStatus::operator= (int32 *aStatus*)

7.93.2.5 OSCL\_INLINE int32 OsclAOStatus::operator== (int32 *aStatus*) const

7.93.2.6 OSCL\_INLINE int32 OsclAOStatus::operator> (int32 *aStatus*) const

7.93.2.7 OSCL\_INLINE int32 OsclAOStatus::operator>= (int32 *aStatus*) const

7.93.2.8 OSCL\_INLINE int32 OsclAOStatus::Value ()

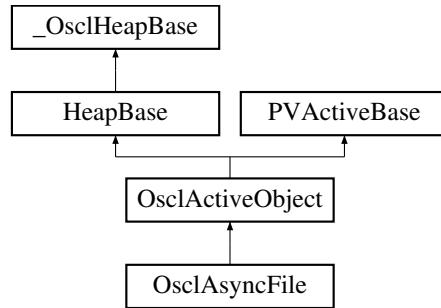
The documentation for this class was generated from the following file:

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

## 7.94 OsclAsyncFile Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclAsyncFile::



### Public Methods

- [`~OsclAsyncFile \(\)`](#)
- [`int32 Open \(const oscl\_wchar \*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\_FileServer &fileserv\)`](#)
- [`int32 Open \(const char \*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\_FileServer &fileserv\)`](#)
- [`int32 Seek \(TOsclFileOffset offset, Oscl\_File::seek\_type origin\)`](#)
- [`TOsclFileOffset Tell \(\)`](#)
- [`uint32 Read \(OsclAny \*aBuffer1, uint32 aDataSize, uint32 aNumElements\)`](#)
- [`int32 EndOfFile \(\)`](#)
- [`TOsclFileOffset Size \(\)`](#)
- [`int32 Close \(\)`](#)
- [`uint32 Write \(const OsclAny \*aBuffer1, uint32 aDataSize, uint32 aNumElements\)`](#)
- [`uint32 Flush \(\)`](#)

### Static Public Methods

- [`OsclAsyncFile \* NewL \(OsclNativeFile &aAsyncFile, int32 aCacheSize, PVLogger \*\)`](#)
- [`void Delete \(OsclAsyncFile \*\)`](#)

### Data Fields

- [`uint32 iNumOfRun`](#)
- [`uint32 iNumOfRunErr`](#)

#### 7.94.1 Detailed Description

OsclAsyncFile

## 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 & aSyncFile, int32 aCacheSize, PVLogger \*) [static]

Two-phased constructor.

#### Parameters:

*aSyncFile*: open handle for async file read. Note: it is the caller's job to open/close this file handle.

*aSyncFile*: duplicate open handle for sync file read. Note: it is the caller's job to open this file handle, but this class will close the handle.

*aCacheSize*: size of one of the individual cache buffers. The total cached data size will be larger, since multiple buffers are used.

*aStartAsyncRead*: When true, async file read will start immediately. When false, read will not begin until StartAsyncRead is called.

- 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

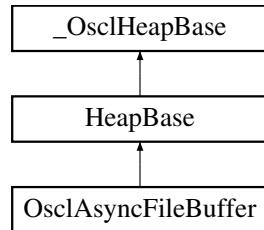
The documentation for this class was generated from the following file:

- [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**

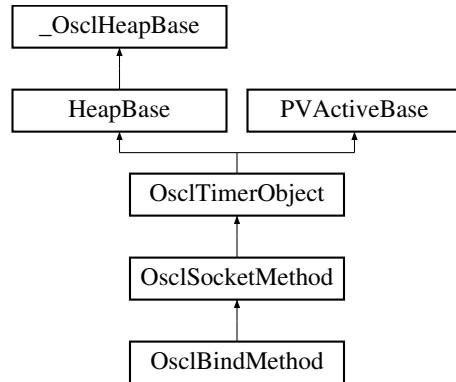
The documentation for this class was generated from the following file:

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

## 7.97 OsclBindMethod Class Reference

```
#include <oscl_socket_bind.h>
```

Inheritance diagram for OsclBindMethod::



### Public Methods

- [~OsclBindMethod \(\)](#)
- [TPVSocketEvent Bind \(OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsclBindRequest \\* BindRequest \(\)](#)

### Static Public Methods

- [OsclBindMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 7.97.1 Constructor & Destructor Documentation

##### 7.97.1.1 OsclBindMethod::~OsclBindMethod ()

#### 7.97.2 Member Function Documentation

##### 7.97.2.1 TPVSocketEvent OsclBindMethod::Bind (OsclNetworkAddress & aAddress, int32 aTimeout)

##### 7.97.2.2 OsclBindRequest\* OsclBindMethod::BindRequest () [inline]

##### 7.97.2.3 OsclBindMethod\* OsclBindMethod::NewL (OsclIPSocketI &c) [static]

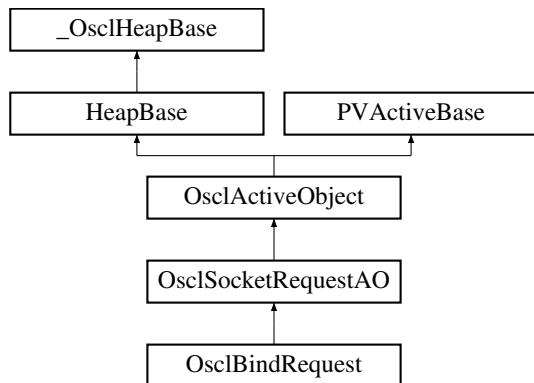
The documentation for this class was generated from the following file:

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

## 7.98 OsclBindRequest Class Reference

```
#include <oscl_socket_bind.h>
```

Inheritance diagram for OsclBindRequest::



### Public Methods

- [OsclBindRequest \(OsclSocketMethod &c\)](#)
- [void Bind \(OsclNetworkAddress &aAddress\)](#)

#### 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 OsclBindRequest::OsclBindRequest ([OsclSocketMethod & c](#)) [inline]

#### 7.98.3 Member Function Documentation

##### 7.98.3.1 void OsclBindRequest::Bind ([OsclNetworkAddress & aAddress](#))

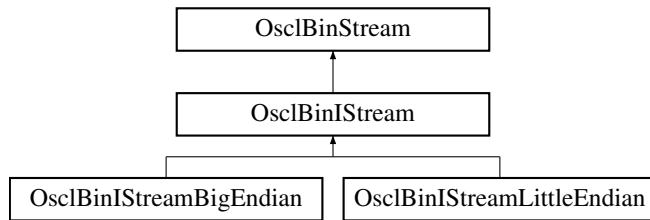
The documentation for this class was generated from the following file:

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

## 7.99 OsclBinIStream Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStream::



### Public Methods

- [OsclBinIStream \(\)](#)
- [~OsclBinIStream \(\)](#)
- uint8 [Read\\_uint8 \(\)](#)

*This method reads an unsigned short from the stream.*

- OsclBinIStream & [get \(int8 \\*data, int32 size\)](#)

*This method reads 'length' number of bytes from the stream and places them in 'data'.*

#### 7.99.1 Constructor & Destructor Documentation

**7.99.1.1 OsclBinIStream::OsclBinIStream () [inline]**

**7.99.1.2 OsclBinIStream::~OsclBinIStream () [inline]**

#### 7.99.2 Member Function Documentation

**7.99.2.1 OsclBinIStream& OsclBinIStream::get (int8 \* data, int32 size)**

This method reads 'length' number of bytes from the stream and places them in 'data'.

**Parameters:**

*data* is a pointer to the place to store the bytes read

*size* is the number of bytes to read

**7.99.2.2 uint8 OsclBinIStream::Read\_uint8 ()**

This method reads an unsigned short from the stream.

**Returns:**

Unsigned short read from the stream.

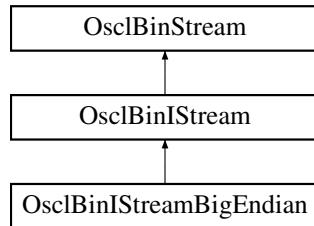
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 7.100 OsclBinIStreamBigEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStreamBigEndian::



### Public Methods

- [OsclBinIStreamBigEndian \(\)](#)
- [void Read \(int8 &data\)](#)
- [void Read \(uint8 &data\)](#)
- [void Read \(int16 &data\)](#)
- [void Read \(uint16 &data\)](#)
- [void Read \(int32 &data\)](#)
- [void Read \(uint32 &data\)](#)
- [OsclBinIStreamBigEndian & operator>> \(int8 &data\)](#)

*This method reads a int8 from the stream and stores it in 'data'.*

- [OsclBinIStream & operator>> \(uint8 &data\)](#)

*This method reads a uint8 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(int16 &data\)](#)

*This method reads a int16 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(uint16 &data\)](#)

*This method reads a uint16 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(int32 &data\)](#)

*This method reads a int32 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(uint32 &data\)](#)

*This method reads a uint32 from the stream and stores it in 'data'.*

- [uint16 Read\\_uint16 \(\)](#)

*This method reads an unsigned short from the stream.*

- [uint32 Read\\_uint32 \(\)](#)

*This method reads an unsigned long from the stream.*

## 7.100.1 Constructor & Destructor Documentation

**7.100.1.1 OsclBinIStreamBigEndian::OsclBinIStreamBigEndian () [inline]**

## 7.100.2 Member Function Documentation

**7.100.2.1 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (uint32 & data)**

This method reads a uint32 from the stream and stores it in 'data'.

**7.100.2.2 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int32 & data)**

This method reads a int32 from the stream and stores it in 'data'.

**7.100.2.3 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (uint16 & data)**

This method reads a uint16 from the stream and stores it in 'data'.

**7.100.2.4 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int16 & data)**

This method reads a int16 from the stream and stores it in 'data'.

**7.100.2.5 OsclBinIStream& OsclBinIStreamBigEndian::operator>> (uint8 & data)**

This method reads a uint8 from the stream and stores it in 'data'.

**7.100.2.6 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int8 & data)**

This method reads a int8 from the stream and stores it in 'data'.

**7.100.2.7 void OsclBinIStreamBigEndian::Read (uint32 & data)**

**7.100.2.8 void OsclBinIStreamBigEndian::Read (int32 & data)**

**7.100.2.9 void OsclBinIStreamBigEndian::Read (uint16 & data)**

**7.100.2.10 void OsclBinIStreamBigEndian::Read (int16 & data)**

**7.100.2.11 void OsclBinIStreamBigEndian::Read (uint8 & data)**

**7.100.2.12 void OsclBinIStreamBigEndian::Read (int8 & data)**

**7.100.2.13 uint16 OsclBinIStreamBigEndian::Read\_uint16 ()**

This method reads an unsigned short from the stream.

### Returns:

Unsigned short read from the stream.

### 7.100.2.14 uint32 OsclBinIStreamBigEndian::Read\_uint32 ()

This method reads an unsigned long from the stream.

**Returns:**

unsigned long read from the stream.

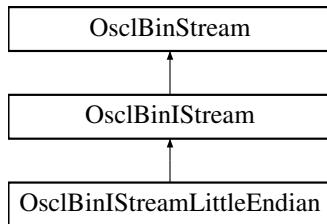
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 7.101 OsclBinIStreamLittleEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStreamLittleEndian::



### Public Methods

- [OsclBinIStreamLittleEndian \(\)](#)  
[OsclBinIStreamLittleEndian & operator>> \(int8 &data\)](#)  
*This method reads a int8 from the stream and stores it in 'data'.*
- [OsclBinIStreamLittleEndian & operator>> \(uint8 &data\)](#)  
*This method reads a uint8 from the stream and stores it in 'data'.*
- [OsclBinIStreamLittleEndian & operator>> \(int16 &data\)](#)  
*This method reads a int16 from the stream and stores it in 'data'.*
- [OsclBinIStreamLittleEndian & operator>> \(uint16 &data\)](#)  
*This method reads a uint16 from the stream and stores it in 'data'.*
- [OsclBinIStreamLittleEndian & operator>> \(int32 &data\)](#)  
*This method reads a int32 from the stream and stores it in 'data'.*
- [OsclBinIStreamLittleEndian & operator>> \(uint32 &data\)](#)  
*This method reads a uint32 from the stream and stores it in 'data'.*

### Protected Methods

- uint16 [Read\\_uint16 \(\)](#)
- uint32 [Read\\_uint32 \(\)](#)

### 7.101.1 Constructor & Destructor Documentation

**7.101.1.1 OsclBinIStreamLittleEndian::OsclBinIStreamLittleEndian () [inline]**

### 7.101.2 Member Function Documentation

**7.101.2.1 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint32 & data)**

This method reads a uint32 from the stream and stores it in 'data'.

**7.101.2.2 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int32 & data)**

This method reads a int32 from the stream and stores it in 'data'.

**7.101.2.3 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint16 & data)**

This method reads a uint16 from the stream and stores it in 'data'.

**7.101.2.4 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int16 & data)**

This method reads a int16 from the stream and stores it in 'data'.

**7.101.2.5 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint8 & data)**

This method reads a uint8 from the stream and stores it in 'data'.

**7.101.2.6 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int8 & data)**

This method reads a int8 from the stream and stores it in 'data'.

**7.101.2.7 uint16 OsclBinIStreamLittleEndian::Read\_uint16 () [protected]**

**7.101.2.8 uint32 OsclBinIStreamLittleEndian::Read\_uint32 () [protected]**

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

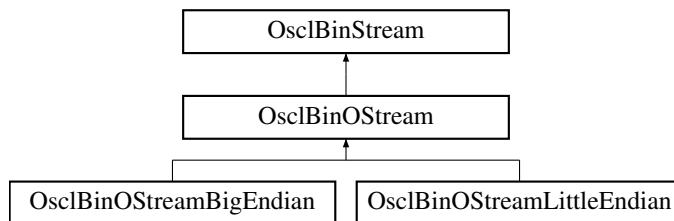
- [oscl\\_bin\\_stream.h](#)

## 7.102 OsclBinOStream Class Reference

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

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStream:::



### Public Methods

- [OsclBinOStream \(\)](#)
- [virtual ~OsclBinOStream \(\)](#)
- [OsclBinOStream & write \(const int8 \\*data, int32 size\)](#)

*This method writes 'length' number of bytes stored in 'data' to the stream.*

#### 7.102.1 Detailed Description

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

#### 7.102.2 Constructor & Destructor Documentation

**7.102.2.1 [OsclBinOStream::OsclBinOStream \(\) \[inline\]](#)**

**7.102.2.2 [virtual OsclBinOStream::~OsclBinOStream \(\) \[inline, virtual\]](#)**

#### 7.102.3 Member Function Documentation

**7.102.3.1 [OsclBinOStream& OsclBinOStream::write \(const int8 \\* data, int32 size\)](#)**

This method writes 'length' number of bytes stored in 'data' to the stream.

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

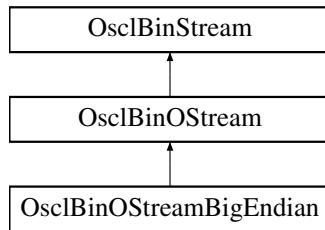
- [oscl\\_bin\\_stream.h](#)

## 7.103 OsclBinOStreamBigEndian Class Reference

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

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStreamBigEndian::



### Public Methods

- [OsclBinOStreamBigEndian \(\)](#)
- OsclBinOStreamBigEndian & [operator<< \(const int8 &data\)](#)  
*This method writes a int8 from 'data' to the stream.*
- OsclBinOStreamBigEndian & [operator<< \(const uint8 &data\)](#)  
*This method writes a uint8 from 'data' to the stream.*
- OsclBinOStreamBigEndian & [operator<< \(const int16 &data\)](#)  
*This method writes a int16 from 'data' to the stream.*
- OsclBinOStreamBigEndian & [operator<< \(const uint16 &data\)](#)  
*This method writes a uint16 from 'data' to the stream.*
- OsclBinOStreamBigEndian & [operator<< \(const int32 &data\)](#)  
*This method writes a int32 from 'data' to the stream.*
- OsclBinOStreamBigEndian & [operator<< \(const uint32 &data\)](#)  
*This method writes a uint32 from 'data' to the stream.*

### Protected Methods

- void [WriteUnsignedShort \(const uint16 data\)](#)
- void [WriteUnsignedLong \(const uint32 data\)](#)

#### 7.103.1 Detailed Description

Class OsclBinOStreamBigEndian 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:

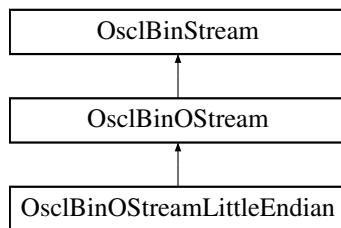
- [oscl\\_bin\\_stream.h](#)

## 7.104 OsclBinOStreamLittleEndian Class Reference

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

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStreamLittleEndian::



### Public Methods

- [OsclBinOStreamLittleEndian \(\)](#)
- [OsclBinOStreamLittleEndian & operator<< \(const int8 &data\)](#)  
*This method writes a int8 from 'data' to the stream.*
- [OsclBinOStreamLittleEndian & operator<< \(const uint8 &data\)](#)  
*This method writes a uint8 from 'data' to the stream.*
- [OsclBinOStreamLittleEndian & operator<< \(const int16 &data\)](#)  
*This method writes a int16 from 'data' to the stream.*
- [OsclBinOStreamLittleEndian & operator<< \(const uint16 &data\)](#)  
*This method writes a uint16 from 'data' to the stream.*
- [OsclBinOStreamLittleEndian & operator<< \(const int32 &data\)](#)  
*This method writes a int32 from 'data' to the stream.*
- [OsclBinOStreamLittleEndian & operator<< \(const uint32 &data\)](#)  
*This method writes a uint32 from 'data' to the stream.*

### Protected Methods

- [void WriteUnsignedShort \(const uint16 data\)](#)  
*This method writes 'data' (unsigned short) to the stream.*
- [void WriteUnsignedLong \(const uint32 data\)](#)  
*This method writes 'data' (unsigned long) to the stream.*

### 7.104.1 Detailed Description

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

## 7.104.2 Constructor & Destructor Documentation

**7.104.2.1 OsclBinOStreamLittleEndian::OsclBinOStreamLittleEndian () [inline]**

## 7.104.3 Member Function Documentation

**7.104.3.1 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint32 & data)**

This method writes a uint32 from 'data' to the stream.

**7.104.3.2 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int32 & data)**

This method writes a int32 from 'data' to the stream.

**7.104.3.3 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint16 & data)**

This method writes a uint16 from 'data' to the stream.

**7.104.3.4 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int16 & data)**

This method writes a int16 from 'data' to the stream.

**7.104.3.5 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint8 & data)**

This method writes a uint8 from 'data' to the stream.

**7.104.3.6 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int8 & data)**

This method writes a int8 from 'data' to the stream.

**7.104.3.7 void OsclBinOStreamLittleEndian::WriteUnsignedLong (const uint32 data) [protected]**

This method writes 'data' (unsigned long) to the stream.

**7.104.3.8 void OsclBinOStreamLittleEndian::WriteUnsignedShort (const uint16 data) [protected]**

This method writes 'data' (unsigned short) to the stream.

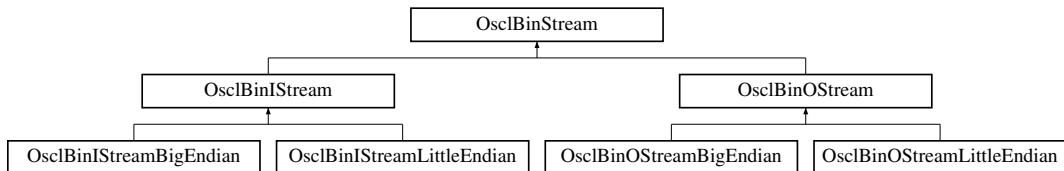
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 7.105 OsclBinStream Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinStream::



### Public Methods

- [OsclBinStream \(\)](#)
- [bool good \(\)](#)

*This method determines if the stream is ok.*

- [bool eof \(\)](#)

*This method determines if end of stream has been reached.*

- [bool fail \(\)](#)

*This method determines if an error has occurred in the stream.*

- [void Attach \(void \\*buffer, uint32 l\\_length\)](#)

*This method specifies the data buffer to attach to the stream.*

- [void Attach \(const uint32 numFragments, const OsclMemoryFragment \\*fragPtr\)](#)

*This method specifies the memory fragment array to use for input.*

- [uint32 tellg \(\)](#)

*This method returns the current stream position.*

- [void Seek \(uint32 absPosition\)](#)

*This method seeks to the specified stream position.*

- [uint32 PositionInBlock \(\)](#)

*This method returns the current stream position.*

- [void seekFromCurrentPosition \(int32 offset\)](#)

*This method seeks to the specified offset from the current location.*

### Protected Types

- enum [state\\_t](#) { [GOOD\\_STATE](#), [EOF\\_STATE](#), [FAIL\\_STATE](#) }

## Protected Methods

- bool [ReserveSpace](#) (uint32 size)
- bool [HaveRoomInCurrentBlock](#) (uint32 size)

## Protected Attributes

- [state\\_t state](#)
- uint8 \* [pBasePosition](#)
- uint8 \* [pPosition](#)
- uint32 [length](#)
- const [OsclMemoryFragment](#) \* [nextFragPtr](#)
- int [fragsLeft](#)
- const [OsclMemoryFragment](#) \* [firstFragPtr](#)
- int [numFrags](#)
- [OsclMemoryFragment](#) [specialFragBuffer](#)

### 7.105.1 Member Enumeration Documentation

#### 7.105.1.1 enum OsclBinStream::state\_t [protected]

Enumeration values:

- GOOD\_STATE**
- EOF\_STATE**
- FAIL\_STATE**

### 7.105.2 Constructor & Destructor Documentation

#### 7.105.2.1 OsclBinStream::OsclBinStream () [inline]

### 7.105.3 Member Function Documentation

#### 7.105.3.1 void OsclBinStream::Attach (const uint32 *numFragments*, const [OsclMemoryFragment](#) \**fragPtr*)

This method specifies the memory fragment array to use for input.

This array should remain static while the stream refers to it.

**Parameters:**

*numFragments* is the number of elements in the array

*fragPtr* is the pointer to the MemoryFragment array

#### 7.105.3.2 void OsclBinStream::Attach (void \**buffer*, uint32 *l\_length*)

This methods specifies the data buffer to attach to the stream.

**Parameters:**

*buffer* will provide the input

*length* of the buffer

**7.105.3.3 bool OsclBinStream::eof ()**

This method determines if end of stream has been reached.

**Returns:**

true if end of stream has been reached.

**7.105.3.4 bool OsclBinStream::fail ()**

This method determines if an error has occurred in the stream.

**Returns:**

true if an error occurred in the stream.

**7.105.3.5 bool OsclBinStream::good ()**

This method determines if the stream is ok.

**Returns:**

true if stream is ok.

**7.105.3.6 bool OsclBinStream::HaveRoomInCurrentBlock (uint32 *size*) [protected]****7.105.3.7 uint32 OsclBinStream::PositionInBlock ()**

This method returns the current stream position.

**Returns:**

stream position.

**7.105.3.8 bool OsclBinStream::ReserveSpace (uint32 *size*) [protected]****7.105.3.9 void OsclBinStream::Seek (uint32 *absPosition*)**

This method seeks to the specified stream position.

**Returns:**

Stream position.

**7.105.3.10 void OsclBinStream::seekFromCurrentPosition (int32 *offset*)**

This method seeks to the specified offset from the current location.

**Parameters:**

*offset* from current stream location

### 7.105.3.11 uint32 OsclBinStream::tellg ()

This method returns the current stream position.

This method is to be used if the input stream is a pointer to the MemoryFragment array

**Returns:**

Stream position.

## 7.105.4 Field Documentation

**7.105.4.1 const OsclMemoryFragment\* OsclBinStream::firstFragPtr [protected]**

**7.105.4.2 int OsclBinStream::fragsLeft [protected]**

**7.105.4.3 uint32 OsclBinStream::length [protected]**

**7.105.4.4 const OsclMemoryFragment\* OsclBinStream::nextFragPtr [protected]**

**7.105.4.5 int OsclBinStream::numFrags [protected]**

**7.105.4.6 uint8\* OsclBinStream::pBasePosition [protected]**

**7.105.4.7 uint8\* OsclBinStream::pPosition [protected]**

**7.105.4.8 OsclMemoryFragment OsclBinStream::specialFragBuffer [protected]**

**7.105.4.9 state\_t OsclBinStream::state [protected]**

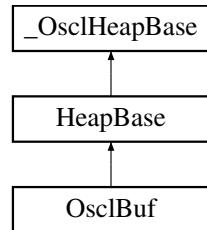
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 7.106 OsclBuf Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclBuf::



### Public Methods

- [OsclBuf](#) (int32 size)
- int32 [Length](#) ()
- [OsclPtr Des](#) ()
- [OsclPtrC DesC](#) ()

### Static Public Methods

- OsclBuf \* [NewL](#) (int32 size)
- void [Delete](#) (OsclBuf \*a)

### Data Fields

- uint8 \* [iBuffer](#)
- int32 [iMaxLength](#)
- int32 [iLength](#)

## 7.106.1 Constructor & Destructor Documentation

**7.106.1.1** `OsclBuf::OsclBuf (int32 size)` [inline]

## 7.106.2 Member Function Documentation

**7.106.2.1** `void OsclBuf::Delete (OsclBuf * a)` [inline, static]

**7.106.2.2** `OsclPtr OsclBuf::Des ()` [inline]

**7.106.2.3** `OsclPtrC OsclBuf::DesC ()` [inline]

**7.106.2.4** `int32 OsclBuf::Length ()` [inline]

**7.106.2.5** `OsclBuf* OsclBuf::NewL (int32 size)` [inline, static]

## 7.106.3 Field Documentation

**7.106.3.1** `uint8* OsclBuf::iBuffer`

**7.106.3.2** `int32 OsclBuf::iLength`

**7.106.3.3** `int32 OsclBuf::iMaxLength`

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

- [oscl\\_file\\_async\\_read.h](#)

## 7.107 OsclCompareLess< T > Class Template Reference

```
#include <oscl_priqueue.h>
```

### Public Methods

- int [compare](#) (T &a, T &b) const

```
template<class T> class OsclCompareLess< T >
```

#### 7.107.1 Member Function Documentation

**7.107.1.1 template<class T> int OsclCompareLess< T >::compare (T & a, T & b) const  
[inline]**

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

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

## 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 \(OSCL\\_String &, Oscl\\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &\)](#)
- [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 `OsclComponentRegistry::OsclComponentRegistry ()`

7.108.2.2 `OsclComponentRegistry::~OsclComponentRegistry ()`

## 7.108.3 Member Function Documentation

7.108.3.1 `void OsclComponentRegistry::CloseSession ()`

7.108.3.2 `OsclComponentFactory OsclComponentRegistry::FindExact (OSCL_String &)`

7.108.3.3 `void OsclComponentRegistry::FindHierarchical (OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &)`

7.108.3.4 `void OsclComponentRegistry::OpenSession ()`

7.108.3.5 `int32 OsclComponentRegistry::Register (uint32 & aId, OSCL_String &, OsclComponentFactory)`

7.108.3.6 `int32 OsclComponentRegistry::Unregister (uint32)`

7.108.3.7 `int32 OsclComponentRegistry::Unregister (OSCL_String &)`

## 7.108.4 Field Documentation

7.108.4.1 `uint32 OsclComponentRegistry::iComponentIdCounter`

7.108.4.2 `OsclComponentRegistryData OsclComponentRegistry::iData`

7.108.4.3 `OsclMutex OsclComponentRegistry::iMutex`

7.108.4.4 `uint32 OsclComponentRegistry::iNumSessions`

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

- `oscl_registry_serv_impl.h`

## 7.109 OsclComponentRegistryData Class Reference

```
#include <oscl_registry_serv_impl.h>
```

### Public Methods

- [OsclComponentRegistryElement \\* Find \(OSCL\\_String &, bool aExact\)](#)

### Data Fields

- [Oscl\\_Vector< OsclComponentRegistryElement, OsclMemAllocator > iVec](#)

#### 7.109.1 Detailed Description

Registry

#### 7.109.2 Member Function Documentation

##### 7.109.2.1 [OsclComponentRegistryElement\\* OsclComponentRegistryData::Find \(OSCL\\_String &, bool aExact\)](#)

#### 7.109.3 Field Documentation

##### 7.109.3.1 [Oscl\\_Vector<OsclComponentRegistryElement, OsclMemAllocator> OsclComponentRegistryData::iVec](#)

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

- [oscl\\_registry\\_serv\\_impl.h](#)

## 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**

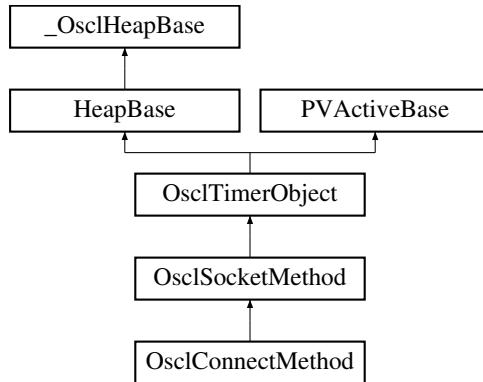
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_serv\\_impl.h](#)

## 7.111 OsclConnectMethod Class Reference

```
#include <oscl_socket_connect.h>
```

Inheritance diagram for OsclConnectMethod::



### Public Methods

- [~OsclConnectMethod \(\)](#)
- [TPVSocketEvent Connect \(OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsclConnectRequest \\* ConnectRequest \(\)](#)

### Static Public Methods

- [OsclConnectMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 7.111.1 Constructor & Destructor Documentation

##### 7.111.1.1 OsclConnectMethod::~OsclConnectMethod ()

#### 7.111.2 Member Function Documentation

##### 7.111.2.1 TPVSocketEvent OsclConnectMethod::Connect (OsclNetworkAddress & aAddress, int32 aTimeout)

##### 7.111.2.2 OsclConnectRequest\* OsclConnectMethod::ConnectRequest () [inline]

##### 7.111.2.3 OsclConnectMethod\* OsclConnectMethod::NewL (OsclIPSocketI & c) [static]

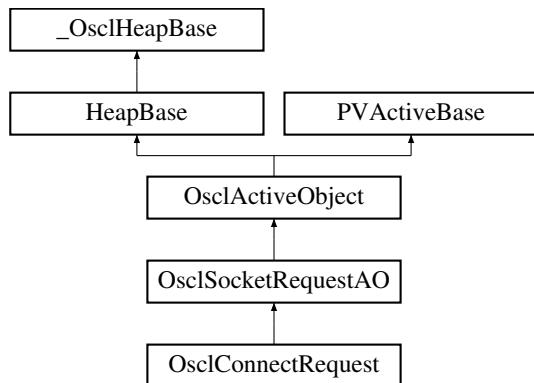
The documentation for this class was generated from the following file:

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

## 7.112 OsclConnectRequest Class Reference

```
#include <oscl_socket_connect.h>
```

Inheritance diagram for OsclConnectRequest::



### Public Methods

- [OsclConnectRequest \(OsclSocketMethod &c\)](#)
- [void Connect \(OsclNetworkAddress &aAddress\)](#)

#### 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 OsclConnectRequest::OsclConnectRequest ([OsclSocketMethod & c](#)) [inline]**

#### 7.112.3 Member Function Documentation

**7.112.3.1 void OsclConnectRequest::Connect ([OsclNetworkAddress & aAddress](#))**

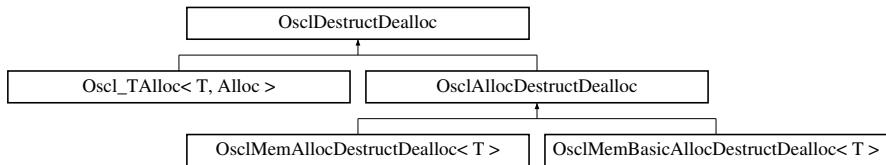
The documentation for this class was generated from the following file:

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

## 7.113 OsclDestructDealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for OsclDestructDealloc::



### Public Methods

- virtual ~OsclDestructDealloc ()
- virtual void [destruct\\_and\\_dealloc \(OsclAny \\*ptr\)=0](#)

#### 7.113.1 Constructor & Destructor Documentation

**7.113.1.1 virtual OsclDestructDealloc::~OsclDestructDealloc () [inline, virtual]**

#### 7.113.2 Member Function Documentation

**7.113.2.1 virtual void OsclDestructDealloc::destruct\_and\_dealloc (OsclAny \* ptr) [pure virtual]**

Implemented in [Oscl\\_TAlloc< T, Alloc >](#), [OsclMemAllocDestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [Oscl\\_TAlloc< entry\\_type, Alloc >](#), [Oscl\\_TAlloc< node\\_type, TagTree\\_Allocator >](#), [Oscl\\_TAlloc< node\\_type, alloc\\_type >](#), [Oscl\\_TAlloc< MM\\_StatsNodeTagTreeType, OsclMemBasicAllocator >](#), [Oscl\\_TAlloc< char, alloc\\_type >](#), [Oscl\\_TAlloc< tag\\_base\\_unit, Alloc >](#), [Oscl\\_TAlloc< PVLogger, alloc\\_type >](#), and [Oscl\\_TAlloc< node\\_type, Alloc >](#).

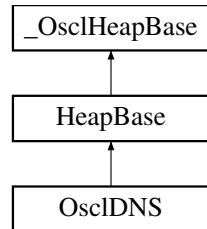
The documentation for this class was generated from the following file:

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

## 7.114 OsclDNS Class Reference

```
#include <oscl_dns.h>
```

Inheritance diagram for OsclDNS::



### Public Methods

- OSCL\_IMPORT\_REF ~OsclDNS ()
- OSCL\_IMPORT\_REF TPVDNSEvent GetHostByName (char \*name, OsclNetworkAddress &addr, int32 aTimeoutMsec=-1, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \*aAddressList=NULL)
- OSCL\_IMPORT\_REF void CancelGetHostByName ()

### Static Public Methods

- OSCL\_IMPORT\_REF OsclDNS \* NewL (Oscl\_DefAlloc &alloc, OsclSocketServ &aServ, OsclDNSObserver &aObserver, uint32 aId)

### Friends

- class OsclDNSRequestAO

#### 7.114.1 Detailed Description

The DNS class

#### 7.114.2 Constructor & Destructor Documentation

##### 7.114.2.1 OSCL\_IMPORT\_REF OsclDNS::~OsclDNS ()

Destructor.

Note: the application must de-allocate the DNS object using the same allocator that was passed in the NewL object creation call.

#### 7.114.3 Member Function Documentation

##### 7.114.3.1 OSCL\_IMPORT\_REF void OsclDNS::CancelGetHostByName ()

Cancel GetHostByName

This method will cancel any pending GetHostByName operation on the current object, causing the GetHostByName to complete with error EPVDNSCancel. If there is no pending GetHostByName operation, this method will have no effect.

#### 7.114.3.2 OSCL\_IMPORT\_REF TPVDNSEvent OsclDNS::GetHostByName (char \* *name*, OsclNetworkAddress & *addr*, int32 *aTimeoutMsec* = -1, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \* *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, EPVDNSFailure for failure.

#### 7.114.3.3 OSCL\_IMPORT\_REF OsclDNS\* OsclDNS::NewL (Oscl\_DefAlloc & *alloc*, OsclSocketServ & *aServ*, OsclDNSObserver & *aObserver*, uint32 *aId*) [static]

DNS object creation.

**Parameters:**

*alloc*: Memory allocator

*aServ*: Socket server.

*aObserver*: DNS Event observer

*aId*: Unique ID for this DNS object. This ID will be included in all callbacks associated with this DNS object.

### 7.114.4 Friends And Related Function Documentation

#### 7.114.4.1 friend class OsclDNSRequestAO [friend]

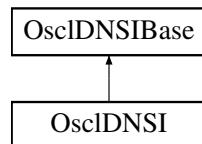
The documentation for this class was generated from the following file:

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

## 7.115 OsclDNSI Class Reference

```
#include <oscl_dns_imp_pv.h>
```

Inheritance diagram for OsclDNSI::



### Public Methods

- [~OsclDNSI \(\)](#)
- int32 [Open \(OsclSocketServI &aServer\)](#)
- int32 [Close \(\)](#)
- void [GetHostByName \(GetHostNameParam &, OsclDNSRequestAO &\)](#)
- void [GetHostByNameSuccess \(GetHostNameParam &\)](#)
- void [GetNextHost \(OsclDNSRequestAO &\)](#)
- void [GetNextHostSuccess \(GetHostNameParam &\)](#)
- bool [GetHostByNameResponseContainsAliasInfo \(\)](#)

### Static Public Methods

- OsclDNSI \* [NewL \(Oscl\\_DefAlloc &a\)](#)

### Friends

- class [OsclDNSRequest](#)
- class [OsclGetHostByNameRequest](#)
- class [DNSRequestParam](#)

### 7.115.1 Detailed Description

OsclDNSI, non-Symbian implementation

### 7.115.2 Constructor & Destructor Documentation

#### 7.115.2.1 OsclDNSI::~OsclDNSI ()

### 7.115.3 Member Function Documentation

#### 7.115.3.1 int32 OsclDNSI::Close () [virtual]

Implements [OsclDNSIBase](#).

**7.115.3.2 void OsclDNSI::GetHostByName ([GetHostNameParam](#) &, [OsclDNSRequestAO](#) &)** [virtual]

Implements [OsclDNSIBase](#).

**7.115.3.3 bool OsclDNSI::GetHostNameResponseContainsAliasInfo ()** [virtual]

Implements [OsclDNSIBase](#).

**7.115.3.4 void OsclDNSI::GetHostNameSuccess ([GetHostNameParam](#) &)** [virtual]

Implements [OsclDNSIBase](#).

**7.115.3.5 void OsclDNSI::GetNextHost ([OsclDNSRequestAO](#) &)** [virtual]

Implements [OsclDNSIBase](#).

**7.115.3.6 void OsclDNSI::GetNextHostSuccess ([GetHostNameParam](#) &)** [virtual]

Implements [OsclDNSIBase](#).

**7.115.3.7 OsclDNSI\* OsclDNSI::NewL ([Oscl\\_DefAlloc](#) & *a*)** [static]

**7.115.3.8 int32 OsclDNSI::Open ([OsclSocketServI](#) & *aServer*)** [virtual]

Implements [OsclDNSIBase](#).

## 7.115.4 Friends And Related Function Documentation

**7.115.4.1 friend class DNSRequestParam** [friend]

**7.115.4.2 friend class OsclDNSRequest** [friend]

Reimplemented from [OsclDNSIBase](#).

**7.115.4.3 friend class OsclGetHostNameRequest** [friend]

Reimplemented from [OsclDNSIBase](#).

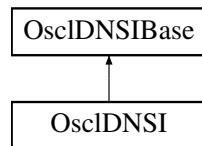
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_imp\\_pv.h](#)

## 7.116 OsclDNSIBase Class Reference

```
#include <oscl_dns_imp_base.h>
```

Inheritance diagram for OsclDNSIBase::



### Public Methods

- virtual ~OsclDNSIBase ()
- virtual int32 Open (OsclSocketServI &aServer)=0
- virtual int32 Close ()=0
- virtual void GetHostByName (GetHostNameParam &, OsclDNSRequestAO &)=0
- virtual void GetHostByNameSuccess (GetHostNameParam &)=0
- virtual bool GetHostByNameResponseContainsAliasInfo ()=0
- virtual void GetNextHost (OsclDNSRequestAO &)=0
- virtual void GetNextHostSuccess (GetHostNameParam &)=0
- void CancelFxn (TPVDNSFx)

### Protected Methods

- OsclDNSIBase (Oscl\_DefAlloc &a)
- virtual bool IsReady (OsclDNSRequestAO &aObject)=0
- virtual void CancelGetHostByName ()=0

### Protected Attributes

- Oscl\_DefAlloc & iAlloc
- OsclSocketServI \* iSocketServ

### Friends

- class OsclDNSRequest
- class OsclGetHostNameRequest

#### 7.116.1 Detailed Description

OsclDNSIBase is a common base class for all implementations.

## 7.116.2 Constructor & Destructor Documentation

**7.116.2.1** `virtual OsclDNSIBase::~OsclDNSIBase () [virtual]`

**7.116.2.2** `OsclDNSIBase::OsclDNSIBase (Oscl_DefAlloc & a) [protected]`

## 7.116.3 Member Function Documentation

**7.116.3.1** `void OsclDNSIBase::CancelFxn (TPVDNSFxn)`

**7.116.3.2** `virtual void OsclDNSIBase::CancelGetHostByName () [protected, pure virtual]`

**7.116.3.3** `virtual int32 OsclDNSIBase::Close () [pure virtual]`

Implemented in [OsclDNSI](#).

**7.116.3.4** `virtual void OsclDNSIBase::GetHostByName (GetHostNameParam &, OsclDNSRequestAO &) [pure virtual]`

Implemented in [OsclDNSI](#).

**7.116.3.5** `virtual bool OsclDNSIBase::GetHostByNameResponseContainsAliasInfo () [pure virtual]`

Implemented in [OsclDNSI](#).

**7.116.3.6** `virtual void OsclDNSIBase::GetHostByNameSuccess (GetHostNameParam &) [pure virtual]`

Implemented in [OsclDNSI](#).

**7.116.3.7** `virtual void OsclDNSIBase::GetNextHost (OsclDNSRequestAO &) [pure virtual]`

Implemented in [OsclDNSI](#).

**7.116.3.8** `virtual void OsclDNSIBase::GetNextHostSuccess (GetHostNameParam &) [pure virtual]`

Implemented in [OsclDNSI](#).

**7.116.3.9** `virtual bool OsclDNSIBase::IsReady (OsclDNSRequestAO & aObject) [protected, pure virtual]`

**7.116.3.10** `virtual int32 OsclDNSIBase::Open (OsclSocketServI & aServer) [pure virtual]`

Implemented in [OsclDNSI](#).

## 7.116.4 Friends And Related Function Documentation

### 7.116.4.1 friend class OsclDNSRequest [friend]

Reimplemented in [OsclDNSI](#).

### 7.116.4.2 friend class OsclGetHostByNameRequest [friend]

Reimplemented in [OsclDNSI](#).

## 7.116.5 Field Documentation

### 7.116.5.1 [Oscl\\_DefAlloc& OsclDNSIBase::iAlloc](#) [protected]

### 7.116.5.2 [OsclSocketServI\\* OsclDNSIBase::iSocketServ](#) [protected]

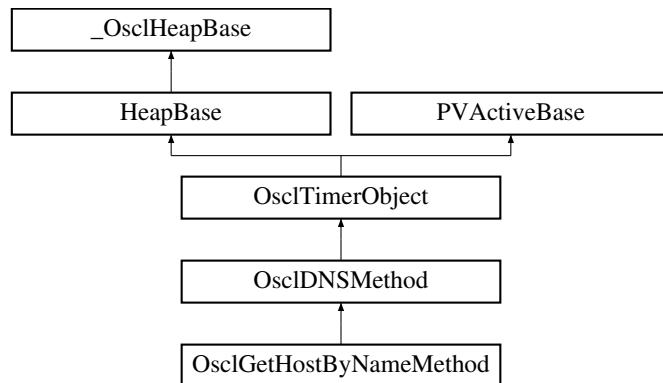
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_imp\\_base.h](#)

## 7.117 OsclDNSMethod Class Reference

```
#include <oscl_dns_method.h>
```

Inheritance diagram for OsclDNSMethod::



### Public Methods

- [OsclDNSMethod \(Oscl\\_DefAlloc &a, const char \\*name, TPVDNSFxn fxn\)](#)
- void [Abort \(\)](#)
- void [AbortAll \(\)](#)
- void [CancelMethod \(\)](#)
- void [Run \(\)](#)

### Data Fields

- [OsclDNSObserver \\* iDNSObserver](#)
- uint32 [iId](#)
- [Oscl\\_DefAlloc & iAlloc](#)
- [TPVDNSFxn iDNSFxn](#)
- [PVLogger \\* iLogger](#)

### Protected Methods

- void [ConstructL \(OsclDNSObserver \\*aObserver, OsclDNSRequestAO \\*aAO, uint32 aId\)](#)
- bool [StartMethod \(int32 aTimeoutMsec\)](#)
- void [MethodDone \(\)](#)

### Protected Attributes

- [OsclDNSRequestAO \\* iDNSRequestAO](#)

#### 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 OsclDNSMethod::OsclDNSMethod ([Oscl\\_DefAlloc](#) & *a*, const char \* *name*, [TPVDNSFxn](#) *fxn*) [inline]**

## 7.117.3 Member Function Documentation

**7.117.3.1 void OsclDNSMethod::Abort ()**

**7.117.3.2 void OsclDNSMethod::AbortAll ()**

**7.117.3.3 void OsclDNSMethod::CancelMethod ()**

**7.117.3.4 void OsclDNSMethod::ConstructL ([OsclDNSObserver](#) \* *aObserver*, [OsclDNSRequestAO](#) \* *aAO*, uint32 *aId*) [protected]**

**7.117.3.5 void OsclDNSMethod::MethodDone () [protected]**

**7.117.3.6 void OsclDNSMethod::Run () [virtual]**

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request

2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implements [PVActiveBase](#).

7.117.3.7 **bool OsclDNSMethod::StartMethod (int32 *aTimeoutMsec*)** [protected]

#### 7.117.4 Field Documentation

7.117.4.1 **Oscl\_DefAlloc& OsclDNSMethod::iAlloc**

7.117.4.2 **TPVDNSFxn OsclDNSMethod::iDNSFxn**

7.117.4.3 **OsclDNSObserver\* OsclDNSMethod::iDNSObserver**

7.117.4.4 **OsclDNSRequestAO\* OsclDNSMethod::iDNSRequestAO** [protected]

7.117.4.5 **uint32 OsclDNSMethod::iId**

7.117.4.6 **PVLogger\* OsclDNSMethod::iLogger**

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

- [oscl\\_dns\\_method.h](#)

## 7.118 OsclDNSObserver Class Reference

```
#include <oscl_dns.h>
```

### Public Methods

- virtual OSCL\_IMPORT\_REF void [HandleDNSEvent](#) (int32 aId, [TPVDNSFxn](#) aFxn, [TPVDNSEvent](#) aEvent, int32 aError)=0
- virtual [~OsclDNSObserver](#) ()

### 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 OsclDNSObserver::~OsclDNSObserver () [inline, virtual]**

### 7.118.3 Member Function Documentation

**7.118.3.1 virtual OSCL\_IMPORT\_REF void OsclDNSObserver::HandleDNSEvent (int32 *aId*,  
[TPVDNSFxn](#) *aFxn*, [TPVDNSEvent](#) *aEvent*, int32 *aError*) [pure virtual]**

DNS Event callback.

#### Parameters:

***aId*:** The ID that was supplied when the DNS object was created.

***aEvent*:** Function completion event. Will be EPVDNSSuccess, EPVDNSTimeout, or EPVDNSFailure.

***aError*:** When the event is EPVDNSFailure, this may contain a platform-specific error code, or zero if none is available.

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

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

## 7.119 OsclDNSRequest Class Reference

```
#include <oscl_dns_request.h>
```

### Public Methods

- [OsclDNSRequest \(\)](#)
- [~OsclDNSRequest \(\)](#)
- void [CancelRequest \(\)](#)
- void [Complete \(bool, int32 aStatus, int32 aSockErr\)](#)
- void [Activate \(DNSRequestParam \\*iParam, OsclDNSRequestAO &a\)](#)

### Data Fields

- [OsclDNSRequestAO \\* iDNSRequestAO](#)
- [DNSRequestParam \\* iDNSRequestParam](#)
- bool [iActive](#)

#### 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** [OsclDNSRequest::OsclDNSRequest \(\) \[inline\]](#)

**7.119.2.2** [OsclDNSRequest::~OsclDNSRequest \(\) \[inline\]](#)

#### 7.119.3 Member Function Documentation

**7.119.3.1** void [OsclDNSRequest::Activate \(DNSRequestParam \\* iParam, OsclDNSRequestAO & a\)](#)

**7.119.3.2** void [OsclDNSRequest::CancelRequest \(\)](#)

**7.119.3.3** void [OsclDNSRequest::Complete \(bool, int32 aStatus, int32 aSockErr\)](#)

#### 7.119.4 Field Documentation

**7.119.4.1** bool [OsclDNSRequest::iActive](#)

**7.119.4.2** [OsclDNSRequestAO\\* OsclDNSRequest::iDNSRequestAO](#)

**7.119.4.3** [DNSRequestParam\\* OsclDNSRequest::iDNSRequestParam](#)

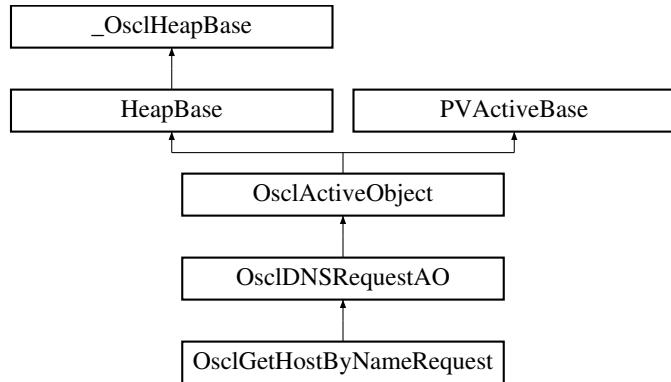
The documentation for this class was generated from the following file:

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

## 7.120 OsclDNSRequestAO Class Reference

```
#include <oscl_dns_method.h>
```

Inheritance diagram for OsclDNSRequestAO::



### Protected Methods

- [OsclDNSRequestAO](#) (const char \*name)
- void [ConstructL](#) ([OsclDNSI](#) \*aDNS, [OsclDNSMethod](#) \*aMethod)
- void [Abort](#) ()
- void [NewRequest](#) ()
- void [RequestDone](#) ()
- int [GetSocketError](#) ()
- [OsclSocketServI](#) \* [Serv](#) ()
- void [DoCancel](#) ()
- void [Run](#) ()
- virtual void [Success](#) ()
- virtual void [Failure](#) ()
- virtual void [Cancelled](#) ()

### Protected Attributes

- [OsclDNSI](#) \* [iDNSI](#)
- [OsclDNSMethod](#) \* [iDNSMethod](#)
- int32 [iSocketError](#)
- [PVLogger](#) \* [iLogger](#)

### Friends

- class [OsclDNSI](#)
- class [OsclDNSMethod](#)
- class [OsclDNSRequest](#)
- class [GetHostNameParam](#)

### 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 OsclDNSRequestAO::OsclDNSRequestAO (const char \* *name*)** [inline, protected]

### 7.120.3 Member Function Documentation

**7.120.3.1 void OsclDNSRequestAO::Abort ()** [inline, protected]

**7.120.3.2 virtual void OsclDNSRequestAO::Cancelled ()** [inline, protected, virtual]

**7.120.3.3 void OsclDNSRequestAO::ConstructL (OsclDNSI \* *aDNS*, OsclDNSMethod \* *aMethod*)** [inline, protected]

**7.120.3.4 void OsclDNSRequestAO::DoCancel ()** [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Reimplemented from [OsclActiveObject](#).

**7.120.3.5 virtual void OsclDNSRequestAO::Failure ()** [inline, protected, virtual]

**7.120.3.6 int OsclDNSRequestAO::GetSocketError ()** [protected]

**7.120.3.7 void OsclDNSRequestAO::NewRequest ()** [protected]

**7.120.3.8 void OsclDNSRequestAO::RequestDone ()** [protected]

**7.120.3.9 void OsclDNSRequestAO::Run ()** [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implements [PVActiveBase](#).

**7.120.3.10** [\*\*OsclSocketServI\\*\*\* OsclDNSRequestAO::Serv \(\)](#) [protected]

**7.120.3.11** [\*\*virtual void OsclDNSRequestAO::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 OsclDNSI\*\*](#) [friend]

**7.120.4.3** [\*\*friend class OsclDNSMethod\*\*](#) [friend]

**7.120.4.4** [\*\*friend class OsclDNSRequest\*\*](#) [friend]

## 7.120.5 Field Documentation

**7.120.5.1** [\*\*OsclDNSI\\* OsclDNSRequestAO::iDNSI\*\*](#) [protected]

**7.120.5.2** [\*\*OsclDNSMethod\\* OsclDNSRequestAO::iDNSMethod\*\*](#) [protected]

**7.120.5.3** [\*\*PVLogger\\* OsclDNSRequestAO::iLogger\*\*](#) [protected]

**7.120.5.4** [\*\*int32 OsclDNSRequestAO::iSocketError\*\*](#) [protected]

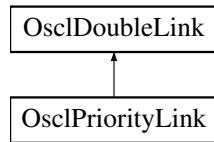
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_method.h](#)

## 7.121 OsclDoubleLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleLink::



### Public Methods

- [OsclDoubleLink \(\)](#)
- void [Remove \(\)](#)
- void [InsertAfter \(OsclDoubleLink \\*aLink\)](#)
- void [InsertBefore \(OsclDoubleLink \\*aLink\)](#)

### Data Fields

- OsclDoubleLink \* [iNext](#)
- OsclDoubleLink \* [iPrev](#)

#### 7.121.1 Constructor & Destructor Documentation

##### 7.121.1.1 [OsclDoubleLink::OsclDoubleLink \(\) \[inline\]](#)

#### 7.121.2 Member Function Documentation

##### 7.121.2.1 [void OsclDoubleLink::InsertAfter \(OsclDoubleLink \\* \*aLink\*\)](#)

##### 7.121.2.2 [void OsclDoubleLink::InsertBefore \(OsclDoubleLink \\* \*aLink\*\)](#)

##### 7.121.2.3 [void OsclDoubleLink::Remove \(\)](#)

#### 7.121.3 Field Documentation

##### 7.121.3.1 [OsclDoubleLink\\* OsclDoubleLink::iNext](#)

##### 7.121.3.2 [OsclDoubleLink\\* OsclDoubleLink::iPrev](#)

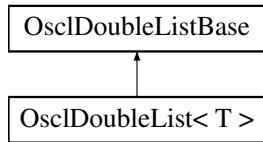
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.122 OsclDoubleList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleList< T >::



### Public Methods

- OSCL\_INLINE OsclDoubleList()
- OSCL\_INLINE OsclDoubleList(int32 anOffset)
- OSCL\_INLINE void InsertHead(T &aRef)
- OSCL\_INLINE void InsertTail(T &aRef)
- OSCL\_INLINE bool IsHead(const T \*aPtr) const
- OSCL\_INLINE bool IsTail(const T \*aPtr) const
- OSCL\_INLINE T \* Head() const
- OSCL\_INLINE T \* Tail() const

```
template<class T> class OsclDoubleList< T >
```

#### 7.122.1 Constructor & Destructor Documentation

**7.122.1.1 template<class T> OSCL\_INLINE OsclDoubleList< T >::OsclDoubleList()**

**7.122.1.2 template<class T> OSCL\_INLINE OsclDoubleList< T >::OsclDoubleList(int32  
anOffset)**

#### 7.122.2 Member Function Documentation

**7.122.2.1 template<class T> OSCL\_INLINE T\* OsclDoubleList< T >::Head()**

**7.122.2.2 template<class T> OSCL\_INLINE void OsclDoubleList< T >::InsertHead(T & aRef)**

**7.122.2.3 template<class T> OSCL\_INLINE void OsclDoubleList< T >::InsertTail(T & aRef)**

**7.122.2.4 template<class T> OSCL\_INLINE bool OsclDoubleList< T >::IsHead(const T \* aPtr)  
const**

**7.122.2.5 template<class T> OSCL\_INLINE bool OsclDoubleList< T >::IsTail(const T \* aPtr)  
const**

**7.122.2.6 template<class T> OSCL\_INLINE T\* OsclDoubleList< T >::Tail()**

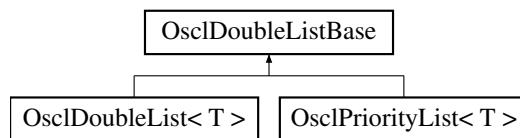
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.123 OsclDoubleListBase Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleListBase::



### Public Methods

- bool [IsEmpty \(\) const](#)
- void [SetOffset \(int32 anOffset\)](#)
- void [Reset \(\)](#)
- [OsclDoubleLink \\* getHead \(\)](#)
- int32 [getOffset \(\)](#)

### Protected Methods

- [OsclDoubleListBase \(\)](#)
- [OsclDoubleListBase \(int32 anOffset\)](#)
- void [InsertHead \(OsclAny \\*aPtr\)](#)
- void [InsertTail \(OsclAny \\*aPtr\)](#)
- void [Insert \(OsclAny \\*aPtr\)](#)

### Protected Attributes

- [OsclDoubleLink iHead](#)
- int32 [iOffset](#)

### 7.123.1 Constructor & Destructor Documentation

7.123.1.1 **OsclDoubleListBase::OsclDoubleListBase ()** [protected]

7.123.1.2 **OsclDoubleListBase::OsclDoubleListBase (int32 *anOffset*)** [protected]

### 7.123.2 Member Function Documentation

7.123.2.1 **OsclDoubleLink\* OsclDoubleListBase::getHead ()** [inline]

7.123.2.2 **int32 OsclDoubleListBase::getOffset ()** [inline]

7.123.2.3 **void OsclDoubleListBase::Insert (OsclAny \* *aPtr*)** [protected]

7.123.2.4 **void OsclDoubleListBase::InsertHead (OsclAny \* *aPtr*)** [protected]

7.123.2.5 **void OsclDoubleListBase::InsertTail (OsclAny \* *aPtr*)** [protected]

7.123.2.6 **bool OsclDoubleListBase::IsEmpty ()**

7.123.2.7 **void OsclDoubleListBase::Reset ()**

7.123.2.8 **void OsclDoubleListBase::SetOffset (int32 *anOffset*)**

### 7.123.3 Field Documentation

7.123.3.1 **OsclDoubleLink OsclDoubleListBase::iHead** [protected]

7.123.3.2 **int32 OsclDoubleListBase::iOffset** [protected]

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

- [oscl\\_double\\_list.h](#)

## 7.124 OsclDoubleRunner< T > Class Template Reference

```
#include <oscl_double_list.h>
```

### Public Methods

- [OsclDoubleRunner \(OsclDoubleListBase &aQue\)](#)
- [void Set \(T &aLink\)](#)
- [operator T \\* \(\)](#)
- [T \\* operator++ \(int\)](#)
- [T \\* operator- \(int\)](#)
- [void SetToHead \(\)](#)
- [void SetToTail \(\)](#)

### Protected Attributes

- [int32 iOffset](#)
- [OsclDoubleLink \\* iHead](#)
- [OsclDoubleLink \\* iNext](#)

```
template<class T> class OsclDoubleRunner< T >
```

#### 7.124.1 Constructor & Destructor Documentation

7.124.1.1 [template<class T> OsclDoubleRunner< T >::OsclDoubleRunner \(OsclDoubleListBase & aQue\) \[inline\]](#)

#### 7.124.2 Member Function Documentation

7.124.2.1 [template<class T> OsclDoubleRunner< T >::operator T \\* \(\) \[inline\]](#)

7.124.2.2 [template<class T> T\\* OsclDoubleRunner< T >::operator++ \(int\) \[inline\]](#)

7.124.2.3 [template<class T> T\\* OsclDoubleRunner< T >::operator- \(int\)](#)

7.124.2.4 [template<class T> void OsclDoubleRunner< T >::Set \(T & aLink\) \[inline\]](#)

7.124.2.5 [template<class T> void OsclDoubleRunner< T >::SetToHead \(\) \[inline\]](#)

7.124.2.6 [template<class T> void OsclDoubleRunner< T >::SetToTail \(\) \[inline\]](#)

#### 7.124.3 Field Documentation

7.124.3.1 [template<class T> OsclDoubleLink\\* OsclDoubleRunner< T >::iHead \[protected\]](#)

7.124.3.2 [template<class T> OsclDoubleLink\\* OsclDoubleRunner< T >::iNext \[protected\]](#)

7.124.3.3 [template<class T> int32 OsclDoubleRunner< T >::iOffset \[protected\]](#)

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

- 
- [oscl\\_double\\_list.h](#)

## 7.125 OsclError Class Reference

```
#include <oscl_error.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [PushL \(\\_OsclHeapBase \\*aPtr\)](#)
- OSCL\_IMPORT\_REF void [PushL \(OsclAny \\*aPtr\)](#)
- OSCL\_IMPORT\_REF void [PushL \(OsclTrapItem anItem\)](#)
- OSCL\_IMPORT\_REF void [Pop \(\)](#)
- OSCL\_IMPORT\_REF void [Pop \(int32 aCount\)](#)
- OSCL\_IMPORT\_REF void [PopDealloc \(\)](#)
- OSCL\_IMPORT\_REF void [PopDealloc \(int32 aCount\)](#)
- OSCL\_IMPORT\_REF void [Leave \(int32 aReason\)](#)
- OSCL\_IMPORT\_REF void [LeaveIfNull \(OsclAny \\*a\)](#)
- OSCL\_IMPORT\_REF void [LeaveIfError \(int32 aReason\)](#)

### 7.125.1 Detailed Description

User Error class

### 7.125.2 Member Function Documentation

#### 7.125.2.1 OSCL\_IMPORT\_REF void OsclError::Leave (int32 *aReason*) [static]

Do a Leave error, with the given reason code. When a leave occurs, all items on the cleanup stack for the current trap level will be destroyed, and execution will jump to the trap handler.

#### 7.125.2.2 OSCL\_IMPORT\_REF void OsclError::LeaveIfError (int32 *aReason*) [static]

Evaluate the input parameter, and if it is an error code (non-zero), then do a Leave with the provided reason code.

#### 7.125.2.3 OSCL\_IMPORT\_REF void OsclError::LeaveIfNull (OsclAny \* *a*) [static]

Evaluate the input parameter, and if it is null, do a Leave with OsclErrNoMemory reason code.

#### 7.125.2.4 OSCL\_IMPORT\_REF void OsclError::Pop (int32 *aCount*) [static]

Pop the cleanup stack N times

#### 7.125.2.5 OSCL\_IMPORT\_REF void OsclError::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 OsclAny OsclErrorAllocator::deallocate (OsclAny \* aPointer) [inline, static]**

static method to deallocate a block of memory on heap

**Parameters:**

*aPointer* - pointer to block of memory to be deallocated

**7.126.3.3 void OsclErrorAllocator::operator delete (OsclAny \* aPtr, OsclAny \* aPtr2) [inline]**

delete operator that doesn't do anything, user has to deallocate manually

**7.126.3.4 void\* OsclErrorAllocator::operator new (uint32 size, OsclAny \* aPtr) [inline]**

placement new operator that allocates memory using the user defined methods

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

- [oscl\\_error\\_allocator.h](#)

## 7.127 OsclErrorTrap Class Reference

```
#include <oscl_error.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF int32 [Init \(Oscl\\_DefAlloc \\*aAlloc=NULL\)](#)
- OSCL\_IMPORT\_REF int32 [Cleanup \(\)](#)
- OSCL\_IMPORT\_REF [OsclErrorTrapImp \\* GetErrorTrapImp \(\)](#)

#### 7.127.1 Member Function Documentation

##### 7.127.1.1 OSCL\_IMPORT\_REF int32 OsclErrorTrap::Cleanup () [static]

Cleanup and destroy error trap for the calling thread.

###### Returns:

0 for success, or an error

##### 7.127.1.2 OSCL\_IMPORT\_REF [OsclErrorTrapImp\\*](#) OsclErrorTrap::GetErrorTrapImp () [static]

Get the ErrorTrapImp for the current thread. Leaves on error.

##### 7.127.1.3 OSCL\_IMPORT\_REF int32 OsclErrorTrap::Init ([Oscl\\_DefAlloc \\* aAlloc = NULL](#)) [static]

Allocate and initialize error trap for the calling thread.

###### Parameters:

*aAlloc*: optional, allocator to use for the internal implementation.

###### Returns:

0 for success, or an error

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

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

## 7.128 OsclErrorTrapImp Class Reference

```
#include <oscl_error_trapcleanup.h>
```

### Public Methods

- OSCL\_IMPORT\_REF void [UnTrap \(\)](#)

### Static Public Methods

- OSCL\_IMPORT\_REF OsclErrorTrapImp \* [Trap \(\)](#)
- OSCL\_IMPORT\_REF OsclErrorTrapImp \* [TrapNoTls \(OsclErrorTrapImp \\*\)](#)

### Data Fields

- [OsclJump \\* iJumpData](#)
- int32 [iLeave](#)
- [OsclTrapStack \\* iTrapStack](#)

### Friends

- class [OsclErrorTrap](#)
- class [OsclError](#)
- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclJump](#)
- class [OsclJumpMark](#)
- class [OsclTrapStack](#)
- class [CPVInterfaceProxy](#)
- class [OsclScheduler](#)

### 7.128.1 Detailed Description

A per-thread cleanup stack with nested trap support.

### 7.128.2 Member Function Documentation

#### 7.128.2.1 OSCL\_IMPORT\_REF OsclErrorTrapImp\* OsclErrorTrapImp::Trap () [static]

PV trap cleanup. Public for use in macros only.

#### 7.128.2.2 OSCL\_IMPORT\_REF OsclErrorTrapImp\* OsclErrorTrapImp::TrapNoTls (OsclErrorTrapImp \*) [static]

#### 7.128.2.3 OSCL\_IMPORT\_REF void OsclErrorTrapImp::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 OsclError [friend]**

7.128.3.3 **friend class OsclErrorTrap [friend]**

7.128.3.4 **friend class OsclExecScheduler [friend]**

7.128.3.5 **friend class OsclExecSchedulerCommonBase [friend]**

7.128.3.6 **friend class OsclJump [friend]**

7.128.3.7 **friend class OsclJumpMark [friend]**

7.128.3.8 **friend class OsclScheduler [friend]**

7.128.3.9 **friend class OsclTrapStack [friend]**

### 7.128.4 Field Documentation

7.128.4.1 **OsclJump\* OsclErrorTrapImp::iJumpData**

7.128.4.2 **int32 OsclErrorTrapImp::iLeave**

7.128.4.3 **OsclTrapStack\* OsclErrorTrapImp::iTrapStack**

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

- [oscl\\_error\\_trapcleanup.h](#)

## 7.129 OsclException< LeaveCode > Class Template Reference

`oscl_exception.h` contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

```
#include <oscl_exception.h>
```

### Public Methods

- `OsclException ()`

### Static Public Methods

- `int getLeaveCode ()`

#### 7.129.1 Detailed Description

`template<int LeaveCode> class OsclException< LeaveCode >`

`oscl_exception.h` contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

All PacketVideo exception classes will be derived from the OsclException class. Each derived class will have a static function where the leave code can be obtained. This avoids the issue of having static members in a DLL. The function needs to be static so it can be called without an instance of the class

#### 7.129.2 Constructor & Destructor Documentation

**7.129.2.1 `template<int LeaveCode> OsclException< LeaveCode >::OsclException ()`**  
[inline]

#### 7.129.3 Member Function Documentation

**7.129.3.1 `template<int LeaveCode> int OsclException< LeaveCode >::getLeaveCode ()`**  
[inline, static]

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

- `oscl_exception.h`

## 7.130 OsclExclusiveArrayPtr< T > Class Template Reference

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

```
#include <oscl_exclusive_ptr.h>
```

### Public Methods

- **OsclExclusiveArrayPtr (T \*inPtr=0)**  
*Default constructor Initializes the pointer and takes ownership.*
- **OsclExclusiveArrayPtr (OsclExclusiveArrayPtr< T > &\_Y)**  
*Copy constructor.*
- **OsclExclusiveArrayPtr< T > & operator= (OsclExclusiveArrayPtr< T > &\_Y)**  
*Assignment operator from an another OsclExclusiveArrayPtr.*
- **virtual ~OsclExclusiveArrayPtr ()**  
*Destructor.*
- **T & operator\* () const**  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- **T \* operator-> () const**  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- **T \* get () const**  
*get() method returns the pointer, currently owned by the class.*
- **T \* release ()**  
*release() method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- **bool set (T \*ptr)**  
*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- **T \* \_Ptr**

#### 7.130.1 Detailed Description

**template<class T> class OsclExclusiveArrayPtr< T >**

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

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an [OsclExclusivePtr](#) object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The [OsclExclusivePtr](#) is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### **7.130.2 Constructor & Destructor Documentation**

**7.130.2.1 template<class T> OsclExclusiveArrayPtr< T >::OsclExclusiveArrayPtr (T \* *inPtr* = 0) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**7.130.2.2 template<class T> OsclExclusiveArrayPtr< T >::OsclExclusiveArrayPtr (OsclExclusiveArrayPtr< T > & *\_Y*) [inline]**

Copy constructor.

Initializes the pointer and takes ownership from another OsclExclusiveArrayPtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

**7.130.2.3 template<class T> virtual OsclExclusiveArrayPtr< T >::~OsclExclusiveArrayPtr () [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\* OsclExclusiveArrayPtr< T >::get () const [inline]**

[get\(\)](#) method returns the pointer, currently owned by the class.

**7.130.3.2 template<class T> T& OsclExclusiveArrayPtr< T >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusiveArrayPtr can be used like the regular pointer that it was initialized with.

**7.130.3.3 template<class T> T\* OsclExclusiveArrayPtr< T >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusiveArrayPtr can be used like the regular pointer that it was initialized with.

**7.130.3.4 template<class T> OsclExclusiveArrayPtr<T>& OsclExclusiveArrayPtr< T >::operator= (OsclExclusiveArrayPtr< T > & \_Y) [inline]**

Assignment operator from an another OsclExclusiveArrayPtr.

**Parameters:**

*\_Y* The value parameter should be another OsclExclusiveArrayPtr

**Returns:**

Returns a reference to this OsclExclusiveArrayPtr instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the OsclExclusiveArrayPtr given as the input parameter. The ownership of the pointer is transferred.

**7.130.3.5 template<class T> T\* OsclExclusiveArrayPtr< T >::release () [inline]**

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**7.130.3.6 template<class T> bool OsclExclusiveArrayPtr< T >::set (T \*ptr) [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

## 7.130.4 Field Documentation

**7.130.4.1 template<class T> T\* OsclExclusiveArrayPtr< T >::\_Ptr [protected]**

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

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

## 7.131 OsclExclusivePtr< T > Class Template Reference

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

```
#include <oscl_exclusive_ptr.h>
```

### Public Methods

- **OsclExclusivePtr** (T \*inPtr=0)  
*Default constructor Initializes the pointer and takes ownership.*
- **OsclExclusivePtr** (OsclExclusivePtr< T > &\_Y)  
*Copy constructor.*
- OsclExclusivePtr< T > & **operator=** (OsclExclusivePtr< T > &\_Y)  
*Assignment operator from an another OsclExclusivePtr.*
- virtual ~**OsclExclusivePtr** ()  
*Destructor.*
- T & **operator \*** () const  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- T \* **operator ->** () const  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- T \* **get** () const  
***get()** method returns the pointer, currently owned by the class.*
- T \* **release** ()  
***release()** method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool **set** (T \*ptr)  
***set()** method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- T \* **\_Ptr**

#### 7.131.1 Detailed Description

**template<class T> class OsclExclusivePtr< T >**

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

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an OsclExclusivePtr object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The OsclExclusivePtr is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### 7.131.2 Constructor & Destructor Documentation

**7.131.2.1 template<class T> OsclExclusivePtr< T >::OsclExclusivePtr (T \* *inPtr* = 0) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**7.131.2.2 template<class T> OsclExclusivePtr< T >::OsclExclusivePtr (OsclExclusivePtr< T > & *\_Y*) [inline]**

Copy constructor.

Initializes the pointer and takes ownership from another OsclExclusivePtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

**7.131.2.3 template<class T> virtual OsclExclusivePtr< T >::~OsclExclusivePtr () [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\* OsclExclusivePtr< T >::get () const [inline]**

[get\(\)](#) method returns the pointer, currently owned by the class.

**7.131.3.2 template<class T> T& OsclExclusivePtr< T >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusivePtr can be used like the regular pointer that it was initialized with.

**7.131.3.3 template<class T> T\* OsclExclusivePtr< T >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusivePtr can be used like the regular pointer that it was initialized with.

**7.131.3.4 template<class T> OsclExclusivePtr<T>& OsclExclusivePtr< T >::operator= (OsclExclusivePtr< T > & \_Y) [inline]**

Assignment operator from an another OsclExclusivePtr.

**Parameters:**

*\_Y* The value parameter should be another OsclExclusivePtr

**Returns:**

Returns a reference to this OsclExclusivePtr instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the OsclExclusivePtr given as the input parameter. The ownership of the pointer is transferred.

**7.131.3.5 template<class T> T\* OsclExclusivePtr< T >::release () [inline]**

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**7.131.3.6 template<class T> bool OsclExclusivePtr< T >::set (T \*ptr) [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

## 7.131.4 Field Documentation

**7.131.4.1 template<class T> T\* OsclExclusivePtr< T >::\_Ptr [protected]**

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

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

## 7.132 OsclExclusivePtrA< T, Alloc > Class Template Reference

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

```
#include <oscl_exclusive_ptr.h>
```

### Public Methods

- `OsclExclusivePtrA (T *inPtr=0)`  
*Default constructor Initializes the pointer and takes ownership.*
- `OsclExclusivePtrA (OsclExclusivePtrA< T, Alloc > &_Y)`  
*Copy constructor.*
- `OsclExclusivePtrA< T, Alloc > & operator= (OsclExclusivePtrA< T, Alloc > &_Y)`  
*Assignment operator from an another `OsclExclusiveArrayPtr`.*
- `virtual ~OsclExclusivePtrA ()`  
*Destructor.*
- `T & operator* () const`  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- `T * operator-> () const`  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- `T * get () const`  
*`get()` method returns the pointer, currently owned by the class.*
- `T * release ()`  
*`release()` method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- `bool set (T *ptr)`  
*`set()` method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- `T * _Ptr`

#### 7.132.1 Detailed Description

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

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

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an [OsclExclusivePtr](#) object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The [OsclExclusivePtr](#) is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### 7.132.2 Constructor & Destructor Documentation

**7.132.2.1 template<class T, class Alloc> OsclExclusivePtrA< T, Alloc >::OsclExclusivePtrA (T \* *inPtr* = 0) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**7.132.2.2 template<class T, class Alloc> OsclExclusivePtrA< T, Alloc >::OsclExclusivePtrA (OsclExclusivePtrA< T, Alloc > & *\_Y*) [inline]**

Copy constructor.

Initializes the pointer and takes ownership from another [OsclExclusiveArrayPtr](#). Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

**7.132.2.3 template<class T, class Alloc> virtual OsclExclusivePtrA< T, Alloc >::~OsclExclusivePtrA () [inline, virtual]**

Destructor.

The pointer is deleted in case this class still has ownership

### 7.132.3 Member Function Documentation

**7.132.3.1 template<class T, class Alloc> T\* OsclExclusivePtrA< T, Alloc >::get () const [inline]**

[get\(\)](#) method returns the pointer, currently owned by the class.

**7.132.3.2 template<class T, class Alloc> T& OsclExclusivePtrA< T, Alloc >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the [OsclExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

**7.132.3.3 template<class T, class Alloc> T\* OsclExclusivePtrA< T, Alloc >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the [OsclExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

**7.132.3.4 template<class T, class Alloc> OsclExclusivePtrA<T, Alloc>& OsclExclusivePtrA< T, Alloc >::operator= (OsclExclusivePtrA< T, Alloc > & \_Y) [inline]**

Assignment operator from an another [OsclExclusiveArrayPtr](#).

**Parameters:**

*\_Y* The value parameter should be another [OsclExclusiveArrayPtr](#)

**Returns:**

Returns a reference to this [OsclExclusiveArrayPtr](#) instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the [OsclExclusiveArrayPtr](#) given as the input parameter. The ownership of the pointer is transferred.

**7.132.3.5 template<class T, class Alloc> T\* OsclExclusivePtrA< T, Alloc >::release () [inline]**

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**7.132.3.6 template<class T, class Alloc> bool OsclExclusivePtrA< T, Alloc >::set (T \*ptr) [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

## 7.132.4 Field Documentation

**7.132.4.1 template<class T, class Alloc> T\* OsclExclusivePtrA< T, Alloc >::\_Ptr [protected]**

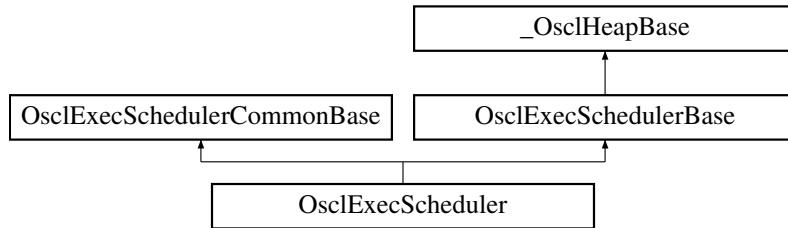
The documentation for this class was generated from the following file:

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

## 7.133 OsclExecScheduler Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for OsclExecScheduler::



### Public Methods

- OSCL\_IMPORT\_REF void [RunSchedulerNonBlocking](#) (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)
- OSCL\_IMPORT\_REF void [RegisterForCallback](#) ([OsclSchedulerObserver](#) \*aCallback, [OsclAny](#) \*aCallbackContext)

### Static Public Methods

- OSCL\_IMPORT\_REF [OsclExecScheduler](#) \* [Current](#) ()

### Friends

- class [OsclScheduler](#)

#### 7.133.1 Member Function Documentation

##### 7.133.1.1 OSCL\_IMPORT\_REF OsclExecScheduler\* OsclExecScheduler::Current () [static]

Get currently installed scheduler for calling thread, or NULL if no scheduler is installed.

##### 7.133.1.2 OSCL\_IMPORT\_REF void OsclExecScheduler::RegisterForCallback ([OsclSchedulerObserver](#) \* aCallback, [OsclAny](#) \* aCallbackContext)

Register for a notification when non-blocking scheduler needs to run again.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

##### 7.133.1.3 OSCL\_IMPORT\_REF void OsclExecScheduler::RunSchedulerNonBlocking (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)

Run PV scheduler in non-blocking mode. This call returns when the desired number of Run calls have been made, or when there are no more active objects that are ready to run.

**Parameters:**

*aTargetCount*: (input param) the maximum number of Run calls to make.

*aReady*: (output param) tells the number of active objects that are currently ready to run.

*aDelayMsec*: (output param) If no active objects are ready to run, but one or more active objects are waiting on timers, this parameter will tell the time interval from the current time until the first of the pending timer objects will be ready to run, in milliseconds.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

## 7.133.2 Friends And Related Function Documentation

### 7.133.2.1 friend class OsclScheduler [friend]

Reimplemented from [OsclExecSchedulerCommonBase](#).

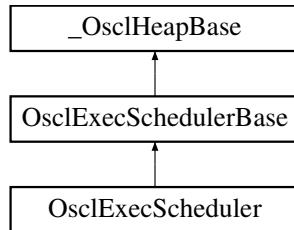
The documentation for this class was generated from the following file:

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

## 7.134 OsclExecSchedulerBase Class Reference

```
#include <oscl_scheduler_types.h>
```

Inheritance diagram for OsclExecSchedulerBase::



### Friends

- class [OsclExecScheduler](#)
- class [OsclCoeActiveScheduler](#)
- class [PVActiveBase](#)

#### 7.134.1 Detailed Description

OsclActiveSchedulerBase is the base for [OsclExecScheduler](#). The non-Symbian OsclActiveSchedulerBase class is functionally similar to a subset of Symbian CActiveScheduler class.

#### 7.134.2 Friends And Related Function Documentation

**7.134.2.1 friend class OsclCoeActiveScheduler [friend]**

**7.134.2.2 friend class OsclExecScheduler [friend]**

**7.134.2.3 friend class PVActiveBase [friend]**

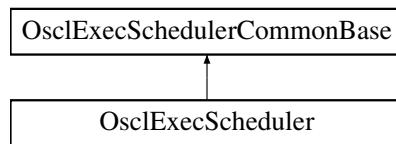
The documentation for this class was generated from the following file:

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

## 7.135 OsclExecSchedulerCommonBase Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for OsclExecSchedulerCommonBase::



### Public Methods

- OSCL\_IMPORT\_REF void [StartScheduler](#) (OsclSemaphore \*sem=NULL)
- OSCL\_IMPORT\_REF void [StopScheduler](#) ()
- OSCL\_IMPORT\_REF void [SuspendScheduler](#) ()
- OSCL\_IMPORT\_REF void [ResumeScheduler](#) ()
- OSCL\_IMPORT\_REF void [StartNativeScheduler](#) ()

### Static Public Methods

- OSCL\_IMPORT\_REF [OsclNameString< PVSCHEDNAMELEN > \\*](#) [GetName](#) ()
- OSCL\_IMPORT\_REF uint32 [GetId](#) ()

### Protected Types

- enum [TOtherExecStats](#) { [EOtherExecStats\\_WaitTime](#), [EOtherExecStats\\_QueueTime](#), [EOtherExecStats\\_NativeOS](#), [EOtherExecStats\\_ReleaseTime](#), [EOtherExecStats\\_Last](#) }

### Protected Methods

- virtual ~[OsclExecSchedulerCommonBase](#) ()
- void [InstallScheduler](#) ()
- void [UninstallScheduler](#) ()
- void [Error](#) (int32 anError) const
- [OsclExecSchedulerCommonBase](#) ([Oscl\\_DefAlloc](#) \*)
- virtual void [ConstructL](#) (const char \*name, int)
- void [BeginScheduling](#) (bool blocking, bool native)
- void [EndScheduling](#) ()
- void [BlockingLoopL](#) ()
- bool [IsStarted](#) ()
- bool [IsInstalled](#) ()
- void [AddToExecTimerQ](#) ([PVActiveBase](#) \*active, uint32)
- void [PendComplete](#) ([PVActiveBase](#) \*, int32 aReason, [TPVThreadContext](#) aContext)
- void [RequestCanceled](#) ([PVActiveBase](#) \*)
- [PVActiveBase](#) \* [UpdateTimers](#) (uint32 &aDelay)
- [PVActiveBase](#) \* [UpdateTimersMsec](#) (uint32 &aDelay)
- [PVActiveBase](#) \* [WaitForReadyAO](#) ()

- void [CallRunExec \(PVActiveBase \\*\)](#)
- void [ConstructStatQ \(\)](#)
- void [BeginStats \(\)](#)
- void [EndStats \(\)](#)
- void [CleanupStatQ \(\)](#)
- PVActiveBase \* [FindPVBase \(PVActiveBase \\*active, OsclDoubleList< PVActiveBase > &\)](#)
- void [CleanupExecQ \(\)](#)
- void [InitExecQ \(int\)](#)
- void [ResetLogPerf \(\)](#)
- void [IncLogPerf \(uint32\)](#)

## Static Protected Methods

- OsclExecSchedulerCommonBase \* [GetScheduler \(\)](#)
- OsclExecSchedulerCommonBase \* [SetScheduler \(OsclExecSchedulerCommonBase \\*\)](#)
- void [ShowStats \(PVActiveStats \\*active\)](#)
- void [ShowSummaryStats \(PVActiveStats \\*active, PVLogger \\*, int64, int64 &, float &\)](#)

## Protected Attributes

- bool [iBlockingMode](#)
- bool [iNativeMode](#)
- PVSchedulerStopper \* [iStopper](#)
- OsclNoYieldMutex [iStopperCrit](#)
- PVThreadContext [iThreadContext](#)
- OsclNameString< PVSCHEDNAMELEN > [iName](#)
- bool [iDoStop](#)
- bool [iDoSuspend](#)
- bool [iSuspended](#)
- OsclSemaphore [iResumeSem](#)
- OsclErrorTrapImp \* [iErrorTrapImp](#)
- OsclReadyQ [iReadyQ](#)
- OsclTimerQ [iExecTimerQ](#)
- uint32 [iNumAOAdded](#)
- OsclDoubleList< PVActiveStats > [iPVStatQ](#)
- PVActiveStats \* [iOtherExecStats](#) [EOtherExecStats\_Last]
- uint8 \* [iTTotalTicksTemp](#)
- int64 [iGrandTotalTicks](#)
- float [iTTotalPercent](#)
- uint32 [iTime](#)
- int32 [iDelta](#)
- PVActiveStats \* [iPVStats](#)
- PVLogger \* [iLogger](#)
- PVLogger \* [iDebugLogger](#)
- char \* [iLogPerfIndentStr](#)
- int32 [iLogPerfIndentStrLen](#)
- uint32 [iLogPerfTotal](#)
- Oscl\_DefAlloc \* [iAlloc](#)
- OsclMemAllocator [iDefAlloc](#)

## Static Protected Attributes

- const uint32 [iTTimeCompareThreshold](#)

## Friends

- class [OsclScheduler](#)
- class [PVThreadContext](#)
- class [OsclCoeActiveScheduler](#)
- class [OsclTimerCompare](#)
- class [OsclReadyQ](#)
- class [OsclError](#)
- class [PVActiveStats](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [PVActiveBase](#)
- class [PVSchedulerStopper](#)
- class [OsclExecScheduler](#)

## 7.135.1 Member Enumeration Documentation

### 7.135.1.1 enum OsclExecSchedulerCommonBase::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 OsclExecSchedulerCommonBase::~OsclExecSchedulerCommonBase ()**  
[protected, virtual]
- 7.135.2.2 **OsclExecSchedulerCommonBase::OsclExecSchedulerCommonBase (Oscl\_DefAlloc \*)**  
[protected]

## 7.135.3 Member Function Documentation

- 7.135.3.1 **void OsclExecSchedulerCommonBase::AddToExecTimerQ (PVActiveBase \* *active*, uint32)** [protected]
- 7.135.3.2 **void OsclExecSchedulerCommonBase::BeginScheduling (bool *blocking*, bool *native*)**  
[protected]
- 7.135.3.3 **void OsclExecSchedulerCommonBase::BeginStats ()** [protected]
- 7.135.3.4 **void OsclExecSchedulerCommonBase::BlockingLoopL ()** [protected]
- 7.135.3.5 **void OsclExecSchedulerCommonBase::CallRunExec (PVActiveBase \*)** [protected]
- 7.135.3.6 **void OsclExecSchedulerCommonBase::CleanupExecQ ()** [protected]
- 7.135.3.7 **void OsclExecSchedulerCommonBase::CleanupStatQ ()** [protected]
- 7.135.3.8 **virtual void OsclExecSchedulerCommonBase::ConstructL (const char \* *name*, int)**  
[protected, virtual]
- 7.135.3.9 **void OsclExecSchedulerCommonBase::ConstructStatQ ()** [protected]
- 7.135.3.10 **void OsclExecSchedulerCommonBase::EndScheduling ()** [protected]
- 7.135.3.11 **void OsclExecSchedulerCommonBase::EndStats ()** [protected]
- 7.135.3.12 **void OsclExecSchedulerCommonBase::Error (int32 *anError*) const** [protected]
- 7.135.3.13 **PVActiveBase\* OsclExecSchedulerCommonBase::FindPVBase (PVActiveBase \* *active*, OsclDoubleList< PVActiveBase > &)** [protected]
- 7.135.3.14 **OSCL\_IMPORT\_REF uint32 OsclExecSchedulerCommonBase::GetId ()** [static]

Get numeric ID of current thread.

- 7.135.3.15 **OSCL\_IMPORT\_REF OsclNameString< PVSCHEDNAMELEN >\* OsclExecSchedulerCommonBase::GetName ()** [static]

Get name of scheduler for current thread.

- 7.135.3.16** `OsclExecSchedulerCommonBase* OsclExecSchedulerCommonBase::GetScheduler ()` [static, protected]
- 7.135.3.17** `void OsclExecSchedulerCommonBase::IncLogPerf (uint32) [protected]`
- 7.135.3.18** `void OsclExecSchedulerCommonBase::InitExecQ (int) [protected]`
- 7.135.3.19** `void OsclExecSchedulerCommonBase::InstallScheduler () [protected]`
- 7.135.3.20** `bool OsclExecSchedulerCommonBase::IsInstalled () [inline, protected]`
- 7.135.3.21** `bool OsclExecSchedulerCommonBase::IsStarted () [protected]`
- 7.135.3.22** `void OsclExecSchedulerCommonBase::PendComplete (PVActiveBase *, int32 aReason, TPVThreadContext aContext) [protected]`
- 7.135.3.23** `void OsclExecSchedulerCommonBase::RequestCanceled (PVActiveBase *) [protected]`
- 7.135.3.24** `void OsclExecSchedulerCommonBase::ResetLogPerf () [protected]`
- 7.135.3.25** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::ResumeScheduler ()`

Resume scheduling immediately. This API only applies to a blocking loop scheduler.

- 7.135.3.26** `OsclExecSchedulerCommonBase* OsclExecSchedulerCommonBase::SetScheduler (OsclExecSchedulerCommonBase *) [static, protected]`
- 7.135.3.27** `void OsclExecSchedulerCommonBase::ShowStats (PVActiveStats * active) [static, protected]`
- 7.135.3.28** `void OsclExecSchedulerCommonBase::ShowSummaryStats (PVActiveStats * active, PVLogger *, int64, int64 &, float &) [static, protected]`
- 7.135.3.29** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::StartNativeScheduler ()`

Start the OS native scheduling loop. This is an alternative to the PV scheduling loop. To stop the native scheduler, use the StopScheduler API.

- 7.135.3.30** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::StartScheduler (OsclSemaphore * sem = NULL)`

Start scheduling. This call blocks until scheduler is stopped or an error occurs.

**Parameters:**

***sem***: optional startup semaphore. If provided, the scheduler will signal this semaphore when the startup has progressed to the point that it's safe to call StopScheduler or SuspendScheduler from another thread.

**7.135.3.31 OSCL\_IMPORT\_REF void OsclExecSchedulerCommonBase::StopScheduler ()**

Stop scheduling. This API may be called from the scheduling thread or some other thread.

**7.135.3.32 OSCL\_IMPORT\_REF void OsclExecSchedulerCommonBase::SuspendScheduler ()**

Suspend scheduling when the current Run is complete. This API only applies to a blocking loop scheduler.

**7.135.3.33 void OsclExecSchedulerCommonBase::UninstallScheduler () [protected]****7.135.3.34 PVActiveBase\* OsclExecSchedulerCommonBase::UpdateTimers (uint32 & aDelay) [protected]****7.135.3.35 PVActiveBase\* OsclExecSchedulerCommonBase::UpdateTimersMsec (uint32 & aDelay) [protected]****7.135.3.36 PVActiveBase\* OsclExecSchedulerCommonBase::WaitForReadyAO () [protected]****7.135.4 Friends And Related Function Documentation****7.135.4.1 friend class OsclActiveObject [friend]****7.135.4.2 friend class OsclCoeActiveScheduler [friend]****7.135.4.3 friend class OsclError [friend]****7.135.4.4 friend class OsclExecScheduler [friend]****7.135.4.5 friend class OsclReadyQ [friend]****7.135.4.6 friend class OsclScheduler [friend]**

Reimplemented in [OsclExecScheduler](#).



7.135.4.7 friend class OsclTimerCompare [friend]

7.135.4.8 friend class OsclTimerObject [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 **Oscl\_DefAlloc\*** OsclExecSchedulerCommonBase::iAlloc [protected]

7.135.5.2 bool OsclExecSchedulerCommonBase::iBlockingMode [protected]

7.135.5.3 **PVLogger\*** OsclExecSchedulerCommonBase::iDebugLogger [protected]

7.135.5.4 **OsclMemAllocator** OsclExecSchedulerCommonBase::iDefAlloc [protected]

7.135.5.5 int32 OsclExecSchedulerCommonBase::iDelta [protected]

7.135.5.6 bool OsclExecSchedulerCommonBase::iDoStop [protected]

7.135.5.7 bool OsclExecSchedulerCommonBase::iDoSuspend [protected]

7.135.5.8 **OsclErrorTrapImp\*** OsclExecSchedulerCommonBase::iErrorTrapImp  
[protected]

7.135.5.9 **OsclTimerQ** OsclExecSchedulerCommonBase::iExecTimerQ [protected]

7.135.5.10 **int64** OsclExecSchedulerCommonBase::iGrandTotalTicks [protected]

7.135.5.11 **PVLogger\*** OsclExecSchedulerCommonBase::iLogger [protected]

7.135.5.12 char\* OsclExecSchedulerCommonBase::iLogPerfIndentStr [protected]

7.135.5.13 int32 OsclExecSchedulerCommonBase::iLogPerfIndentStrLen [protected]

7.135.5.14 uint32 OsclExecSchedulerCommonBase::iLogPerfTotal [protected]

7.135.5.15 **OsclNameString<PVSCHEDEXNAMELEN>** OsclExecSchedulerCommonBase::iName  
[protected]

7.135.5.16 bool OsclExecSchedulerCommonBase::iNativeMode [protected]

7.135.5.17 uint32 OsclExecSchedulerCommonBase::iNumAOAdded [protected]

7.135.5.18 **PVActiveStats\*** OsclExecSchedulerCommonBase::iOtherExecStats[EOtherExecStats\_-  
Last] [protected]

7.135.5.19 **OsclDoubleList<PVActiveStats>** OsclExecSchedulerCommonBase::iPVStatQ

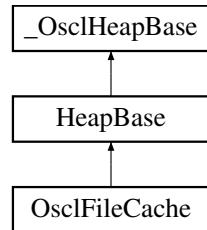
[protected]

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

## 7.136 OsclFileCache Class Reference

```
#include <oscl_file_cache.h>
```

Inheritance diagram for OsclFileCache::



### Public Methods

- [OsclFileCache \(Oscl\\_File &aContainer\)](#)
- [~OsclFileCache \(\)](#)
- int32 [Open \(uint32 mode, uint32 cache\\_size\)](#)
- void [Close \(\)](#)
- uint32 [Read \(void \\*outputBuffer, uint32 size, uint32 numelements\)](#)
- uint32 [Write \(const void \\*inputBuffer, uint32 size, uint32 numelements\)](#)
- [TOsclFileOffset FileSize \(\)](#)
- int32 [Seek \(TOsclFileOffset offset, Oscl\\_File::seek\\_type origin\)](#)
- [TOsclFileOffset Tell \(\)](#)
- int32 [Flush \(\)](#)
- int32 [EndOfFile \(\)](#)
- OSCL\_IMPORT\_REF [OsclFileCacheBuffer \\* AddFixedCache \(const Oscl\\_File::OsclFixedCacheParam &\)](#)

### Data Fields

- [OsclFileCacheBuffer \\_movableCache](#)
- [Oscl\\_Vector< OsclFileCacheBuffer, OsclMemAllocator > \\_fixedCaches](#)

### Friends

- class [OsclFileCacheBuffer](#)

### 7.136.1 Constructor & Destructor Documentation

7.136.1.1 **OsclFileCache::OsclFileCache ([Oscl\\_File](#) & *aContainer*)**

7.136.1.2 **OsclFileCache::~OsclFileCache ()**

### 7.136.2 Member Function Documentation

7.136.2.1 **OSCL\_IMPORT\_REF [OsclFileCacheBuffer](#)\* OsclFileCache::AddFixedCache (const [Oscl\\_File::OsclFixedCacheParam](#) &)**

7.136.2.2 **void OsclFileCache::Close ()**

7.136.2.3 **int32 OsclFileCache::EndOfFile () [inline]**

7.136.2.4 **[TOsclFileOffset](#) OsclFileCache::FileSize () [inline]**

7.136.2.5 **int32 OsclFileCache::Flush ()**

7.136.2.6 **int32 OsclFileCache::Open (uint32 *mode*, uint32 *cache\_size*)**

7.136.2.7 **uint32 OsclFileCache::Read (void \* *outputBuffer*, uint32 *size*, uint32 *numelements*)**

7.136.2.8 **int32 OsclFileCache::Seek ([TOsclFileOffset](#) *offset*, [Oscl\\_File::seek\\_type](#) *origin*)**

7.136.2.9 **[TOsclFileOffset](#) OsclFileCache::Tell () [inline]**

7.136.2.10 **uint32 OsclFileCache::Write (const void \* *inputBuffer*, uint32 *size*, uint32 *numelements*)**

### 7.136.3 Friends And Related Function Documentation

7.136.3.1 **friend class OsclFileCacheBuffer [friend]**

### 7.136.4 Field Documentation

7.136.4.1 **[Oscl\\_Vector](#)<[OsclFileCacheBuffer](#), [OsclMemAllocator](#)> OsclFileCache::\_fixedCaches**

7.136.4.2 **[OsclFileCacheBuffer](#) OsclFileCache::\_movableCache**

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

- [oscl\\_file\\_cache.h](#)

## 7.137 OsclFileCacheBuffer Class Reference

```
#include <oscl_file_cache.h>
```

### Public Methods

- [OsclFileCacheBuffer \(\)](#)
- [int32 SetPosition \(TOsclFileOffset pos\)](#)
- [int32 PrepRead \(\)](#)
- [int32 PrepWrite \(\)](#)
- [int32 WriteUpdatesToFile \(\)](#)
- [int32 FillFromFile \(uint32, uint32\)](#)
- [bool IsUpdated \(\)](#)
- [bool Contains \(TOsclFileOffset pos\)](#)
- [bool Preceeds \(TOsclFileOffset pos\)](#)

### Data Fields

- [OsclFileCache \\* iContainer](#)
- [bool isFixed](#)
- [uint32 capacity](#)
- [uint32 usableSize](#)
- [uint8 \\* pBuffer](#)
- [TOsclFileOffset filePosition](#)
- [uint32 currentPos](#)
- [uint32 endPos](#)
- [uint32 updateStart](#)
- [uint32 updateEnd](#)

## 7.137.1 Constructor & Destructor Documentation

**7.137.1.1** `OsclFileCacheBuffer::OsclFileCacheBuffer () [inline]`

## 7.137.2 Member Function Documentation

**7.137.2.1** `bool OsclFileCacheBuffer::Contains (TOsclFileOffset pos) [inline]`

**7.137.2.2** `int32 OsclFileCacheBuffer::FillFromFile (uint32, uint32)`

**7.137.2.3** `bool OsclFileCacheBuffer::IsUpdated () [inline]`

**7.137.2.4** `bool OsclFileCacheBuffer::Preceeds (TOsclFileOffset pos) [inline]`

**7.137.2.5** `int32 OsclFileCacheBuffer::PreRead ()`

**7.137.2.6** `int32 OsclFileCacheBuffer::PrepWrite ()`

**7.137.2.7** `int32 OsclFileCacheBuffer::SetPosition (TOsclFileOffset pos)`

**7.137.2.8** `int32 OsclFileCacheBuffer::WriteUpdatesToFile ()`

## 7.137.3 Field Documentation

**7.137.3.1** `uint32 OsclFileCacheBuffer::capacity`

**7.137.3.2** `uint32 OsclFileCacheBuffer::currentPos`

**7.137.3.3** `uint32 OsclFileCacheBuffer::endPos`

**7.137.3.4** `TOsclFileOffset OsclFileCacheBuffer::filePosition`

**7.137.3.5** `OsclFileCache* OsclFileCacheBuffer::iContainer`

**7.137.3.6** `bool OsclFileCacheBuffer::isFixed`

**7.137.3.7** `uint8* OsclFileCacheBuffer::pBuffer`

**7.137.3.8** `uint32 OsclFileCacheBuffer::updateEnd`

**7.137.3.9** `uint32 OsclFileCacheBuffer::updateStart`

**7.137.3.10** `uint32 OsclFileCacheBuffer::usableSize`

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

- [oscl\\_file\\_cache.h](#)

## 7.138 OsclFileHandle Class Reference

```
#include <oscl_file_handle.h>
```

### Public Methods

- [OsclFileHandle \(TOsclFileHandle aHandle\)](#)
- [OsclFileHandle \(const OsclFileHandle &aHandle\)](#)
- [TOsclFileHandle Handle \(\) const](#)

### Friends

- class [Oscl\\_File](#)

#### 7.138.1 Detailed Description

OsclFileHandle is a container for a handle to a previously-opened file.

#### 7.138.2 Constructor & Destructor Documentation

**7.138.2.1 OsclFileHandle::OsclFileHandle (TOsclFileHandle *aHandle*) [inline]**

**7.138.2.2 OsclFileHandle::OsclFileHandle (const OsclFileHandle & *aHandle*) [inline]**

#### 7.138.3 Member Function Documentation

**7.138.3.1 TOsclFileHandle OsclFileHandle::Handle () const [inline]**

#### 7.138.4 Friends And Related Function Documentation

**7.138.4.1 friend class Oscl\_File [friend]**

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

- [oscl\\_file\\_handle.h](#)

## 7.139 OsclFileManager 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 **OsclGetFileSize** (const **oscl\_wchar** \*aFileName, **uint64** &aFileSize)
- OSCL\_IMPORT\_REF bool **OsclGetFileSize** (const char \*aFileName, **uint64** &aFileSize)
- OSCL\_IMPORT\_REF bool **OsclGetFileCreationTime** (const **oscl\_wchar** \*aFileName, **uint64** &aFileCreationTime)
- OSCL\_IMPORT\_REF bool **OsclGetFileCreationTime** (const char \*aFileName, **uint64** &aFileCreationTime)
- OSCL\_IMPORT\_REF bool **OsclGetFileLastAccessTime** (const **oscl\_wchar** \*aFileName, **uint64** &aFileLastAccessTime)
- OSCL\_IMPORT\_REF bool **OsclGetFileLastAccessTime** (const char \*aFileName, **uint64** &aFileLastAccessTime)
- OSCL\_IMPORT\_REF bool **OsclGetFileLastWriteTime** (const **oscl\_wchar** \*aFileName, **uint64** &aFileLastWriteTime)
- OSCL\_IMPORT\_REF bool **OsclGetFileLastWriteTime** (const char \*aFileName, **uint64** &aFileLastWriteTime)
- OSCL\_IMPORT\_REF bool **OsclGetFileAttributes** (const **oscl\_wchar** \*aFileName, **uint32** &aFileAttributes)
- OSCL\_IMPORT\_REF bool **OsclGetFileAttributes** (const char \*aFileName, **uint32** &aFileAttributes)
- OSCL\_IMPORT\_REF void **OsclExtractFilenameFromFullPath** (const char \*aPath, char \*&aFileName)
- OSCL\_IMPORT\_REF void **OsclExtractFilenameFromFullPath** (const **oscl\_wchar** \*aPath, **oscl\_wchar** \*&aFileName)

### 7.139.1 Member Enumeration Documentation

#### 7.139.1.1 enum OsclFileManager::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 OsclFileManager::OsclExtractFilenameFromFullPath  
(const oscl\_wchar \* aPath, oscl\_wchar \*& aFileName) [static]**

**7.139.2.2 OSCL\_IMPORT\_REF void OsclFileManager::OsclExtractFilenameFromFullPath  
(const char \* aPath, char \*& aFileName) [static]**

OsclExtractFilenameFromFullPath 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 OsclFileManager::OsclGetFileAttributes (const char \*  
aFileName, uint32 & aFileAttributes) [static]**

OsclGetFileAttributes 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 OsclFileManager::OsclGetFileAttributes (const oscl\_wchar  
\* aFileName, uint32 & aFileAttributes) [static]**

OsclGetFileAttributes 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 OsclFileManager::OsclGetFileCreationTime (const char \*  
aFileName, uint64 & aFileCreationTime) [static]**

OsclGetFileCreationTime 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 OsclFileManager::OsclGetFileCreationTime (const  
oscl\_wchar \* aFileName, uint64 & aFileCreationTime) [static]**

OsclGetFileCreationTime 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 OsclFileManager::OsclGetFileLastAccessTime (const char  
\* aFileName, uint64 & aFileLastAccessTime) [static]**

OsclGetFileLastAccessTime 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 OsclFileManager::OsclGetFileLastAccessTime (const oscl\_wchar \* aFileName, uint64 & aFileLastAccessTime) [static]**

OsclGetFileLastAccessTime 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 OsclFileManager::OsclGetFileLastWriteTime (const char \* aFileName, uint64 & aFileLastWriteTime) [static]**

OsclGetFileLastWriteTime 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 OsclFileManager::OsclGetFileLastWriteTime (const oscl\_wchar \* aFileName, uint64 & aFileLastWriteTime) [static]**

OsclGetFileLastWriteTime 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 OsclFileManager::OsclGetFileSize (const char \* aFileName, uint64 & aFileSize) [static]**

OsclGetFileSize 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 OsclFileManager::OsclGetFileSize (const oscl\_wchar \*  
*aFileName*, uint64 & *aFileSize*) [static]**

OsclGetFileSize 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 OsclFileStats Class Reference

```
#include <oscl_file_stats.h>
```

### Public Methods

- [OsclFileStats \(Oscl\\_File \\*c\)](#)
- void [Start \(uint32 &aTicks\)](#)
- void [End \(TOsclFileOp aOp, uint32 aStart, uint32 aParam=0, TOsclFileOffset aParam2=0\)](#)
- void [Log \(TOsclFileOp, PVLogger \\*, uint32\)](#)
- void [LogAll \(PVLogger \\*, uint32\)](#)

#### 7.140.1 Constructor & Destructor Documentation

##### 7.140.1.1 OsclFileStats::OsclFileStats ([Oscl\\_File](#) \* *c*)

#### 7.140.2 Member Function Documentation

##### 7.140.2.1 void OsclFileStats::End ([TOsclFileOp](#) *aOp*, [uint32](#) *aStart*, [uint32](#) *aParam* = 0, [TOsclFileOffset](#) *aParam2* = 0)

##### 7.140.2.2 void OsclFileStats::Log ([TOsclFileOp](#), [PVLogger](#) \*, [uint32](#))

##### 7.140.2.3 void OsclFileStats::LogAll ([PVLogger](#) \*, [uint32](#))

##### 7.140.2.4 void OsclFileStats::Start ([uint32](#) & *aTicks*)

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

- [oscl\\_file\\_stats.h](#)

## 7.141 OsclFileStatsItem Class Reference

```
#include <oscl_file_stats.h>
```

### Data Fields

- uint32 [iOpCount](#)
- uint64 [iParam](#)
- [TOsclFileOffset](#) [iParam2](#)
- uint32 [iStartTick](#)
- uint32 [iTTotalTicks](#)

#### 7.141.1 Field Documentation

**7.141.1.1 uint32 OsclFileStatsItem::iOpCount**

**7.141.1.2 uint64 OsclFileStatsItem::iParam**

**7.141.1.3 TOsclFileOffset OsclFileStatsItem::iParam2**

**7.141.1.4 uint32 OsclFileStatsItem::iStartTick**

**7.141.1.5 uint32 OsclFileStatsItem::iTTotalTicks**

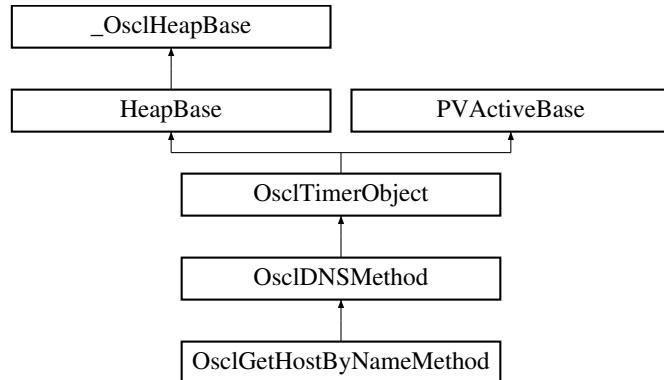
The documentation for this class was generated from the following file:

- [oscl\\_file\\_stats.h](#)

## 7.142 OsclGetHostByNameMethod Class Reference

```
#include <oscl_dns_gethostbyname.h>
```

Inheritance diagram for OsclGetHostByNameMethod::



### Public Methods

- `~OsclGetHostByNameMethod ()`
- `TPVDNSEvent GetHostByName (char *name, OsclNetworkAddress *addr, int32 aTimeout, Oscl_-Vector< OsclNetworkAddress, OsclMemAllocator > *aAddressList)`

### Static Public Methods

- `OsclGetHostByNameMethod * NewL (Oscl_DefAlloc &a, OsclDNSI *aDNS, OsclDNSObserver *aObserver, uint32 aId)`

#### 7.142.1 Constructor & Destructor Documentation

##### 7.142.1.1 OsclGetHostByNameMethod::~OsclGetHostByNameMethod ()

#### 7.142.2 Member Function Documentation

##### 7.142.2.1 TPVDNSEvent OsclGetHostByNameMethod::GetHostByName (char \* name, OsclNetworkAddress \* addr, int32 aTimeout, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \* aAddressList)

##### 7.142.2.2 OsclGetHostByNameMethod\* OsclGetHostByNameMethod::NewL (Oscl\_DefAlloc &a, OsclDNSI \*aDNS, OsclDNSObserver \*aObserver, uint32 aId) [static]

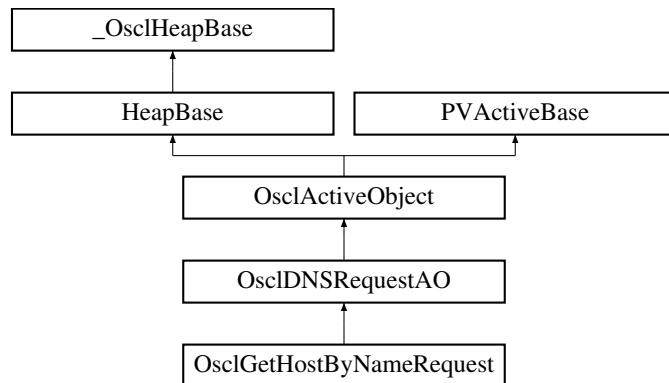
The documentation for this class was generated from the following file:

- `oscl_dns_gethostbyname.h`

## 7.143 OsclGetHostByNameRequest Class Reference

```
#include <oscl_dns_gethostbyname.h>
```

Inheritance diagram for OsclGetHostByNameRequest::



### Friends

- class [OsclGetHostByNameMethod](#)

#### 7.143.1 Friends And Related Function Documentation

##### 7.143.1.1 friend class OsclGetHostByNameMethod [friend]

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

- [oscl\\_dns\\_gethostbyname.h](#)

## 7.144 OsclInit Class Reference

```
#include <oscl_init.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [Init](#) (int32 &aError, const [OsclSelect](#) \*aSelect=NULL)
- OSCL\_IMPORT\_REF void [Cleanup](#) (int32 &aError, const [OsclSelect](#) \*aSelect=NULL)

#### 7.144.1 Detailed Description

Per-thread oscl initialization and cleanup.

#### 7.144.2 Member Function Documentation

##### 7.144.2.1 OSCL\_IMPORT\_REF void OsclInit::Cleanup (int32 & aError, const [OsclSelect](#) \* aSelect = NULL) [static]

This routine cleans up the Oscl modules in the calling thread.

###### Parameters:

*err*: (output) error code of any leave that occurs in initialization.

*config*: (input param) optional set of initialization parameters. If null, then full initialization with default parameters will be performed. For proper cleanup, the parameters should match the ones used during the Init call.

##### 7.144.2.2 OSCL\_IMPORT\_REF void OsclInit::Init (int32 & aError, const [OsclSelect](#) \* aSelect = NULL) [static]

This routine initializes the Oscl modules in the calling thread.

###### Parameters:

*err*: (output) error code of any leave that occurs in initialization.

*config*: (input param) optional set of initialization parameters. If null, then full initialization with default parameters will be performed.

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

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

## 7.145 OsclInteger64Transport Struct Reference

```
#include <oscl_int64_utils.h>
```

### Data Fields

- uint32 [iHigh](#)
- uint32 [iLow](#)

#### 7.145.1 Detailed Description

OsclInteger64Transport Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

#### 7.145.2 Field Documentation

##### 7.145.2.1 uint32 OsclInteger64Transport::iHigh

##### 7.145.2.2 uint32 OsclInteger64Transport::iLow

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

- [oscl\\_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

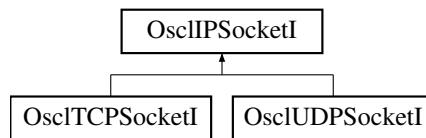
The documentation for this class was generated from the following file:

- [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 OsclIPSocketI::~OsclIPSocketI () [inline, virtual]`

7.147.1.2 `OsclIPSocketI::OsclIPSocketI (Oscl_DefAlloc & a) [inline, protected]`

### 7.147.2 Member Function Documentation

7.147.2.1 `Oscl_DefAlloc& OsclIPSocketI::Alloc () [inline]`

7.147.2.2 `int32 OsclIPSocketI::Bind (OsclNetworkAddress & aAddress)`

7.147.2.3 `virtual int32 OsclIPSocketI::Close () [pure virtual]`

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

7.147.2.4 `void OsclIPSocketI::ConstructL (OsclSocketObserver * aObs, OsclSocketI * aSock, OsclSocketServI * aServ, uint32 aId) [protected]`

7.147.2.5 `int32 OsclIPSocketI::GetPeerName (OsclNetworkAddress & peerName)`

7.147.2.6 `virtual uint8* OsclIPSocketI::GetRecvData (int32 * aLength) [pure virtual]`

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

7.147.2.7 `virtual uint8* OsclIPSocketI::GetSendData (int32 * aLength) [pure virtual]`

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

7.147.2.8 `int32 OsclIPSocketI::Join (OsclNetworkAddress & aAddress)`

7.147.2.9 `int32 OsclIPSocketI::SetOptionToReuseAddress ()`

7.147.2.10 `int32 OsclIPSocketI::SetRecvBufferSize (uint32 size)`

7.147.2.11 `int32 OsclIPSocketI::SetTOS (const OsclSocketTOS & aTOS)`

7.147.2.12 `OsclSocketServI* OsclIPSocketI::SocketServ () [inline]`

7.147.2.13 `void OsclIPSocketI::ThreadLogoff ()`

Reimplemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

7.147.2.14 void OsclIPSocketI::ThreadLogon ([OsclSocketObserver](#) \* *aObs*, [OsclSocketServI](#) \* *aServ*)

### 7.147.3 Friends And Related Function Documentation

7.147.3.1 friend class [OsclSocketMethod](#) [friend]

7.147.3.2 friend class [OsclSocketRequestAO](#) [friend]

### 7.147.4 Field Documentation

7.147.4.1 [OsclNetworkAddress](#) [OsclIPSocketI::iAddress](#) [protected]

7.147.4.2 [Oscl\\_DefAlloc&](#) [OsclIPSocketI::iAlloc](#) [protected]

7.147.4.3 uint32 [OsclIPSocketI::iId](#) [protected]

7.147.4.4 [PVLogger\\*](#) [OsclIPSocketI::iLogger](#) [protected]

7.147.4.5 [OsclSocketObserver\\*](#) [OsclIPSocketI::iObserver](#) [protected]

7.147.4.6 [OsclSocketI\\*](#) [OsclIPSocketI::iSocket](#) [protected]

7.147.4.7 [OsclSocketServI\\*](#) [OsclIPSocketI::iSocketServ](#) [protected]

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

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

## 7.148 OsclJump Class Reference

```
#include <oscl_error_imp_jumps.h>
```

### Public Methods

- void [Jump](#) (int a)
- jmp\_buf \* [Top](#) ()
- [~OsclJump](#) ()

### Static Public Methods

- OSCL\_IMPORT\_REF void [StaticJump](#) (int a)

### Friends

- class [OsclErrorTrapImp](#)

#### 7.148.1 Constructor & Destructor Documentation

**7.148.1.1 OsclJump::~OsclJump () [inline]**

#### 7.148.2 Member Function Documentation

**7.148.2.1 void OsclJump::Jump (int a) [inline]**

**7.148.2.2 OSCL\_IMPORT\_REF void OsclJump::StaticJump (int a) [static]**

**7.148.2.3 jmp\_buf\* OsclJump::Top () [inline]**

#### 7.148.3 Friends And Related Function Documentation

**7.148.3.1 friend class OsclErrorTrapImp [friend]**

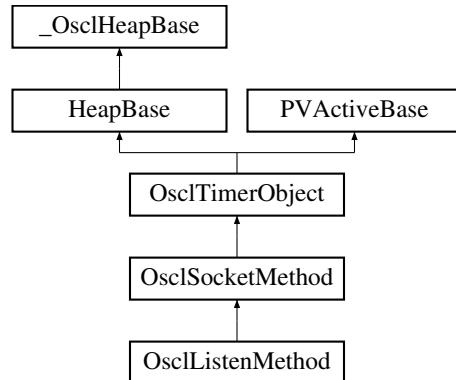
The documentation for this class was generated from the following file:

- [oscl\\_error\\_imp\\_jumps.h](#)

## 7.149 OsclListenMethod Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OsclListenMethod::



### Public Methods

- [~OsclListenMethod \(\)](#)
- [TPVSocketEvent Listen \(uint32 qsize, int32 aTimeout\)](#)
- [OsclListenRequest \\* ListenRequest \(\)](#)

### Static Public Methods

- [OsclListenMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 7.149.1 Constructor & Destructor Documentation

##### 7.149.1.1 OsclListenMethod::~OsclListenMethod ()

#### 7.149.2 Member Function Documentation

##### 7.149.2.1 TPVSocketEvent OsclListenMethod::Listen (uint32 qsize, int32 aTimeout)

##### 7.149.2.2 OsclListenRequest\* OsclListenMethod::ListenRequest () [inline]

##### 7.149.2.3 OsclListenMethod\* OsclListenMethod::NewL (OsclIPSocketI &c) [static]

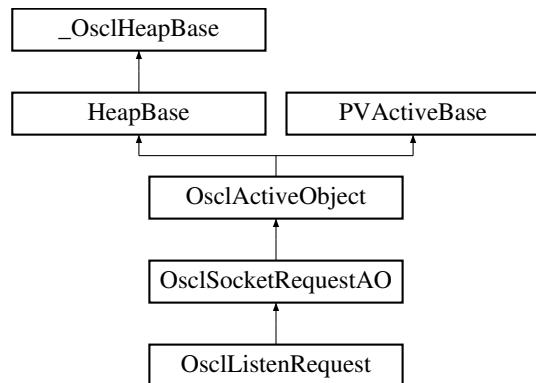
The documentation for this class was generated from the following file:

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

## 7.150 OsclListenRequest Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OsclListenRequest::



### Public Methods

- [OsclListenRequest \(OsclSocketMethod &c\)](#)
- 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 OsclListenRequest::OsclListenRequest ([OsclSocketMethod & c](#)) [inline]**

#### 7.150.3 Member Function Documentation

**7.150.3.1 void OsclListenRequest::Listen (uint32 *qsize*)**

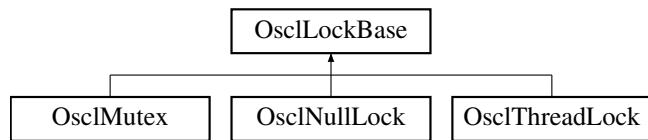
The documentation for this class was generated from the following file:

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

## 7.151 OsclLockBase Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclLockBase::



### Public Methods

- virtual void [Lock \(\)=0](#)
- virtual void [Unlock \(\)=0](#)
- virtual [~OsclLockBase \(\)](#)

#### 7.151.1 Constructor & Destructor Documentation

**7.151.1.1 virtual OsclLockBase::~OsclLockBase () [inline, virtual]**

#### 7.151.2 Member Function Documentation

**7.151.2.1 virtual void OsclLockBase::Lock () [pure virtual]**

Implemented in [OsclNullLock](#), [OsclMutex](#), and [OsclThreadLock](#).

**7.151.2.2 virtual void OsclLockBase::Unlock () [pure virtual]**

Implemented in [OsclNullLock](#), [OsclMutex](#), and [OsclThreadLock](#).

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

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

## 7.152 OsclMem 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 OsclMem::Cleanup () [static]

Per-thread cleanup of Oscl Memory @exception: Leaves on error;

##### 7.152.1.2 OSCL\_IMPORT\_REF void OsclMem::Init () [static]

Per-thread initialization of Oscl Memory

#### Parameters:

*lock*: A lock class for use with multi-threaded applications. The lock is needed in use cases where memory may be allocated in one thread and freed in another. In this case, there must be a single lock object, and its pointer must be passed to the [OsclMem::Init](#) call in each thread. If no lock is provided, the memory manager will not be thread-safe. @exception: Leaves on error

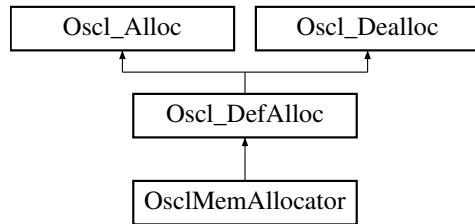
The documentation for this class was generated from the following file:

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

## 7.153 OsclMemAllocator Class Reference

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

Inheritance diagram for OsclMemAllocator::



### Public Methods

- [OsclAny \\* allocate \(const uint32 n\)](#)
- [OsclAny \\* allocate\\_fl \(const uint32 n, const char \\*file\\_name, const int line\\_num\)](#)
- void [deallocate \(OsclAny \\*p\)](#)

#### 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 [OsclAny\\* OsclMemAllocator::allocate \(const uint32 n\)](#) [inline, virtual]

This API throws an exception when malloc returns NULL. n must be greater than 0.

**Returns:**

pointer (or Leave with OsclErrNoMemory )

Implements [Oscl\\_DefAlloc](#).

##### 7.153.2.2 [OsclAny\\* OsclMemAllocator::allocate\\_fl \(const uint32 n, const char \\*file\\_name, const int line\\_num\)](#) [inline, virtual]

Reimplemented from [Oscl\\_DefAlloc](#).

##### 7.153.2.3 [void OsclMemAllocator::deallocate \(OsclAny \\*p\)](#) [inline, virtual]

Implements [Oscl\\_DefAlloc](#).

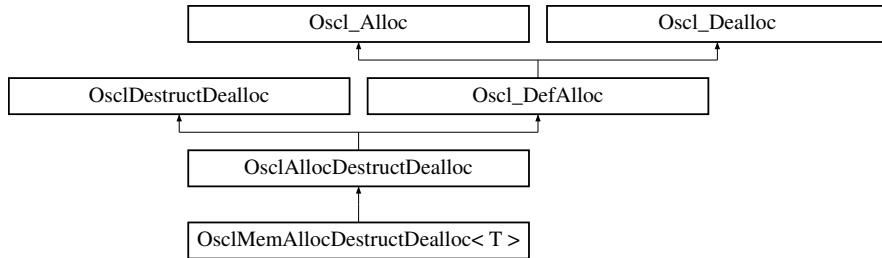
The documentation for this class was generated from the following file:

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

## 7.154 OsclMemAllocDestructDealloc< T > Class Template Reference

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

Inheritance diagram for OsclMemAllocDestructDealloc< T >::



### Public Methods

- [OsclAny \\* allocate\\_fl](#) (const uint32 size, const char \*file\_name, const int line\_num)
- [OsclAny \\* allocate](#) (const uint32 size)
- void [deallocate](#) (OsclAny \*p)
- void [destruct\\_and\\_dealloc](#) (OsclAny \*p)

#### 7.154.1 Detailed Description

`template<class T> class OsclMemAllocDestructDealloc< T >`

An [OsclAllocDestructDealloc](#) class that uses [OsclMemAllocator](#).

#### 7.154.2 Member Function Documentation

**7.154.2.1 template<class T> [OsclAny\\*](#) OsclMemAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

**7.154.2.2 template<class T> [OsclAny\\*](#) OsclMemAllocDestructDealloc< T >::allocate\_fl (const uint32 size, const char \*file\_name, const int line\_num) [inline, virtual]**

Reimplemented from [Oscl\\_DefAlloc](#).

**7.154.2.3 template<class T> void OsclMemAllocDestructDealloc< T >::deallocate ([OsclAny](#) \* p) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

**7.154.2.4 template<class T> void OsclMemAllocDestructDealloc< T >::destruct\_and\_dealloc  
(OsclAny \* p) [inline, virtual]**

Implements [OsclDestructDealloc](#).

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

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

## 7.155 OsclMemAudit Class Reference

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

### Public Methods

- [OsclMemAudit \(\)](#)
- [~OsclMemAudit \(\)](#)
- [void \\* MM\\_allocate \(const OsclMemStatsNode \\*statsNode, uint32 sizeIn, const char \\*pFileName, uint32 lineNumber, bool allocNodeTracking=false\)](#)
- [bool MM\\_deallocate \(void \\*pMemBlockIn\)](#)
- [MM\\_Stats\\_t \\* MM\\_GetStats \(const char \\*const tagIn\)](#)
- [uint32 MM\\_GetStatsInDepth \(const char \\*tagIn, MM\\_Stats\\_CB \\*array\\_ptr, uint32 max\\_nodes\)](#)
- [uint32 MM\\_GetTreeNodes \(const char \\*tagIn\)](#)
- [bool MM\\_AddTag \(const char \\*tagIn\)](#)
- [const OsclMemStatsNode \\* MM\\_GetTagName \(const char \\*tagIn\)](#)
- [const OsclMemStatsNode \\* MM\\_GetExistingTag \(const char \\*tagIn\)](#)
- [const OsclMemStatsNode \\* MM\\_GetRootNode \(\)](#)
- [uint32 MM\\_GetAllocNodeInfo \(MM\\_AllocQueryInfo \\*output\\_array, uint32 max\\_array\\_size, uint32 offset\)](#)
- [MM\\_AllocQueryInfo \\* MM\\_CreateAllocNodeInfo \(uint32 max\\_array\\_size\)](#)
- [void MM\\_ReleaseAllocNodeInfo \(MM\\_AllocQueryInfo \\*info\)](#)
- [bool MM\\_Validate \(const void \\*ptrIn\)](#)
- [uint32 MM\\_GetAllocNo \(void\)](#)
- [void MM\\_GetOverheadStats \(MM\\_AuditOverheadStats &stats\)](#)
- [uint32 MM\\_GetNumAllocNodes \(\)](#)
- [uint32 MM\\_GetMode \(void\)](#)
- [uint8 MM\\_GetPrefillPattern \(void\)](#)
- [uint32 MM\\_GetPostfillPattern \(void\)](#)
- [void MM\\_SetMode \(uint32 inMode\)](#)
- [void MM\\_SetPrefillPattern \(uint8 pattern\)](#)
- [void MM\\_SetPostfillPattern \(uint8 pattern\)](#)
- [void MM\\_SetTagLevel \(uint32 level\)](#)
- [bool MM\\_SetFailurePoint \(const char \\*tagIn, uint32 alloc\\_number\)](#)
- [void MM\\_UnsetFailurePoint \(const char \\*tagIn\)](#)
- [int32 MM\\_GetRefCount \(\)](#)
- [OsclLockBase \\* GetLock \(\)](#)

### Friends

- class [OsclMemGlobalAuditObject](#)

#### 7.155.1 Constructor & Destructor Documentation

##### 7.155.1.1 OsclMemAudit::OsclMemAudit () [inline]

Constructor, create the root node in statistics table

**7.155.1.2 OsclMemAudit::~OsclMemAudit () [inline]**

A destructor, remove all the nodes in allocation andstatistics table

**7.155.2 Member Function Documentation****7.155.2.1 OsclLockBase\* OsclMemAudit::GetLock () [inline]**

API to obtain mem lock ptr

**7.155.2.2 bool OsclMemAudit::MM\_AddTag (const char \* tagIn) [inline]**

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

true if operation succeeds;

**7.155.2.3 void\* OsclMemAudit::MM\_allocate (const OsclMemStatsNode \* statsNode, uint32 sizeIn, const char \* pFileName, uint32 lineNumber, bool allocNodeTracking = false) [inline]**

The following are APIs to \_\_nothrow\_ / const \_\_nothrow\_

**Returns:**

the memory pointer if operation succeeds.

**7.155.2.4 MM\_AllocQueryInfo\* OsclMemAudit::MM\_CreateAllocNodeInfo (uint32 max\_array\_size) [inline]****7.155.2.5 bool OsclMemAudit::MM\_deallocate (void \* pMemBlockIn) [inline]****Returns:**

true if operation succeeds;

**7.155.2.6 uint32 OsclMemAudit::MM\_GetAllocNo (void) [inline]**

API to get the current allocation number

**Returns:**

the current allocation number

**7.155.2.7** `uint32 OsclMemAudit::MM_GetAllocNodeInfo (MM_AllocQueryInfo * output_array, uint32 max_array_size, uint32 offset) [inline]`

API to query the list of alloc nodes. It copies the information into the provided output array.

**Parameters:**

*output\_array* the array where the data will be written

*max\_array\_size* the max number of output array elements

*offset* the offset into the alloc node list from which the data should begin.

**Returns:**

the number of valid nodes in the output array

**7.155.2.8** `const OsclMemStatsNode* OsclMemAudit::MM_GetExistingTag (const char * tagIn) [inline]`

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

true if operation succeeds;

**7.155.2.9** `uint32 OsclMemAudit::MM_GetMode (void) [inline]`

API to get the operating mode of the mm\_audit class.

**7.155.2.10** `uint32 OsclMemAudit::MM_GetNumAllocNodes () [inline]`

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

**7.155.2.11** `void OsclMemAudit::MM_GetOverheadStats (MM_AuditOverheadStats & stats) [inline]`

API to get the overhead statistics for the memory used by the mm\_audit class.

**7.155.2.12** `uint32 OsclMemAudit::MM_GetPostfillPattern (void) [inline]`

API to get the postfill pattern. The pattern is used to fill the memory before freeing it.

**7.155.2.13** `uint8 OsclMemAudit::MM_GetPrefillPattern (void) [inline]`

API to get the prefill pattern. The pattern is used to fill the memory before returning it to the caller.

**7.155.2.14** `int32 OsclMemAudit::MM_GetRefCount () [inline]`

**7.155.2.15** `const OsclMemStatsNode* OsclMemAudit::MM_GetRootNode () [inline]`

**7.155.2.16** `MM_Stats_t* OsclMemAudit::MM_GetStats (const char *const tagIn) [inline]`

API to get memory statistics through context string(tag)

**Returns:**

statistics pointer if operation succeeds

**7.155.2.17** `uint32 OsclMemAudit::MM_GetStatsInDepth (const char * tagIn, MM_Stats_CB * array_ptr, uint32 max_nodes) [inline]`

API to get memory statistics in detail through context string(tag) including its subtree

**Returns:**

statistics pointer array and actual number of nodes if operation succeeds

**7.155.2.18** `const OsclMemStatsNode* OsclMemAudit::MM_GetTagName (const char * tagIn) [inline]`

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

pointer to `OsclMemStatsNode` which should be passed to `MM_allocate`

**7.155.2.19** `uint32 OsclMemAudit::MM_GetTreeNodes (const char * tagIn) [inline]`

API to get the number of tree nodes including the tag node and its subtree

**Parameters:**

*tagIn* input tag

**Returns:**

the number of tree nodes ; 0 means no tag node

**7.155.2.20** `void OsclMemAudit::MM_ReleaseAllocNodeInfo (MM_AllocQueryInfo * info) [inline]`

**7.155.2.21** `bool OsclMemAudit::MM_SetFailurePoint (const char * tagIn, uint32 alloc_number) [inline]`

API to insert allocation failure deterministically according to allocation number associated with tag

**Parameters:**

*tagIn* input tag  
*alloc\_number* allocation number associated with tag

**Returns:**

true if operation succeeds;

**7.155.2.22 void OsclMemAudit::MM\_SetMode (uint32 *inMode*) [inline]**

API to set the operating mode of the mm\_audit class.

**7.155.2.23 void OsclMemAudit::MM\_SetPostfillPattern (uint8 *pattern*) [inline]**

API to set the postfill pattern.

**7.155.2.24 void OsclMemAudit::MM\_SetPrefillPattern (uint8 *pattern*) [inline]**

API to set the prefill pattern.

**7.155.2.25 void OsclMemAudit::MM\_SetTagLevel (uint32 *level*) [inline]**

API to set the maximum tag level,i.e. tag level for a.b.c.d = 4

**Parameters:**

*level* input tag level to be set

**7.155.2.26 void OsclMemAudit::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 OsclMemAudit::MM\_Validate (const void \* *ptrIn*) [inline]**

API to check the input pointer is a valid pointer to a chunk of memory

**Parameters:**

*ptrIn* input pointer to be validated

**Returns:**

true if operation succeeds;

### 7.155.3 Friends And Related Function Documentation

#### 7.155.3.1 friend class OsclMemGlobalAuditObject [friend]

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

- [oscl\\_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 = Oscl_TAlloc<T, OsclMemAllocator>> class OSCLMemAuto-  
Ptr< T, _Allocator >
```

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

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an oscl\_auto\_ptr object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The oscl\_auto\_ptr is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### 7.156.2 Constructor & Destructor Documentation

```
7.156.2.1 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::OSCLMemAutoPtr (T * inPtr = 0) [inline,  
explicit]
```

Default constructor Initializes the pointer and takes ownership.

```
7.156.2.2 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::OSCLMemAutoPtr (const OSCLMemAutoPtr<  
T > & _Y) [inline]
```

Copy constructor.

Initializes the pointer and takes ownership from another oscl\_auto\_ptr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

```
7.156.2.3 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::~OSCLMemAutoPtr () [inline]
```

Destructor.

The pointer is deleted in case this class still has ownership

### 7.156.3 Member Function Documentation

**7.156.3.1 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, \_Allocator >::allocate (oscl\_memsize\_t size) [inline]**

**7.156.3.2 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, \_Allocator >::deallocate (T \*ptr) [inline, static]**

**7.156.3.3 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> T\* OSCLMemAutoPtr< T, \_Allocator >::get () const [inline]**

[get\(\)](#) method returns the pointer, currently owned by the class.

**7.156.3.4 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> T& OSCLMemAutoPtr< T, \_Allocator >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OSCLMemAutoPtr can be used like the regular pointer that it was initialized with.

**7.156.3.5 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> T\* OSCLMemAutoPtr< T, \_Allocator >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OSCLMemAutoPtr can be used like the regular pointer that it was initialized with.

**7.156.3.6 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> OSCLMemAutoPtr< T, \_Allocator >& OSCLMemAutoPtr< T, \_Allocator >::operator=(const OSCLMemAutoPtr< T, \_Allocator > & \_Y) [inline]**

Assignment operator from an another oscl\_auto\_ptr.

**Parameters:**

*\_Y* The value parameter should be another oscl\_auto\_ptr

**Returns:**

Returns a reference to this oscl\_auto\_ptr instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the oscl\_auto\_ptr given as the input parameter. The ownership of the pointer is transferred.

**7.156.3.7 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> T\* OSCLMemAutoPtr< T, \_Allocator >::release () const [inline]**

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**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**

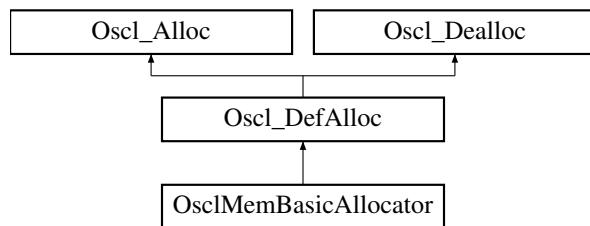
The documentation for this class was generated from the following file:

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

## 7.157 OsclMemBasicAllocator Class Reference

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

Inheritance diagram for OsclMemBasicAllocator::



### Public Methods

- [OsclAny \\* allocate \(const uint32 n\)](#)
- [void deallocate \(OsclAny \\*p\)](#)

#### 7.157.1 Detailed Description

A simple allocator class that does not use the memory management.

Note: this allocator is for internal use by Oscl only. Higher level code should use [OsclMemAllocator](#).

#### 7.157.2 Member Function Documentation

##### 7.157.2.1 [OsclAny\\* OsclMemBasicAllocator::allocate \(const uint32 n\) \[inline, virtual\]](#)

This API throws an exception when malloc returns NULL. n must be greater than 0.

###### Returns:

pointer (or Leave with OsclErrNoMemory )

Implements [Oscl\\_DefAlloc](#).

##### 7.157.2.2 [void OsclMemBasicAllocator::deallocate \(OsclAny \\*p\) \[inline, virtual\]](#)

Implements [Oscl\\_DefAlloc](#).

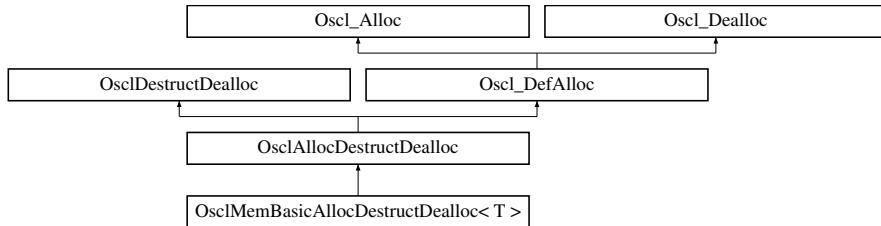
The documentation for this class was generated from the following file:

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

## 7.158 OsclMemBasicAllocDestructDealloc< T > Class Template Reference

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

Inheritance diagram for OsclMemBasicAllocDestructDealloc< T >::



### Public Methods

- [OsclAny \\* allocate \(const uint32 size\)](#)
- [void deallocate \(OsclAny \\*p\)](#)
- [void destruct\\_and\\_dealloc \(OsclAny \\*p\)](#)

#### 7.158.1 Detailed Description

**template<class T> class OsclMemBasicAllocDestructDealloc< T >**

An [OsclAllocDestructDealloc](#) class that uses [OsclMemBasicAllocator](#).

#### 7.158.2 Member Function Documentation

**7.158.2.1 template<class T> [OsclAny\\*](#) OsclMemBasicAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

**7.158.2.2 template<class T> void OsclMemBasicAllocDestructDealloc< T >::deallocate ([OsclAny](#) \* p) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

**7.158.2.3 template<class T> void OsclMemBasicAllocDestructDealloc< T >::destruct\_and\_dealloc ([OsclAny](#) \* p) [inline, virtual]**

Implements [OsclDestructDealloc](#).

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

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

## 7.159 OsclMemGlobalAuditObject Class Reference

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

### Public Types

- `typedef OsclMemAudit audit_type`

### Static Public Methods

- `OSCL_IMPORT_REF audit_type * getGlobalMemAuditObject ()`

### Friends

- class `OsclMem`

### 7.159.1 Member Typedef Documentation

#### 7.159.1.1 `typedef OsclMemAudit OsclMemGlobalAuditObject::audit_type`

### 7.159.2 Member Function Documentation

#### 7.159.2.1 `OSCL_IMPORT_REF audit_type* OsclMemGlobalAuditObject::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 OsclMem [friend]`

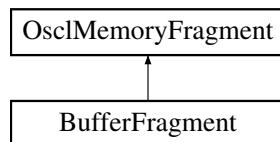
The documentation for this class was generated from the following file:

- `oscl_mem.h`

## 7.160 OsclMemoryFragment Struct Reference

```
#include <oscl_types.h>
```

Inheritance diagram for OsclMemoryFragment::



### Data Fields

- `uint32 len`
- `void * ptr`

#### 7.160.1 Field Documentation

##### 7.160.1.1 `uint32 OsclMemoryFragment::len`

##### 7.160.1.2 `void* OsclMemoryFragment::ptr`

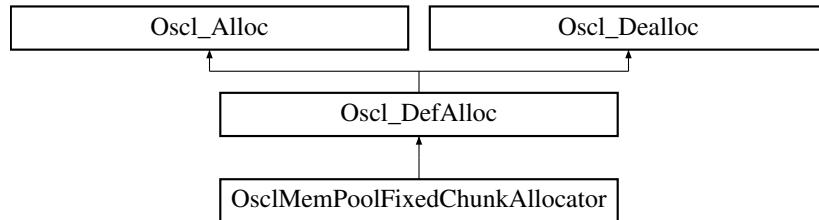
The documentation for this struct was generated from the following file:

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

## 7.161 OsclMemPoolFixedChunkAllocator Class Reference

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

Inheritance diagram for OsclMemPoolFixedChunkAllocator::



### Public Methods

- OSCL\_IMPORT\_REF OsclMemPoolFixedChunkAllocator (const uint32 numchunk=1, const uint32 chunksze=0, Oscl\_DefAlloc \*gen\_alloc=NULL, const uint32 chunkalignment=0)
- virtual OSCL\_IMPORT\_REF void enablenullpointerreturn ()
- virtual OSCL\_IMPORT\_REF ~OsclMemPoolFixedChunkAllocator ()
- virtual OSCL\_IMPORT\_REF OsclAny \* allocate (const uint32 n)
- virtual OSCL\_IMPORT\_REF void deallocate (OsclAny \*p)
- virtual OSCL\_IMPORT\_REF void notifyfreechunkavailable (OsclMemPoolFixedChunkAllocatorObserver &obs, OsclAny \*aContextData=NULL)
- virtual OSCL\_IMPORT\_REF void CancelFreeChunkAvailableCallback ()
- OSCL\_IMPORT\_REF void addRef ()
- OSCL\_IMPORT\_REF void removeRef ()

### Protected Methods

- virtual OSCL\_IMPORT\_REF void createmempool ()
- virtual OSCL\_IMPORT\_REF void destroymempool ()

### Protected Attributes

- uint32 iNumChunk
- uint32 iChunkSize
- uint32 iChunkSizeMemAligned
- uint32 iChunkAlignment
- Oscl\_DefAlloc \* iMemPoolAllocator
- OsclAny \* iMemPool
- OsclAny \* iMemPoolAligned
- Oscl\_Vector< OsclAny \*, OsclMemAllocator > iFreeMemChunkList
- bool iCheckNextAvailableFreeChunk
- OsclMemPoolFixedChunkAllocatorObserver \* iObserver
- OsclAny \* iNextAvailableContextData
- int32 iRefCount
- bool iEnableNullPtrReturn

### 7.161.1 Constructor & Destructor Documentation

**7.161.1.1 OSCL\_IMPORT\_REF OsclMemPoolFixedChunkAllocator::OsclMemPoolFixedChunkAllocator (const uint32 numchunk = 1, const uint32 chunkszie = 0, Oscl\_DefAlloc \* gen\_alloc = NULL, const uint32 chunkalignment = 0)**

This API throws an exception when the memory allocation for pool fails If numchunk and chunkszie parameters are not set, memory pool of 1 chunk will be created in the first call to allocate. The chunk size will be set to the n passed in for [allocate\(\)](#). If numchunk parameter is set to 0, the memory pool will use 1 for numchunk. 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

**7.161.1.2 virtual OSCL\_IMPORT\_REF OsclMemPoolFixedChunkAllocator::~OsclMemPoolFixedChunkAllocator () [virtual]**

The destructor for the memory pool

### 7.161.2 Member Function Documentation

**7.161.2.1 OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::addRef ()**

Increments the reference count for this memory pool allocator

**Returns:**

void

**7.161.2.2 virtual OSCL\_IMPORT\_REF OsclAny\* OsclMemPoolFixedChunkAllocator::allocate (const uint32 n) [virtual]**

This API throws an exception when n is greater than the fixed chunk size or there are no free chunk available in the pool, if "enablenullpointerreturn" has not been called. If the memory pool hasn't been created yet, the pool will be created with chunk size equal to n so n must be greater than 0. Exception will be thrown if memory allocation for the memory pool fails.

**Returns:**

pointer to available chunk from memory pool

Implements [Oscl\\_DefAlloc](#).

**7.161.2.3 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::CancelFreeChunkAvailableCallback () [virtual]**

This API will cancel any past callback requests..

**Returns:**

void

**7.161.2.4 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::createmempool()** [protected, virtual]

**7.161.2.5 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::deallocate(  
OsclAny \*p)** [virtual]

This API throws an exception when the pointer p passed in is not part of the memory pool. Exception will be thrown if the memory pool is not set up yet.

**Returns:**

void

Implements [Oscl\\_DefAlloc](#).

**7.161.2.6 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::destroymempool()** [protected, virtual]

**7.161.2.7 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::enablenullpointerreturn()** [virtual]

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL\\_LEAVE\(OsclErrNoResources\)](#)" allocate API will return NULL.

**Returns:**

void

**7.161.2.8 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::notifyfreechunkavailable(  
OsclMemPoolFixedChunkAllocatorObserver &obs, OsclAny \*aContextData = NULL)** [virtual]

This API will set the flag to send a callback via specified observer object when the next memory chunk is deallocated by [deallocate\(\)](#) call..

**Returns:**

void

**7.161.2.9 OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::removeRef()**

Decrements the reference count for this memory pool allocator When the reference count goes to 0, this instance of the memory pool object is deleted

**Returns:**

void

### 7.161.3 Field Documentation

- 7.161.3.1 **bool OsclMemPoolFixedChunkAllocator::iCheckNextAvailableFreeChunk** [protected]
- 7.161.3.2 **uint32 OsclMemPoolFixedChunkAllocator::iChunkAlignment** [protected]
- 7.161.3.3 **uint32 OsclMemPoolFixedChunkAllocator::iChunkSize** [protected]
- 7.161.3.4 **uint32 OsclMemPoolFixedChunkAllocator::iChunkSizeMemAligned** [protected]
- 7.161.3.5 **bool OsclMemPoolFixedChunkAllocator::iEnableNullPtrReturn** [protected]
- 7.161.3.6 **Oscl\_Vector<OsclAny\*, OsclMemAllocator> OsclMemPoolFixedChunkAllocator::iFreeMemChunkList** [protected]
- 7.161.3.7 **OsclAny\* OsclMemPoolFixedChunkAllocator::iMemPool** [protected]
- 7.161.3.8 **OsclAny\* OsclMemPoolFixedChunkAllocator::iMemPoolAligned** [protected]
- 7.161.3.9 **Oscl\_DefAlloc\* OsclMemPoolFixedChunkAllocator::iMemPoolAllocator** [protected]
- 7.161.3.10 **OsclAny\* OsclMemPoolFixedChunkAllocator::iNextAvailableContextData** [protected]
- 7.161.3.11 **uint32 OsclMemPoolFixedChunkAllocator::iNumChunk** [protected]
- 7.161.3.12 **OsclMemPoolFixedChunkAllocatorObserver\* OsclMemPoolFixedChunkAllocator::iObserver** [protected]
- 7.161.3.13 **int32 OsclMemPoolFixedChunkAllocator::iRefCount** [protected]

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

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

## 7.162 OsclMemPoolFixedChunkAllocatorObserver Class Reference

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

### Public Methods

- virtual void [freechunkavailable \(OsclAny \\*aContextData\)=0](#)
- virtual [~OsclMemPoolFixedChunkAllocatorObserver \(\)](#)

#### 7.162.1 Constructor & Destructor Documentation

**7.162.1.1** virtual [OsclMemPoolFixedChunkAllocatorObserver::~OsclMemPoolFixedChunkAllocatorObserver \(\) \[inline, virtual\]](#)

#### 7.162.2 Member Function Documentation

**7.162.2.1** virtual void [OsclMemPoolFixedChunkAllocatorObserver::freechunkavailable \(OsclAny \\* aContextData\) \[pure virtual\]](#)

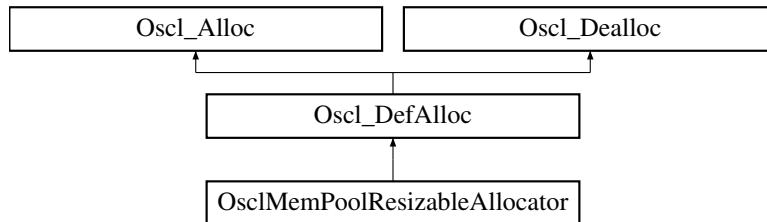
The documentation for this class was generated from the following file:

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

## 7.163 OsclMemPoolResizableAllocator Class Reference

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

Inheritance diagram for OsclMemPoolResizableAllocator::



### Public Methods

- OSCL\_IMPORT\_REF [OsclMemPoolResizableAllocator](#) (uint32 aMemPoolBufferSize, uint32 aMemPoolBufferNumLimit=0, uint32 aExpectedNumBlocksPerBuffer=0, [Oscl\\_DefAlloc](#) \*gen\_alloc=NULL)
- virtual OSCL\_IMPORT\_REF void [enablenullpointerreturn](#) ()
- virtual OSCL\_IMPORT\_REF [OsclAny](#) \* [allocate](#) (const uint32 aNumBytes)
- virtual OSCL\_IMPORT\_REF void [deallocate](#) ([OsclAny](#) \*aPtr)
- virtual OSCL\_IMPORT\_REF bool [trim](#) ([OsclAny](#) \*aPtr, uint32 aBytesToFree)
- OSCL\_IMPORT\_REF uint32 [getBufferSize](#) () const
- virtual OSCL\_IMPORT\_REF uint32 [getAllocatedSize](#) () const
- virtual OSCL\_IMPORT\_REF uint32 [getAvailableSize](#) () const
- virtual OSCL\_IMPORT\_REF uint32 [getLargestContiguousFreeBlockSize](#) () const
- virtual OSCL\_IMPORT\_REF bool [setMaxSzForNewMemPoolBuffer](#) (uint32 aMaxNewMemPoolBufferSz)
- virtual OSCL\_IMPORT\_REF void [notifyfreeblockavailable](#) ([OsclMemPoolResizableAllocatorObserver](#) &aObserver, uint32 aRequestedSize=0, [OsclAny](#) \*aContextData=NULL)
- virtual OSCL\_IMPORT\_REF void [CancelFreeChunkAvailableCallback](#) ()
- virtual OSCL\_IMPORT\_REF void [notifyfreememoryavailable](#) ([OsclMemPoolResizableAllocatorMemoryObserver](#) &aObserver, uint32 aRequestedSize=0, [OsclAny](#) \*aContextData=NULL)
- OSCL\_IMPORT\_REF void [CancelFreeMemoryAvailableCallback](#) ()
- OSCL\_IMPORT\_REF void [addRef](#) ()
- OSCL\_IMPORT\_REF void [removeRef](#) ()

### Protected Methods

- virtual ~[OsclMemPoolResizableAllocator](#) ()
- [MemPoolBufferInfo](#) \* [addnewmempoolbuffer](#) (uint32 aBufferSize)
- void [destroyallmempoolbuffers](#) ()
- [MemPoolBlockInfo](#) \* [findfreeblock](#) (uint32 aBlockSize)
- [OsclAny](#) \* [allocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr, uint32 aNumBytes)
- void [deallocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr)
- bool [validateblock](#) ([OsclAny](#) \*aBlockBufPtr)
- uint32 [getMemPoolBufferSize](#) ([MemPoolBufferInfo](#) \*aBufferInfo) const
- uint32 [getMemPoolBufferAllocatedSize](#) ([MemPoolBufferInfo](#) \*aBufferInfo) const
- uint32 [memoryPoolBufferMgmtOverhead](#) () const

## Protected Attributes

- uint32 `iMemPoolBufferSize`
- uint32 `iMemPoolBufferNumLimit`
- uint32 `iExpectedNumBlocksPerBuffer`
- uint32 `iMaxNewMemPoolBufferSz`
- `Oscl_DefAlloc * iMemPoolBufferAllocator`
- `Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator > iMemPoolBufferList`
- uint32 `iBufferInfoAlignedSize`
- uint32 `iBlockInfoAlignedSize`
- bool `iCheckNextAvailable`
- uint32 `iRequestedNextAvailableSize`
- `OsclAny * iNextAvailableContextData`
- `OsclMemPoolResizableAllocatorObserver * iObserver`
- bool `iCheckFreeMemoryAvailable`
- uint32 `iRequestedAvailableFreeMemSize`
- `OsclAny * iFreeMemContextData`
- `OsclMemPoolResizableAllocatorMemoryObserver * iFreeMemPoolObserver`
- int32 `iRefCount`
- bool `iEnableNullPtrReturn`

### 7.163.1 Constructor & Destructor Documentation

**7.163.1.1 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`

**7.163.1.2 virtual OsclMemPoolResizableAllocator::~OsclMemPoolResizableAllocator () [inline, protected, virtual]**

The destructor for the memory pool. Should not be called directly. Use `removeRef()` instead.

### 7.163.2 Member Function Documentation

**7.163.2.1 `MemPoolBufferInfo* OsclMemPoolResizableAllocator::addnewmempoolbuffer (uint32 aBufferSize)` [protected]**

**7.163.2.2 OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::addRef ()**

Increments the reference count for this memory pool allocator

**Returns:**

void

**7.163.2.3 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](#).

**7.163.2.4 OsclAny\* OsclMemPoolResizableAllocator::allocateblock (MemPoolBlockInfo &  
aBlockPtr, uint32 aNumBytes) [protected]****7.163.2.5 virtual OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::CancelFree-  
ChunkAvailableCallback () [virtual]**

This API will cancel any past callback requests..

**Returns:**

void

**7.163.2.6 OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::CancelFreeMemory-  
AvailableCallback ()****7.163.2.7 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](#).

**7.163.2.8 void OsclMemPoolResizableAllocator::deallocateblock (MemPoolBlockInfo &  
aBlockPtr) [protected]****7.163.2.9 void OsclMemPoolResizableAllocator::destroyallmempoolbuffers () [protected]****7.163.2.10 virtual OSCL\_IMPORT\_REF void OsclMemPoolResizable-  
Allocator::enablenullpointerreturn () [virtual]**

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL\\_LEAVE\(OsclErrNoResources\)](#)" allocate API will return NULL.

**Returns:**

void

**7.163.2.11** **MemPoolBlockInfo\*** OsclMemPoolResizableAllocator::findfreeblock (**uint32 aBlockSize**) [protected]

**7.163.2.12** **virtual OSCL\_IMPORT\_REF uint32 OsclMemPoolResizableAllocator::getAllocatedSize ()** [virtual]

Returns the number of bytes allocated from the buffer<including the overhead bytes that may be allocated by the allocator to keep track of the chunks allocated>

**7.163.2.13** **virtual OSCL\_IMPORT\_REF uint32 OsclMemPoolResizableAllocator::getAvailableSize ()** [virtual]

Returns the number of bytes available with the buffer

**7.163.2.14** **OSCL\_IMPORT\_REF uint32 OsclMemPoolResizableAllocator::getBufferSize ()**

Returns the size of the buffer <including the overhead bytes that may be allocated by the allocator>

**7.163.2.15** **virtual OSCL\_IMPORT\_REF uint32 OsclMemPoolResizableAllocator::getLargestContiguousFreeBlockSize ()** [virtual]

Returns the size of the largest available chunk in the memory.

**7.163.2.16** **uint32 OsclMemPoolResizableAllocator::getMemPoolBufferAllocatedSize (MemPoolBufferInfo \* aBufferInfo) const** [protected]

**7.163.2.17** **uint32 OsclMemPoolResizableAllocator::getMemPoolBufferSize (MemPoolBufferInfo \* aBufferInfo) const** [protected]

**7.163.2.18** **uint32 OsclMemPoolResizableAllocator::memoryPoolBufferMgmtOverhead ()** [protected]

**7.163.2.19** **virtual OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::notifyfreeblockavailable (OsclMemPoolResizableAllocatorObserver & aObserver, uint32 aRequestedSize = 0, OsclAny \* aContextData = NULL)** [virtual]

This API will set the flag to send a callback via specified observer object when the next memory block is deallocated by [deallocate\(\)](#) call. If the optional requested size parameter is set, the callback is sent when a free memory space of requested size becomes available. The optional context data is returned with the callback and can be used by the user to differentiate between different instances of memory pool objects. This memory pool only allows one notify to be queued. Another call to this function will just overwrite the previous call.

**Returns:**

void

**7.163.2.20** `virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::notifyfreememoryavailable (OsclMemPoolResizableAllocatorMemoryObserver & aObserver, uint32 aRequestedSize = 0, OsclAny * aContextData = NULL)`  
[virtual]

**7.163.2.21** `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`

**7.163.2.22** `virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::setMaxSzForNewMemPoolBuffer (uint32 aMaxNewMemPoolBufferSz)`  
[virtual]

**7.163.2.23** `virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::trim (OsclAny * aPtr, uint32 aBytesToFree) [virtual]`

Returns a tail segment of a previously allocated memory block back to the memory pool. The passed-in pointer to the memory buffer is still valid after the call completes but the buffer size is smaller by by specified amount that was freed. This function allows the user to allocate a larger size block initially when the amount needed is unknown and then return the unused portion of the block when the amount becomes known. This API throws an exception if the pointer passed in is not part of the memory pool or the size to return is bigger than the size of the passed-in block. Exception will be thrown if the memory pool is not set up yet.

**Returns:**

`bool` True if trim operation successful. False if the block wasn't trimmed

7.163.2.24 **bool OsclMemPoolResizableAllocator::validateblock ([OsclAny](#) \* *aBlockBufPtr*)** [protected]

### 7.163.3 Field Documentation

7.163.3.1 **uint32 OsclMemPoolResizableAllocator::iBlockInfoAlignedSize** [protected]

7.163.3.2 **uint32 OsclMemPoolResizableAllocator::iBufferInfoAlignedSize** [protected]

7.163.3.3 **bool OsclMemPoolResizableAllocator::iCheckFreeMemoryAvailable** [protected]

7.163.3.4 **bool OsclMemPoolResizableAllocator::iCheckNextAvailable** [protected]

7.163.3.5 **bool OsclMemPoolResizableAllocator::iEnableNullPtrReturn** [protected]

7.163.3.6 **uint32 OsclMemPoolResizableAllocator::iExpectedNumBlocksPerBuffer** [protected]

7.163.3.7 **[OsclAny](#)\* OsclMemPoolResizableAllocator::iFreeMemContextData** [protected]

7.163.3.8 **[OsclMemPoolResizableAllocatorMemoryObserver](#)\* OsclMemPoolResizableAllocator::iFreeMemPoolObserver** [protected]

7.163.3.9 **uint32 OsclMemPoolResizableAllocator::iMaxNewMemPoolBufferSz** [protected]

7.163.3.10 **[Oscl\\_DefAlloc](#)\* OsclMemPoolResizableAllocator::iMemPoolBufferAllocator** [protected]

7.163.3.11 **[Oscl\\_Vector](#)<[MemPoolBufferInfo](#)\*, [OsclMemAllocator](#)> OsclMemPoolResizableAllocator::iMemPoolBufferList** [protected]

7.163.3.12 **uint32 OsclMemPoolResizableAllocator::iMemPoolBufferNumLimit** [protected]

7.163.3.13 **uint32 OsclMemPoolResizableAllocator::iMemPoolBufferSize** [protected]

7.163.3.14 **[OsclAny](#)\* OsclMemPoolResizableAllocator::iNextAvailableContextData** [protected]

7.163.3.15 **[OsclMemPoolResizableAllocatorObserver](#)\* OsclMemPoolResizableAllocator::iObserver** [protected]

7.163.3.16 **int32 OsclMemPoolResizableAllocator::iRefCount** [protected]

7.163.3.17 **uint32 OsclMemPoolResizableAllocator::iRequestedAvailableFreeMemSize** [protected]

7.163.3.18 **uint32 OsclMemPoolResizableAllocator::iRequestedNextAvailableSize** [protected]

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

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

## 7.164 OsclMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference

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

### Data Fields

- uint32 iBlockPreFence
- MemPoolBlockInfo \* iNextFreeBlock
- MemPoolBlockInfo \* iPrevFreeBlock
- uint32 iBlockSize
- uint8 \* iBlockBuffer
- MemPoolBufferInfo \* iParentBuffer
- uint32 iBlockPostFence

#### 7.164.1 Field Documentation

**7.164.1.1 uint8\* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockBuffer**

**7.164.1.2 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPostFence**

**7.164.1.3 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPreFence**

**7.164.1.4 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockSize**

**7.164.1.5 MemPoolBlockInfo\* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iNextFree-Block**

**7.164.1.6 MemPoolBufferInfo\* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iParent-Buffer**

**7.164.1.7 MemPoolBlockInfo\* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iPrevFree-Block**

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

- oscl\_mem\_mempool.h

## 7.165 OsclMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference

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

### Data Fields

- uint32 iBufferPreFence
- [OsclAny](#) \* iStartAddr
- [OsclAny](#) \* iEndAddr
- uint32 iBufferSize
- uint32 iNumOutstanding
- [MemPoolBlockInfo](#) \* iNextFreeBlock
- uint32 iAllocatedSz
- uint32 iBufferPostFence

#### 7.165.1 Field Documentation

**7.165.1.1 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iAllocatedSz**

**7.165.1.2 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPostFence**

**7.165.1.3 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPreFence**

**7.165.1.4 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferSize**

**7.165.1.5 [OsclAny](#)\* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iEndAddr**

**7.165.1.6 [MemPoolBlockInfo](#)\* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iNextFree-Block**

**7.165.1.7 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iNumOutstanding**

**7.165.1.8 [OsclAny](#)\* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iStartAddr**

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

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

## 7.166 OsclMemPoolResizableAllocatorMemoryObserver Class Reference

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

### Public Methods

- virtual void [freememoryavailable \(OsclAny \\*aContextData\)=0](#)
- virtual [~OsclMemPoolResizableAllocatorMemoryObserver \(\)](#)

#### 7.166.1 Constructor & Destructor Documentation

**7.166.1.1 virtual OsclMemPoolResizableAllocatorMemoryObserver::~OsclMemPoolResizableAllocatorMemoryObserver () [inline, virtual]**

#### 7.166.2 Member Function Documentation

**7.166.2.1 virtual void OsclMemPoolResizableAllocatorMemoryObserver::freememoryavailable (OsclAny \* aContextData) [pure virtual]**

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

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

## 7.167 OsclMemPoolResizableAllocatorObserver Class Reference

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

### Public Methods

- virtual void [freeblockavailable \(OsclAny \\*aContextData\)=0](#)
- virtual [~OsclMemPoolResizableAllocatorObserver \(\)](#)

#### 7.167.1 Constructor & Destructor Documentation

**7.167.1.1** [virtual OsclMemPoolResizableAllocatorObserver::~OsclMemPoolResizableAllocatorObserver \(\) \[inline, virtual\]](#)

#### 7.167.2 Member Function Documentation

**7.167.2.1** [virtual void OsclMemPoolResizableAllocatorObserver::freeblockavailable \(OsclAny \\*\*aContextData\*\) \[pure virtual\]](#)

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

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

## 7.168 OsclMemStatsNode Class Reference

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

### Public Methods

- [OsclMemStatsNode \(\)](#)
- [void reset \(\)](#)
- [~OsclMemStatsNode \(\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size, OsclMemStatsNode \\*ptr\)](#)
- [void operator delete \(void \\*ptr\) throw \(\)](#)

### Data Fields

- [MM\\_Stats\\_t \\* pMMStats](#)
- [MM\\_FailInsertParam \\* pMMFIParam](#)
- [char \\* tag](#)

#### 7.168.1 Constructor & Destructor Documentation

**7.168.1.1 OsclMemStatsNode::OsclMemStatsNode () [inline]**

**7.168.1.2 OsclMemStatsNode::~OsclMemStatsNode () [inline]**

#### 7.168.2 Member Function Documentation

**7.168.2.1 void OsclMemStatsNode::operator delete (void \*ptr) throw () [inline]**

**7.168.2.2 void\* OsclMemStatsNode::operator new (oscl\_memsize\_t size, OsclMemStatsNode \*ptr) [inline]**

**7.168.2.3 void\* OsclMemStatsNode::operator new (oscl\_memsize\_t size) [inline]**

**7.168.2.4 void OsclMemStatsNode::reset () [inline]**

#### 7.168.3 Field Documentation

**7.168.3.1 MM\_FailInsertParam\* OsclMemStatsNode::pMMFIParam**

**7.168.3.2 MM\_Stats\_t\* OsclMemStatsNode::pMMStats**

**7.168.3.3 char\* OsclMemStatsNode::tag**

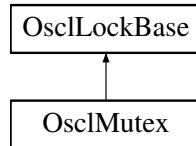
The documentation for this class was generated from the following file:

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

## 7.169 OsclMutex Class Reference

```
#include <oscl_mutex.h>
```

Inheritance diagram for OsclMutex::



### Public Methods

- OSCL\_IMPORT\_REF OsclMutex ()
- virtual OSCL\_IMPORT\_REF ~OsclMutex ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Create (void)
- OSCL\_IMPORT\_REF void Lock ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError TryLock ()
- OSCL\_IMPORT\_REF void Unlock ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Close (void)

#### 7.169.1 Detailed Description

Class OsclMutex

#### 7.169.2 Constructor & Destructor Documentation

##### 7.169.2.1 OSCL\_IMPORT\_REF OsclMutex::OsclMutex ()

Class constructor

##### 7.169.2.2 virtual OSCL\_IMPORT\_REF OsclMutex::~OsclMutex () [virtual]

Class destructor

#### 7.169.3 Member Function Documentation

##### 7.169.3.1 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclMutex::Close (void)

Closes the Mutex

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

**7.169.3.2 OSCL\_IMPORT\_REF [OsclProcStatus::eOsclProcError](#) OsclMutex::Create (void)**

Creates the Mutex

**Parameters:**

*No* input arguments

**Returns:**

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

**7.169.3.3 OSCL\_IMPORT\_REF void OsclMutex::Lock () [virtual]**

Locks the Mutex

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns nothing

Implements [OsclLockBase](#).

**7.169.3.4 OSCL\_IMPORT\_REF [OsclProcStatus::eOsclProcError](#) OsclMutex::TryLock ()**

Try to lock the mutex,if the Mutex is already locked calling thread immediately returns with out blocking

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns SUCCESS\_ERROR if the mutex was acquired, MUTEX\_LOCKED\_ERROR if the mutex cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.169.3.5 OSCL\_IMPORT\_REF void OsclMutex::Unlock () [virtual]**

Releases the Mutex

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns nothing

Implements [OsclLockBase](#).

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

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

## 7.170 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](#)

### 7.170.1 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.

### 7.170.2 Constructor & Destructor Documentation

**7.170.2.1 template<int \_\_len> OsclNameString< \_\_len >::OsclNameString () [inline]**

**7.170.2.2 template<int \_\_len> OsclNameString< \_\_len >::OsclNameString (const char a[ ]) [inline]**

**7.170.2.3 template<int \_\_len> OsclNameString< \_\_len >::OsclNameString (uint8 \* a) [inline]**

### 7.170.3 Member Function Documentation

**7.170.3.1 template<int \_\_len> int32 OsclNameString< \_\_len >::MaxLen () const [inline]**

**7.170.3.2 template<int \_\_len> void OsclNameString< \_\_len >::Set (const char a[ ]) [inline]**

**7.170.3.3 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.

**7.170.3.4 template<int \_\_len> uint8\* OsclNameString< \_\_len >::Str () const [inline]**

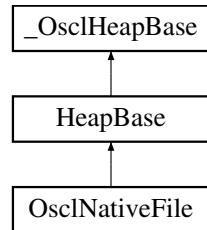
The documentation for this class was generated from the following file:

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

## 7.171 OsclNativeFile Class Reference

```
#include <oscl_file_native.h>
```

Inheritance diagram for OsclNativeFile::



### Public Methods

- [OsclNativeFile \(\)](#)
- [~OsclNativeFile \(\)](#)
- [int32 Open \(const OsclFileHandle &, uint32 mode, const OsclNativeFileParams &params, Oscl\\_FileServer &fileserv\)](#)
- [int32 Open \(const oscl\\_wchar \\*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\\_FileServer &fileserv\)](#)
- [int32 Open \(const char \\*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\\_FileServer &fileserv\)](#)
- [uint32 Read \(OsclAny \\*buffer, uint32 size, uint32 numelements\)](#)
- [uint32 Write \(const OsclAny \\*buffer, uint32 size, uint32 numelements\)](#)
- [int32 Seek \(TOsclFileOffset offset, Oscl\\_File::seek\\_type origin\)](#)
- [TOsclFileOffset Tell \(\)](#)
- [int32 Flush \(\)](#)
- [int32 EndOfFile \(\)](#)
- [TOsclFileOffset Size \(\)](#)
- [int32 Close \(\)](#)
- [int32 SetSize \(uint32 size\)](#)
- [uint32 Mode \(\)](#)
- [int32 GetError \(\)](#)
- [int32 ReadAsync \(OsclAny \\*buffer, uint32 size, uint32 numelements, OsclAOStatus &status\)](#)
- [uint32 GetReadAsyncNumElements \(\)](#)
- [bool HasAsyncRead \(\)](#)
- [void ReadAsyncCancel \(\)](#)

### 7.171.1 Constructor & Destructor Documentation

**7.171.1.1 OsclNativeFile::OsclNativeFile ()**

**7.171.1.2 OsclNativeFile::~OsclNativeFile ()**

### 7.171.2 Member Function Documentation

**7.171.2.1 int32 OsclNativeFile::Close ()**

**7.171.2.2 int32 OsclNativeFile::EndOfFile ()**

**7.171.2.3 int32 OsclNativeFile::Flush ()**

**7.171.2.4 int32 OsclNativeFile::GetError ()**

**7.171.2.5 uint32 OsclNativeFile::GetReadAsyncNumElements ()**

Get the number of elements read in the last call to ReadAsync. @returns: number of elements read.

**7.171.2.6 bool OsclNativeFile::HasAsyncRead ()**

@returns: true if async read is supported natively.

**7.171.2.7 uint32 OsclNativeFile::Mode () [inline]**

**7.171.2.8 int32 OsclNativeFile::Open (const char \*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\_FileServer &fileserv)**

**7.171.2.9 int32 OsclNativeFile::Open (const oscl\_wchar \*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\_FileServer &fileserv)**

**7.171.2.10 int32 OsclNativeFile::Open (const OsclFileHandle &, uint32 mode, const OsclNativeFileParams &params, Oscl\_FileServer &fileserv)**

**7.171.2.11 uint32 OsclNativeFile::Read (OsclAny \*buffer, uint32 size, uint32 numelements)**

**7.171.2.12 int32 OsclNativeFile::ReadAsync (OsclAny \*buffer, uint32 size, uint32 numelements, OsclAOStatus &status)**

Asynchronous read.

#### Parameters:

**buffer:** data buffer, must be at least size\*numelements bytes

**size:** size of elements

**numelements:** number of elements to read

**status:** Request status for asynchronous completion @returns: 0 for success.

**7.171.2.13 void OsclNativeFile::ReadAsyncCancel ()**

Cancel any pending async read.

**7.171.2.14 int32 OsclNativeFile::Seek ([TOsclFileOffset offset](#), [Oscl\\_File::seek\\_type origin](#))****7.171.2.15 int32 OsclNativeFile::SetSize (uint32 *size*)****7.171.2.16 [TOsclFileOffset](#) OsclNativeFile::Size ()****7.171.2.17 [TOsclFileOffset](#) OsclNativeFile::Tell ()****7.171.2.18 uint32 OsclNativeFile::Write (const [OsclAny](#) \* *buffer*, uint32 *size*, uint32 *numelements*)**

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

- [oscl\\_file\\_native.h](#)

## 7.172 OsclNativeFileParams Class Reference

```
#include <oscl_file_types.h>
```

### Public Methods

- [OsclNativeFileParams \(uint32 mode=0, uint32 bufsize=0, uint32 asyncsize=0\)](#)

### Data Fields

- uint32 [iNativeAccessMode](#)
- uint32 [iNativeBufferSize](#)
- uint32 [iAsyncReadBufferSize](#)

#### 7.172.1 Constructor & Destructor Documentation

7.172.1.1 **OsclNativeFileParams::OsclNativeFileParams (uint32 mode = 0, uint32 bufsize = 0, uint32 asyncsize = 0) [inline]**

#### 7.172.2 Field Documentation

7.172.2.1 **uint32 OsclNativeFileParams::iAsyncReadBufferSize**

7.172.2.2 **uint32 OsclNativeFileParams::iNativeAccessMode**

7.172.2.3 **uint32 OsclNativeFileParams::iNativeBufferSize**

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

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

## 7.173 OsclNetworkAddress Class Reference

```
#include <oscl_socket_types.h>
```

### Public Methods

- [OsclNetworkAddress \(\)](#)
- [OsclNetworkAddress \(const char \\*addr, int p\)](#)
- [bool operator== \(const OsclNetworkAddress &rhs\) const](#)

### Data Fields

- [OsclNameString< PVNETWORKADDRESS\\_LEN > ipAddr](#)
- [int port](#)

#### 7.173.1 Constructor & Destructor Documentation

7.173.1.1 [OsclNetworkAddress::OsclNetworkAddress \(\) \[inline\]](#)

7.173.1.2 [OsclNetworkAddress::OsclNetworkAddress \(const char \\*addr, int p\) \[inline\]](#)

#### 7.173.2 Member Function Documentation

7.173.2.1 [bool OsclNetworkAddress::operator== \(const OsclNetworkAddress & rhs\) const \[inline\]](#)

#### 7.173.3 Field Documentation

7.173.3.1 [OsclNameString<PVNETWORKADDRESS\\_LEN> OsclNetworkAddress::ipAddr](#)

7.173.3.2 [int OsclNetworkAddress::port](#)

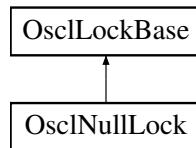
The documentation for this class was generated from the following file:

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

## 7.174 OsclNullLock Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclNullLock::



### Public Methods

- virtual void [Lock \(\)](#)
- virtual void [Unlock \(\)](#)
- virtual [~OsclNullLock \(\)](#)

#### 7.174.1 Constructor & Destructor Documentation

**7.174.1.1 virtual OsclNullLock::~OsclNullLock () [inline, virtual]**

#### 7.174.2 Member Function Documentation

**7.174.2.1 virtual void OsclNullLock::Lock () [inline, virtual]**

Implements [OsclLockBase](#).

**7.174.2.2 virtual void OsclNullLock::Unlock () [inline, virtual]**

Implements [OsclLockBase](#).

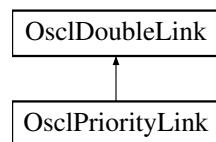
The documentation for this class was generated from the following file:

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

## 7.175 OsclPriorityLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclPriorityLink::



### Data Fields

- int32 [iPriority](#)

#### 7.175.1 Field Documentation

##### 7.175.1.1 int32 OsclPriorityLink::iPriority

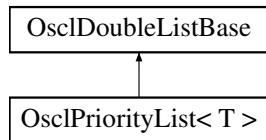
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.176 OsclPriorityList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclPriorityList< T >::



### Public Methods

- OSCL\_INLINE OsclPriorityList ()
- OSCL\_INLINE OsclPriorityList (int32 anOffset)
- OSCL\_INLINE void Insert (T &aRef)
- OSCL\_INLINE bool IsHead (const T \*aPtr) const
- OSCL\_INLINE bool IsTail (const T \*aPtr) const
- OSCL\_INLINE T \* Head () const
- OSCL\_INLINE T \* Tail () const

```
template<class T> class OsclPriorityList< T >
```

#### 7.176.1 Constructor & Destructor Documentation

**7.176.1.1 template<class T> OSCL\_INLINE OsclPriorityList< T >::OsclPriorityList ()**

**7.176.1.2 template<class T> OSCL\_INLINE OsclPriorityList< T >::OsclPriorityList (int32 *anOffset*)**

#### 7.176.2 Member Function Documentation

**7.176.2.1 template<class T> OSCL\_INLINE T\* OsclPriorityList< T >::Head ()**

**7.176.2.2 template<class T> OSCL\_INLINE void OsclPriorityList< T >::Insert (T &*aRef*)**

**7.176.2.3 template<class T> OSCL\_INLINE bool OsclPriorityList< T >::IsHead (const T \**aPtr*) const**

**7.176.2.4 template<class T> OSCL\_INLINE bool OsclPriorityList< T >::IsTail (const T \**aPtr*) const**

**7.176.2.5 template<class T> OSCL\_INLINE T\* OsclPriorityList< T >::Tail ()**

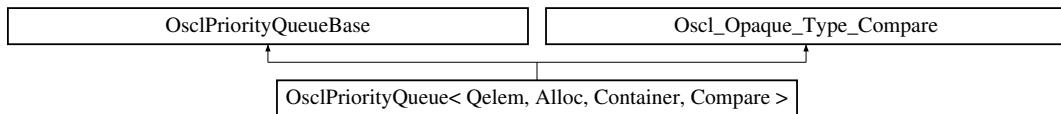
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.177 OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference

```
#include <oscl_priqueue.h>
```

Inheritance diagram for OsclPriorityQueue< Qelem, Alloc, Container, Compare >::



### Public Types

- `typedef Container::value_type value_type`
- `typedef Container container_type`
- `typedef Container::iterator iterator`
- `typedef Container::const_reference const_reference`

### Public Methods

- `bool empty () const`
- `uint32 size () const`
- `void reserve (uint32 n)`
- `const_reference top () const`
- `const Container & vec ()`
- `void push (const value_type &input)`
- `void pop ()`
- `int remove (const value_type &input)`
- `OsclPriorityQueue ()`
- `virtual ~OsclPriorityQueue ()`

### Protected Methods

- `void push_heap (iterator first, iterator last)`
- `void pop_heap (iterator first, iterator last)`
- `iterator find_heap (const value_type &input, iterator first, iterator last)`
- `int validate ()`
- `void swap (OsclAny *dest, const OsclAny *src)`
- `int compare_LT (OsclAny *a, OsclAny *b) const`
- `int compare_EQ (const OsclAny *a, const OsclAny *b) const`

### Protected Attributes

- `Container c`
- `Compare comp`

## Friends

- class [oscl\\_priqueue\\_test](#)

```
template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> class OsclPriorityQueue< Qelem, Alloc, Container, Compare >
```

### 7.177.1 Member Typedef Documentation

- 7.177.1.1 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container::const\_reference OsclPriorityQueue< Qelem, Alloc, Container, Compare >::const\_reference**
- 7.177.1.2 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::container\_type**
- 7.177.1.3 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container::iterator OsclPriorityQueue< Qelem, Alloc, Container, Compare >::iterator**
- 7.177.1.4 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container::value\_type OsclPriorityQueue< Qelem, Alloc, Container, Compare >::value\_type**

### 7.177.2 Constructor & Destructor Documentation

- 7.177.2.1 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **OsclPriorityQueue< Qelem, Alloc, Container, Compare >::OsclPriorityQueue () [inline]**
- 7.177.2.2 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **virtual OsclPriorityQueue< Qelem, Alloc, Container, Compare >::~OsclPriorityQueue () [inline, virtual]**

### 7.177.3 Member Function Documentation

- 7.177.3.1 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::compare\_EQ (const OsclAny \* a, const OsclAny \* b) const [inline, protected, virtual]**

Return a==b.

Implements [Oscl\\_Opaque\\_Type\\_Compare](#).

- 7.177.3.2 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::compare\_LT (OsclAny \* a, OsclAny \* b) const [inline, protected, virtual]**

Return a<b.

Implements [Oscl\\_Opaque\\_Type\\_Compare](#).

- 7.177.3.3 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> bool OsclPriorityQueue< Qelem, Alloc, Container, Compare >::empty () const [inline]`
- 7.177.3.4 `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]`
- 7.177.3.5 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::pop () [inline]`
- 7.177.3.6 `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]`
- 7.177.3.7 `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]`
- 7.177.3.8 `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]`
- 7.177.3.9 `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]`
- 7.177.3.10 `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]`
- 7.177.3.11 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> uint32 OsclPriorityQueue< Qelem, Alloc, Container, Compare >::size () const [inline]`
- 7.177.3.12 `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](#).

- 7.177.3.13 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]
- 7.177.3.14 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::validate () [inline, protected]
- 7.177.3.15 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **const Container&** OsclPriorityQueue< Qelem, Alloc, Container, Compare >::vec () [inline]

#### 7.177.4 Friends And Related Function Documentation

- 7.177.4.1 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> friend class oscl\_priqueue\_test [friend]

#### 7.177.5 Field Documentation

- 7.177.5.1 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::c [protected]
- 7.177.5.2 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Compare OsclPriorityQueue< Qelem, Alloc, Container, Compare >::comp [protected]

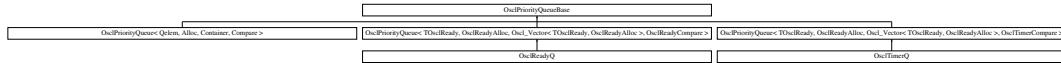
The documentation for this class was generated from the following file:

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

## 7.178 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)

### **7.178.1 Detailed Description**

`OsclPriorityQueueBase` is a non-templatized 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`.

## 7.178.2 Constructor & Destructor Documentation

**7.178.2.1 virtual OsclPriorityQueueBase::~OsclPriorityQueueBase ()** [inline, protected, virtual]

### 7.178.3 Member Function Documentation

**7.178.3.1 void OsclPriorityQueueBase::construct (Oscl\_Opaque\_Type\_Compare \* *ot*, Oscl\_Vector\_Base \* *vec*) [inline, protected]**

**7.178.3.2 OSCL\_IMPORT\_REF OsclAny\* OsclPriorityQueueBase::find\_heap (const OsclAny \* input, OsclAny \* first, OsclAny \* last) [protected]**

**7.178.3.3 OSCL\_IMPORT\_REF void OsclPriorityQueueBase::pop\_heap (OsclAny \*first, OsclAny \*last) [protected]**

**7.178.3.4 OSCL\_IMPORT\_REF void OsclPriorityQueueBase::push\_heap (OsclAny \*first, OsclAny \*last) [protected]**

**7.178.3.5 OSCL\_IMPORT\_REF int OsclPriorityQueueBase::remove (const OsclAny \* input)**  
[protected]

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

- oscl priqueue.h

## 7.179 OsclProcStatus Class Reference

```
#include <oscl_procstatus.h>
```

### Public Types

- enum `eOsclProcError` { `SUCCESS_ERROR` = 0, `OTHER_ERROR`, `TOO_MANY_THREADS_ERROR`, `BAD_THREADID_ADDR_ERROR`, `MAX_THRDS_REACHED_ERROR`, `INVALID_THREAD_ID_ERROR`, `NOT_ENOUGH_MEMORY_ERROR`, `OUTOFMEMORY_ERROR`, `NOT_ENOUGH_RESOURCES_ERROR`, `THREAD_1_INACTIVE_ERROR`, `ALREADY_SUSPENDED_ERROR`, `NOT_SUSPENDED_ERROR`, `INVALID_THREAD_ERROR`, `INVALID_PARAM_ERROR`, `NO_PERMISSION_ERROR`, `INVALID_PRIORITY_ERROR`, `PSHARED_NOT_ZERO_ERROR`, `EXCEED_MAX_COUNT_VARIABLE_ERROR`, `THREAD_BLOCK_ERROR`, `EXCEED_MAX_SEM_COUNT_ERROR`, `INVALID_HANDLE_ERROR`, `INVALID_OPERATION_ERROR`, `INVALID_FUNCTION_ERROR`, `INVALID_ACCESS_ERROR`, `INVALID_ARGUMENT_ERROR`, `SYSTEM_RESOURCES_UNAVAILABLE_ERROR`, `INVALID_POINTER_ERROR`, `RELOCK_MUTEX_ERROR`, `THREAD_NOT_OWN_MUTEX_ERROR`, `MUTEX_LOCKED_ERROR`, `WAIT_ABANDONED_ERROR`, `WAIT_TIMEOUT_ERROR`, `SEM_NOT_SIGNALED_ERROR`, `PSHARED_ATTRIBUTE_SETTING_ERROR`, `NOT_IMPLEMENTED` }

### 7.179.1 Detailed Description

Class OsclProcStatus

### 7.179.2 Member Enumeration Documentation

#### 7.179.2.1 enum OsclProcStatus::eOsclProcError

List of enums which contain error codes

Enumeration values:

`SUCCESS_ERROR`  
`OTHER_ERROR`  
`TOO_MANY_THREADS_ERROR`  
`BAD_THREADID_ADDR_ERROR`  
`MAX_THRDS_REACHED_ERROR`  
`INVALID_THREAD_ID_ERROR`  
`NOT_ENOUGH_MEMORY_ERROR`  
`OUTOFMEMORY_ERROR`  
`NOT_ENOUGH_RESOURCES_ERROR`  
`THREAD_1_INACTIVE_ERROR`  
`ALREADY_SUSPENDED_ERROR`  
`NOT_SUSPENDED_ERROR`  
`INVALID_THREAD_ERROR`  
`INVALID_PARAM_ERROR`  
`NO_PERMISSION_ERROR`

**INVALID\_PRIORITY\_ERROR**  
**PSHARED\_NOT\_ZERO\_ERROR**  
**EXCEED\_MAX\_COUNT\_VARIABLE\_ERROR**  
**THREAD\_BLOCK\_ERROR**  
**EXCEED\_MAX\_SEM\_COUNT\_ERROR**  
**INVALID\_HANDLE\_ERROR**  
**INVALID\_OPERATION\_ERROR**  
**INVALID\_FUNCTION\_ERROR**  
**INVALID\_ACCESS\_ERROR**  
**INVALID\_ARGUMENT\_ERROR**  
**SYSTEM\_RESOURCES\_UNAVAILABLE\_ERROR**  
**INVALID\_POINTER\_ERROR**  
**RELOCK\_MUTEX\_ERROR**  
**THREAD\_NOT\_OWN\_MUTEX\_ERROR**  
**MUTEX\_LOCKED\_ERROR**  
**WAIT\_ABANDONED\_ERROR**  
**WAIT\_TIMEOUT\_ERROR**  
**SEM\_NOT\_SIGNALED\_ERROR**  
**PSHARED\_ATTRIBUTE\_SETTING\_ERROR**  
**NOT\_IMPLEMENTED**

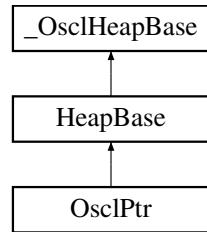
The documentation for this class was generated from the following file:

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

## 7.180 OsclPtr Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclPtr::



### Public Methods

- [OsclPtr \(uint8 \\*ptr, int32 &len, int32 max\)](#)
- [OsclPtr \(const OsclPtr &d\)](#)
- [uint8 \\* Ptr \(\)](#)
- [void SetLength \(int32 l\)](#)
- [int32 Length \(\)](#)
- [void Zero \(\)](#)
- [void Set \(OsclPtr &v\)](#)
- [void Set \(uint8 \\*ptr, int32 len, int32 max\)](#)
- [void Append \(OsclPtrC &v\)](#)

#### 7.180.1 Constructor & Destructor Documentation

**7.180.1.1 OsclPtr::OsclPtr (uint8 \*ptr, int32 &len, int32 max) [inline]**

**7.180.1.2 OsclPtr::OsclPtr (const OsclPtr &d) [inline]**

#### 7.180.2 Member Function Documentation

**7.180.2.1 void OsclPtr::Append (OsclPtrC &v) [inline]**

**7.180.2.2 int32 OsclPtr::Length () [inline]**

**7.180.2.3 uint8\* OsclPtr::Ptr () [inline]**

**7.180.2.4 void OsclPtr::Set (uint8 \*ptr, int32 len, int32 max) [inline]**

**7.180.2.5 void OsclPtr::Set (OsclPtr &v) [inline]**

**7.180.2.6 void OsclPtr::SetLength (int32 l) [inline]**

**7.180.2.7 void OsclPtr::Zero () [inline]**

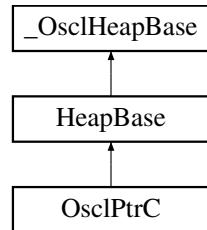
The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 7.181 OsclPtrC Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclPtrC::



### Public Methods

- [OsclPtrC](#) (const uint8 \*ptr, int32 len, int32 max)
- [OsclPtrC](#) (const OsclPtrC &d)
- const uint8 \* [Ptr](#) ()
- void [SetLength](#) (int32 l)
- int32 [Length](#) ()
- void [Zero](#) ()
- void [Set](#) (OsclPtrC \*v)
- void [Set](#) (uint8 \*ptr, int32 len, int32 max)
- OsclPtrC [Right](#) (int32 size)
- OsclPtrC [Left](#) (int32 size)

### 7.181.1 Constructor & Destructor Documentation

**7.181.1.1** `OsclPtrC::OsclPtrC (const uint8 *ptr, int32 len, int32 max)` [inline]

**7.181.1.2** `OsclPtrC::OsclPtrC (const OsclPtrC & d)` [inline]

### 7.181.2 Member Function Documentation

**7.181.2.1** `OsclPtrC OsclPtrC::Left (int32 size)` [inline]

**7.181.2.2** `int32 OsclPtrC::Length ()` [inline]

**7.181.2.3** `const uint8* OsclPtrC::Ptr ()` [inline]

**7.181.2.4** `OsclPtrC OsclPtrC::Right (int32 size)` [inline]

**7.181.2.5** `void OsclPtrC::Set (uint8 *ptr, int32 len, int32 max)` [inline]

**7.181.2.6** `void OsclPtrC::Set (OsclPtrC *v)` [inline]

**7.181.2.7** `void OsclPtrC::SetLength (int32 l)` [inline]

**7.181.2.8** `void OsclPtrC::Zero ()` [inline]

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

- [oscl\\_file\\_async\\_read.h](#)

## 7.182 OsclRand Class Reference

```
#include <oscl_rand.h>
```

### Public Methods

- OSCL\_COND\_IMPORT\_REF void [Seed](#) (int32 seed)
- OSCL\_COND\_IMPORT\_REF int32 [Rand](#) ()

#### 7.182.1 Member Function Documentation

##### 7.182.1.1 OSCL\_COND\_IMPORT\_REF int32 OsclRand::Rand ()

##### 7.182.1.2 OSCL\_COND\_IMPORT\_REF void OsclRand::Seed (int32 *seed*)

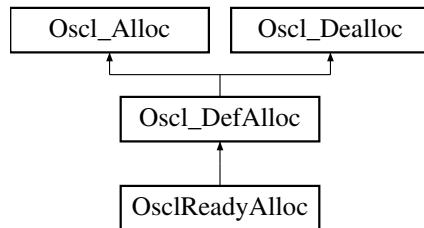
The documentation for this class was generated from the following file:

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

## 7.183 OsclReadyAlloc Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyAlloc::



### Public Methods

- [OsclAny \\* allocate \(const uint32 size\)](#)
- [OsclAny \\* allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\)](#)
- void [deallocate \(OsclAny \\*p\)](#)

#### 7.183.1 Member Function Documentation

##### 7.183.1.1 [OsclAny\\* OsclReadyAlloc::allocate \(const uint32 size\) \[virtual\]](#)

Implements [Oscl\\_DefAlloc](#).

##### 7.183.1.2 [OsclAny\\* OsclReadyAlloc::allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\) \[virtual\]](#)

Reimplemented from [Oscl\\_DefAlloc](#).

##### 7.183.1.3 [void OsclReadyAlloc::deallocate \(OsclAny \\*p\) \[virtual\]](#)

Implements [Oscl\\_DefAlloc](#).

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

- [oscl\\_scheduler\\_readyq.h](#)

## 7.184 OsclReadyCompare Class Reference

```
#include <oscl_scheduler_readyq.h>
```

### Static Public Methods

- int [compare \(TOsclReady &a, TOsclReady &b\)](#)

#### 7.184.1 Member Function Documentation

##### 7.184.1.1 int OsclReadyCompare::compare (TOsclReady &*a*, TOsclReady &*b*) [static]

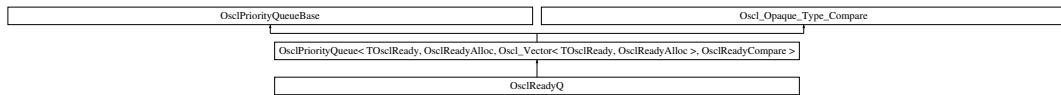
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 7.185 OsclReadyQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyQ::



### Public Methods

- void [Construct](#) (int)
- void [ThreadLogon](#) ()
- void [ThreadLogoff](#) ()
- void [Remove](#) (TOscIReady)
- bool [IsIn](#) (TOscIReady)
- uint32 [Depth](#) ()
- [TOscIReady PopTop](#) ()
- [TOscIReady Top](#) ()
- [TOscIReady WaitAndPopTop](#) ()
- [TOscIReady WaitAndPopTop](#) (uint32)
- int32 [PendComplete](#) (PVActiveBase \*pvbase, int32 aReason)
- int32 [WaitForRequestComplete](#) (PVActiveBase \*)
- void [RegisterForCallback](#) (OsclSchedulerObserver \*aCallback, OsclAny \*aCallbackContext)
- void [TimerCallback](#) (uint32 aDelayMicrosec)
- [OsclSchedulerObserver \\* Callback](#) ()

## 7.185.1 Member Function Documentation

7.185.1.1 **OsclSchedulerObserver\*** OsclReadyQ::Callback () [inline]

7.185.1.2 void OsclReadyQ::Construct (int)

7.185.1.3 uint32 OsclReadyQ::Depth () [inline]

7.185.1.4 bool OsclReadyQ::IsIn (**TOsclReady**)

7.185.1.5 int32 OsclReadyQ::PendComplete (**PVActiveBase** \**pvbase*, int32 *aReason*)

7.185.1.6 **TOsclReady** OsclReadyQ::PopTop ()

7.185.1.7 void OsclReadyQ::RegisterForCallback (**OsclSchedulerObserver** \**aCallback*, **OsclAny** \**aCallbackContext*)

7.185.1.8 void OsclReadyQ::Remove (**TOsclReady**)

7.185.1.9 void OsclReadyQ::ThreadLogoff ()

7.185.1.10 void OsclReadyQ::ThreadLogon ()

7.185.1.11 void OsclReadyQ::TimerCallback (uint32 *aDelayMicrosec*)

7.185.1.12 **TOsclReady** OsclReadyQ::Top ()

7.185.1.13 **TOsclReady** OsclReadyQ::WaitAndPopTop (uint32)

7.185.1.14 **TOsclReady** OsclReadyQ::WaitAndPopTop ()

7.185.1.15 int32 OsclReadyQ::WaitForRequestComplete (**PVActiveBase** \*)

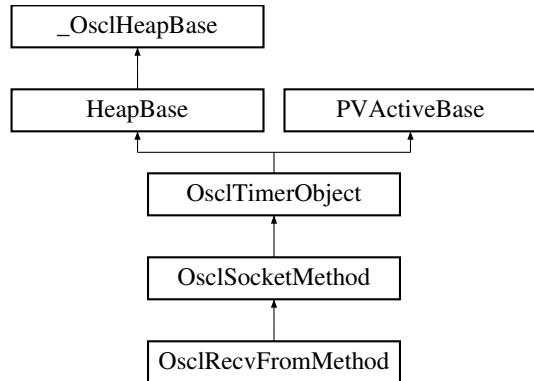
The documentation for this class was generated from the following file:

- `oscl_scheduler_readyq.h`

## 7.186 OsclRecvFromMethod Class Reference

```
#include <oscl_socket_recv_from.h>
```

Inheritance diagram for OsclRecvFromMethod:::



### Public Methods

- [`~OsclRecvFromMethod \(\)`](#)
- [`TPVSocketEvent RecvFrom \(uint8 \*&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeout, uint32 aMultiMax, Oscl\_Vector< uint32, OsclMemAllocator > \*aPacketLen, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \*aPacketSource\)`](#)
- [`uint8 \* GetRecvData \(int32 \*aLength\)`](#)
- [`OsclRecvFromRequest \* RecvFromRequest \(\)`](#)

### Static Public Methods

- [`OsclRecvFromMethod \* NewL \(OsclIPSocketI &c\)`](#)

#### 7.186.1 Constructor & Destructor Documentation

##### 7.186.1.1 OsclRecvFromMethod::~OsclRecvFromMethod ()

#### 7.186.2 Member Function Documentation

##### 7.186.2.1 uint8\* OsclRecvFromMethod::GetRecvData (int32 \* aLength)

##### 7.186.2.2 OsclRecvFromMethod\* OsclRecvFromMethod::NewL (OsclIPSocketI & c) [static]

##### 7.186.2.3 TPVSocketEvent OsclRecvFromMethod::RecvFrom (uint8 \*& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeout, uint32 aMultiMax, Oscl\_Vector< uint32, OsclMemAllocator > \* aPacketLen, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \* aPacketSource)

##### 7.186.2.4 OsclRecvFromRequest\* OsclRecvFromMethod::RecvFromRequest () [inline]

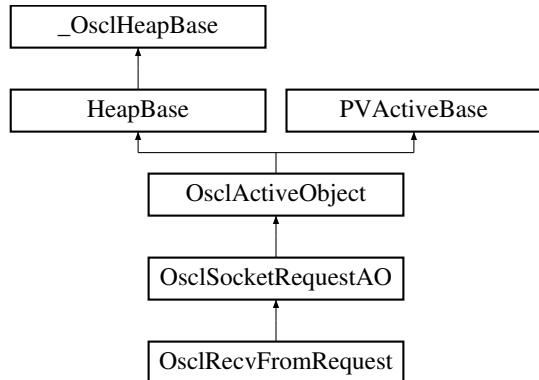
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_recv\\_from.h](#)

## 7.187 OsclRecvFromRequest Class Reference

```
#include <oscl_socket_recv_from.h>
```

Inheritance diagram for OsclRecvFromRequest::



### Public Methods

- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvFromRequest (OsclSocketMethod &c)`
- `void RecvFrom (uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource)`
- `void Success ()`

#### 7.187.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.187.2 Constructor & Destructor Documentation

**7.187.2.1 OsclRecvFromRequest::OsclRecvFromRequest (`OsclSocketMethod &c`) [inline]**

#### 7.187.3 Member Function Documentation

**7.187.3.1 `uint8* OsclRecvFromRequest::GetRecvData (int32 * aLength)`**

**7.187.3.2 `void OsclRecvFromRequest::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource)`**

**7.187.3.3 `void OsclRecvFromRequest::Success () [virtual]`**

Reimplemented from `OsclSocketRequestAO`.

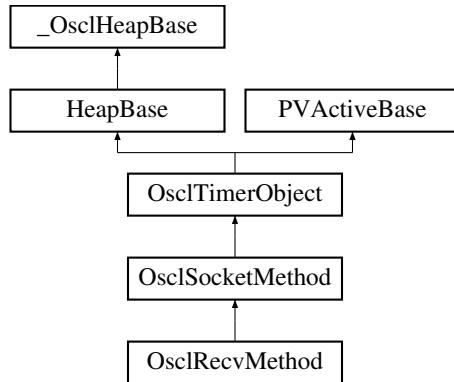
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_recv\\_from.h](#)

## 7.188 OsclRecvMethod Class Reference

```
#include <oscl_socket_recv.h>
```

Inheritance diagram for OsclRecvMethod::



### Public Methods

- [~OsclRecvMethod \(\)](#)
- [TPVSocketEvent Recv \(uint8 \\*&aPtr, uint32 aMaxLen, int32 aTimeout\)](#)
- [uint8 \\* GetRecvData \(int32 \\*aLength\)](#)
- [OsclRecvRequest \\* RecvRequest \(\)](#)

### Static Public Methods

- [OsclRecvMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 7.188.1 Constructor & Destructor Documentation

##### 7.188.1.1 OsclRecvMethod::~OsclRecvMethod ()

#### 7.188.2 Member Function Documentation

##### 7.188.2.1 uint8\* OsclRecvMethod::GetRecvData (int32 \* aLength)

##### 7.188.2.2 OsclRecvMethod\* OsclRecvMethod::NewL (OsclIPSocketI & c) [static]

##### 7.188.2.3 TPVSocketEvent OsclRecvMethod::Recv (uint8 \*& aPtr, uint32 aMaxLen, int32 aTimeout)

##### 7.188.2.4 OsclRecvRequest\* OsclRecvMethod::RecvRequest () [inline]

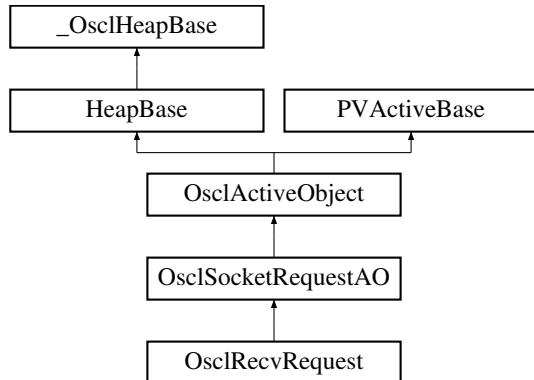
The documentation for this class was generated from the following file:

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

## 7.189 OsclRecvRequest Class Reference

```
#include <oscl_socket_recv.h>
```

Inheritance diagram for OsclRecvRequest::



### Public Methods

- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvRequest (OsclSocketMethod &c)`
- `void Recv (uint8 *&aPtr, uint32 aMaxLen)`
- `void Success ()`

#### 7.189.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.189.2 Constructor & Destructor Documentation

**7.189.2.1 OsclRecvRequest::OsclRecvRequest (`OsclSocketMethod &c`) [inline]**

#### 7.189.3 Member Function Documentation

**7.189.3.1 `uint8* OsclRecvRequest::GetRecvData (int32 * aLength)`**

**7.189.3.2 `void OsclRecvRequest::Recv (uint8 *& aPtr, uint32 aMaxLen)`**

**7.189.3.3 `void OsclRecvRequest::Success () [virtual]`**

Reimplemented from `OsclSocketRequestAO`.

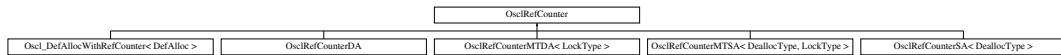
The documentation for this class was generated from the following file:

- `oscl_socket_recv.h`

## 7.190 OsclRefCounter Class Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounter:::



### Public Methods

- virtual void `addRef ()=0`
- virtual void `removeRef ()=0`
- virtual uint32 `getCount ()=0`
- virtual `~OsclRefCounter ()`

#### 7.190.1 Detailed Description

Interface class for OsclRefCounter implementations

#### 7.190.2 Constructor & Destructor Documentation

**7.190.2.1 virtual OsclRefCounter::~OsclRefCounter () [inline, virtual]**

#### 7.190.3 Member Function Documentation

**7.190.3.1 virtual void OsclRefCounter::addRef () [pure virtual]**

Add to the reference count

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_RefAllocWithRefCounter< DefAlloc >`.

**7.190.3.2 virtual uint32 OsclRefCounter::getCount () [pure virtual]**

Gets the current number of references

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_RefAllocWithRefCounter< DefAlloc >`.

**7.190.3.3 virtual void OsclRefCounter::removeRef () [pure virtual]**

Delete from reference count

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_RefAllocWithRefCounter< DefAlloc >`.

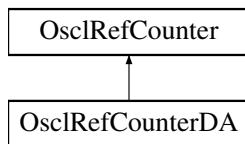
The documentation for this class was generated from the following file:

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

## 7.191 OsclRefCounterDA Class Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterDA::



### Public Methods

- [OsclRefCounterDA \(OsclAny \\*p, OsclDestructDealloc \\*dealloc\)](#)
- virtual [~OsclRefCounterDA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 7.191.1 Detailed Description

Implementation of an [OsclRefCounter](#) that uses a dynamically created deallocator.

#### 7.191.2 Constructor & Destructor Documentation

##### 7.191.2.1 OsclRefCounterDA::OsclRefCounterDA ([OsclAny \\*p](#), [OsclDestructDealloc \\*dealloc](#)) [inline]

Constructor Takes a pointer to the buffer to track, and a pointer to the deallocator object which will be used to delete the buffer.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to `delete()`.

##### Parameters:

*p* pointer to the buffer to track

*dealloc* pointer to the deallocator to use when deleting the buffer

##### 7.191.2.2 virtual OsclRefCounterDA::~OsclRefCounterDA () [inline, virtual]

Destructor empty

### 7.191.3 Member Function Documentation

#### 7.191.3.1 void OsclRefCounterDA::addRef () [inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

#### 7.191.3.2 uint32 OsclRefCounterDA::getCount () [inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

#### 7.191.3.3 void OsclRefCounterDA::removeRef () [inline, virtual]

Remove from the reference count

Implements [OsclRefCounter](#).

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

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

## 7.192 OsclRefCounterMemFrag Class Reference

```
#include <oscl_refcounter_memfrag.h>
```

### Public Methods

- [OsclRefCounterMemFrag \(OsclMemoryFragment &m, OsclRefCounter \\*r, uint32 in\\_capacity\)](#)
- [OsclRefCounterMemFrag \(const OsclRefCounterMemFrag &x\)](#)
- [OsclRefCounterMemFrag \(\)](#)
- [OsclRefCounterMemFrag & operator= \(const OsclRefCounterMemFrag &x\)](#)
- [~OsclRefCounterMemFrag \(\)](#)
- [OsclRefCounter \\* getRefCounter \(\)](#)
- [OsclMemoryFragment & getMemFrag \(\)](#)
- [OsclAny \\* getMemFragPtr \(\)](#)
- [uint32 getMemFragSize \(\)](#)
- [uint32 getCapacity \(\)](#)
- [uint32 getCount \(\)](#)

### 7.192.1 Detailed Description

Class to contain a memory fragment with it's associated reference counter.

### 7.192.2 Constructor & Destructor Documentation

#### 7.192.2.1 OsclRefCounterMemFrag::OsclRefCounterMemFrag ([OsclMemoryFragment & m](#), [OsclRefCounter \\* r](#), [uint32 in\\_capacity](#)) [inline]

Constructor. A valid memory fragment and reference counter are required as input. The memory fragment structure will be copied locally.

##### Parameters:

- m* reference to memory fragment
- r* pointer to the reference counter associated with the memory fragment.

#### 7.192.2.2 OsclRefCounterMemFrag::OsclRefCounterMemFrag ([const OsclRefCounterMemFrag & x](#)) [inline]

Copy constructor.

#### 7.192.2.3 OsclRefCounterMemFrag::OsclRefCounterMemFrag () [inline]

Default constructor.

#### 7.192.2.4 OsclRefCounterMemFrag::~OsclRefCounterMemFrag () [inline]

Destructor. Removes this object's reference from the reference counter. The reference counter will not be deleted. The reference counter is designed to self-delete when it's reference count reaches 0.

### 7.192.3 Member Function Documentation

#### 7.192.3.1 **uint32 OsclRefCounterMemFrag::getCapacity () [inline]**

Returns the capacity of the memory fragment

**Returns:**

#### 7.192.3.2 **uint32 OsclRefCounterMemFrag::getCount () [inline]**

Returns the reference counter's current count.

#### 7.192.3.3 **OsclMemoryFragment& OsclRefCounterMemFrag::getMemFrag () [inline]**

Returns a reference to the contained memory fragment structure.

#### 7.192.3.4 **OsclAny\* OsclRefCounterMemFrag::getMemFragPtr () [inline]**

Returns a pointer to the memory fragment data.

#### 7.192.3.5 **uint32 OsclRefCounterMemFrag::getMemFragSize () [inline]**

Returns the size of the memory fragment data which equals its filled size.

**Returns:**

#### 7.192.3.6 **OsclRefCounter\* OsclRefCounterMemFrag::getRefCounter () [inline]**

Returns a pointer to the contained reference counter object

#### 7.192.3.7 **OsclRefCounterMemFrag& OsclRefCounterMemFrag::operator= (const OsclRefCounterMemFrag & x) [inline]**

Assignment Operator

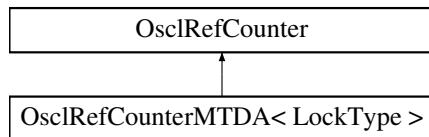
The documentation for this class was generated from the following file:

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

## 7.193 OsclRefCounterMTDA< LockType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterMTDA< LockType >::



### Public Methods

- [OsclRefCounterMTDA \(OsclAny \\*p, OsclDestructDealloc \\*dealloc\)](#)
- virtual [~OsclRefCounterMTDA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 7.193.1 Detailed Description

**template<class LockType> class OsclRefCounterMTDA< LockType >**

Implementation of [OsclRefCounterDA](#) for multi-threaded use. A templated lock class must be specified.

#### 7.193.2 Constructor & Destructor Documentation

**7.193.2.1 template<class LockType> OsclRefCounterMTDA< LockType >::OsclRefCounterMTDA (OsclAny \*p, OsclDestructDealloc \*dealloc) [inline]**

Constructor Takes a pointer to the buffer to track, and a pointer to the deallocator object which will be used to delete the buffer.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to [delete\(\)](#).

##### Parameters:

*p* pointer to the buffer to track

*dealloc* pointer to the deallocator to use when deleting the buffer

**7.193.2.2 template<class LockType> virtual OsclRefCounterMTDA< LockType >::~OsclRefCounterMTDA () [inline, virtual]**

Destructor empty

### 7.193.3 Member Function Documentation

**7.193.3.1 template<class LockType> void OsclRefCounterMTDA< LockType >::addRef ()  
[inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**7.193.3.2 template<class LockType> uint32 OsclRefCounterMTDA< LockType >::getCount ()  
[inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**7.193.3.3 template<class LockType> void OsclRefCounterMTDA< LockType >::removeRef ()  
[inline, virtual]**

Remove from the reference count

Implements [OsclRefCounter](#).

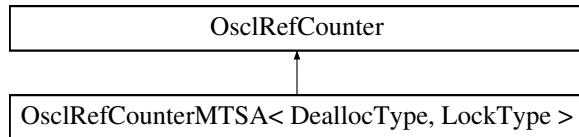
The documentation for this class was generated from the following file:

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

## 7.194 OsclRefCounterMTSA< DeallocType, LockType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterMTSA< DeallocType, LockType >::



### Public Methods

- [OsclRefCounterMTSA \(OsclAny \\*p\)](#)
- virtual [~OsclRefCounterMTSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 7.194.1 Detailed Description

```
template<class DeallocType, class LockType> class OsclRefCounterMTSA< DeallocType, LockType >
```

Implementation of [OsclRefCounterSA](#) for multi-threaded use. A templated lock class must be specified.

#### 7.194.2 Constructor & Destructor Documentation

**7.194.2.1 template<class DeallocType, class LockType> OsclRefCounterMTSA< DeallocType, LockType >::OsclRefCounterMTSA (OsclAny \* p) [inline]**

Constructor Takes a pointer to the buffer to track.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to `delete()`.

**Parameters:**

*p* pointer to the buffer to track

**7.194.2.2 template<class DeallocType, class LockType> virtual OsclRefCounterMTSA< DeallocType, LockType >::~OsclRefCounterMTSA () [inline, virtual]**

Destructor empty

### 7.194.3 Member Function Documentation

**7.194.3.1 template<class DeallocType, class LockType> void OsclRefCounterMTSA< DeallocType, LockType >::addRef () [inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**7.194.3.2 template<class DeallocType, class LockType> uint32 OsclRefCounterMTSA< DeallocType, LockType >::getCount () [inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**7.194.3.3 template<class DeallocType, class LockType> void OsclRefCounterMTSA< DeallocType, LockType >::removeRef () [inline, virtual]**

Remove from the reference count

Implements [OsclRefCounter](#).

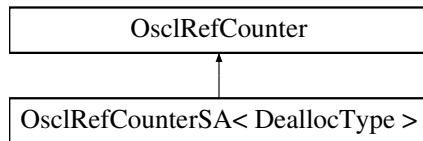
The documentation for this class was generated from the following file:

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

## 7.195 OsclRefCounterSA< DeallocType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterSA< DeallocType >::



### Public Methods

- [OsclRefCounterSA \(OsclAny \\*p\)](#)
- virtual [~OsclRefCounterSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 7.195.1 Detailed Description

**template<class DeallocType> class OsclRefCounterSA< DeallocType >**

Implementation of an [OsclRefCounter](#) that uses a statically created deallocator.

#### 7.195.2 Constructor & Destructor Documentation

##### 7.195.2.1 template<class DeallocType> OsclRefCounterSA< DeallocType >::OsclRefCounterSA (OsclAny \*p) [inline]

Constructor Takes a pointer to the buffer to track.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to [delete\(\)](#).

**Parameters:**

*p* pointer to the buffer to track

##### 7.195.2.2 template<class DeallocType> virtual OsclRefCounterSA< DeallocType >::~OsclRefCounterSA () [inline, virtual]

Destructor empty

### 7.195.3 Member Function Documentation

**7.195.3.1 template<class DeallocType> void OsclRefCounterSA< DeallocType >::addRef () [inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**7.195.3.2 template<class DeallocType> uint32 OsclRefCounterSA< DeallocType >::getCount () [inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**7.195.3.3 template<class DeallocType> void OsclRefCounterSA< DeallocType >::removeRef () [inline, virtual]**

Remove from the reference count

Implements [OsclRefCounter](#).

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

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

## 7.196 OsclRegistryAccessClient Class Reference

```
#include <oscl_registry_access_client.h>
```

### Public Methods

- OSCL\_IMPORT\_REF OsclRegistryAccessClient ()
- OSCL\_IMPORT\_REF ~OsclRegistryAccessClient ()
- OSCL\_IMPORT\_REF int32 Connect ()
- OSCL\_IMPORT\_REF OsclComponentFactory GetFactory (OSCL\_String &aComponent)
- OSCL\_IMPORT\_REF void GetFactories (OSCL\_String &aRegistry, Oscl\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &aVec)
- OSCL\_IMPORT\_REF void Close ()

#### 7.196.1 Constructor & Destructor Documentation

**7.196.1.1 OSCL\_IMPORT\_REF OsclRegistryAccessClient::OsclRegistryAccessClient ()**

**7.196.1.2 OSCL\_IMPORT\_REF OsclRegistryAccessClient::~OsclRegistryAccessClient ()**

#### 7.196.2 Member Function Documentation

**7.196.2.1 OSCL\_IMPORT\_REF void OsclRegistryAccessClient::Close ()**

Close and cleanup session.

**7.196.2.2 OSCL\_IMPORT\_REF int32 OsclRegistryAccessClient::Connect ()**

Create a session.

**Returns:**

OsclErrNone on success.

**7.196.2.3 OSCL\_IMPORT\_REF void OsclRegistryAccessClient::GetFactories (OSCL\_String & aRegistry, Oscl\_Vector< OsclRegistryAccessElement, OsclMemAllocator > & aVec)**

Get all factories for a given registry type.

**Parameters:**

*aRegistry*: registry MIME type

*aVec*: output component factory + mimestring vector.

**7.196.2.4 OSCL\_IMPORT\_REF OsclComponentFactory OsclRegistryAccessClient::GetFactory (OSCL\_String & aComponent)**

Lookup a factory by registry and component mime type.

**Parameters:**

*aComponent*: registry+component MIME type

**Returns:**

Factory. Factory will be NULL if not found.

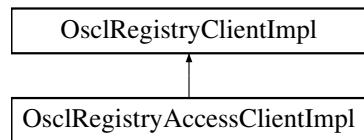
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_access\\_client.h](#)

## 7.197 OsclRegistryAccessClientImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryAccessClientImpl::



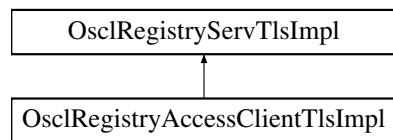
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 7.198 OsclRegistryAccessClientTlsImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryAccessClientTlsImpl::



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

- [oscl\\_registry\\_client\\_impl.h](#)

## 7.199 OsclRegistryAccessElement Class Reference

```
#include <oscl_registry_types.h>
```

### Data Fields

- [OsclComponentFactory](#) iFactory
- [OSCL\\_HeapString< OsclMemAllocator >](#) iMimeType

#### 7.199.1 Detailed Description

A class used to access the registry data

#### 7.199.2 Field Documentation

##### 7.199.2.1 [OsclComponentFactory](#) OsclRegistryAccessElement::iFactory

##### 7.199.2.2 [OSCL\\_HeapString<OsclMemAllocator>](#) OsclRegistryAccessElement::iMimeType

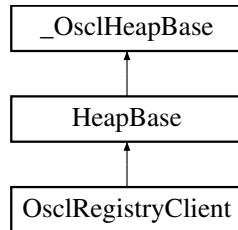
The documentation for this class was generated from the following file:

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

## 7.200 OsclRegistryClient Class Reference

```
#include <oscl_registry_client.h>
```

Inheritance diagram for OsclRegistryClient::



### Public Methods

- OSCL\_IMPORT\_REF [OsclRegistryClient \(\)](#)
- OSCL\_IMPORT\_REF [~OsclRegistryClient \(\)](#)
- OSCL\_IMPORT\_REF int32 [Connect \(bool aPerThread=false\)](#)
- OSCL\_IMPORT\_REF int32 [Register \(OSCL\\_String &aComponentID, OsclComponentFactory aFactory\)](#)
- OSCL\_IMPORT\_REF int32 [UnRegister \(OSCL\\_String &aComponentID\)](#)
- OSCL\_IMPORT\_REF void [Close \(\)](#)

#### 7.200.1 Constructor & Destructor Documentation

**7.200.1.1 OSCL\_IMPORT\_REF OsclRegistryClient::OsclRegistryClient ()**

**7.200.1.2 OSCL\_IMPORT\_REF OsclRegistryClient::~OsclRegistryClient ()**

#### 7.200.2 Member Function Documentation

**7.200.2.1 OSCL\_IMPORT\_REF void OsclRegistryClient::Close ()**

Close and cleanup. All components registered in this session are automatically unregistered.

**7.200.2.2 OSCL\_IMPORT\_REF int32 OsclRegistryClient::Connect (bool *aPerThread* = false)**

Create a session.

**Parameters:**

*aPerThread*: Select per-thread registry instead of global registry.

**Returns:**

OsclErrNone on success.

**7.200.2.3 OSCL\_IMPORT\_REF int32 OsclRegistryClient::Register ([OSCL\\_String &](#)  
*aComponentID*, [OsclComponentFactory](#) *aFactory*)**

Register a component factory by registry ID and component ID.

**Parameters:**

*aComponentID*: registry + component mime-string.

*aFactory*: factory function pointer.

*aParam*: component Create param.

**Returns:**

OsclErrNone on success.

**7.200.2.4 OSCL\_IMPORT\_REF int32 OsclRegistryClient::UnRegister ([OSCL\\_String &](#)  
*aComponentID*)**

Unregister a previously registered component.

**Returns:**

OsclErrNone on success.

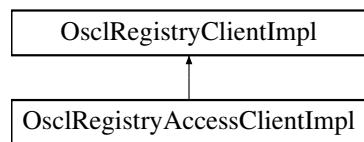
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client.h](#)

## 7.201 OsclRegistryClientImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryClientImpl:



### Protected Methods

- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL\\_String &, OsclComponentFactory\)](#)
- int32 [UnRegister \(OSCL\\_String &\)](#)
- [OsclComponentFactory GetFactory \(OSCL\\_String &\)](#)
- void [GetFactories \(OSCL\\_String &, Oscl\\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &\)](#)

### Friends

- class [OsclRegistryClient](#)
- class [OsclRegistryAccessClient](#)

## 7.201.1 Member Function Documentation

7.201.1.1 **void OsclRegistryClientImpl::Close (void)** [inline, protected]

7.201.1.2 **int32 OsclRegistryClientImpl::Connect ()** [inline, protected]

7.201.1.3 **void OsclRegistryClientImpl::GetFactories (OSCL\_String &, Oscl\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &)** [inline, protected]

7.201.1.4 **OsclComponentFactory OsclRegistryClientImpl::GetFactory (OSCL\_String &)**  
[inline, protected]

7.201.1.5 **int32 OsclRegistryClientImpl::Register (OSCL\_String &, OsclComponentFactory)**  
[inline, protected]

7.201.1.6 **int32 OsclRegistryClientImpl::UnRegister (OSCL\_String &)** [inline,  
protected]

## 7.201.2 Friends And Related Function Documentation

7.201.2.1 **friend class OsclRegistryAccessClient** [friend]

7.201.2.2 **friend class OsclRegistryClient** [friend]

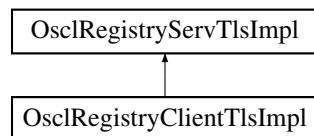
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 7.202 OsclRegistryClientTlsImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryClientTlsImpl::



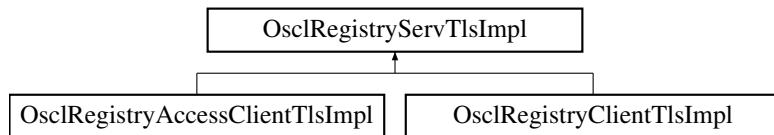
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 7.203 OsclRegistryServTlsImpl Class Reference

```
#include <oscl_registry_serv_impl_tls.h>
```

Inheritance diagram for OsclRegistryServTlsImpl::



### Protected Methods

- [OsclRegistryServTlsImpl \(\)](#)
- virtual [~OsclRegistryServTlsImpl \(\)](#)
- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL\\_String &aComponentID, OsclComponentFactory aFactory\)](#)
- int32 [UnRegister \(OSCL\\_String &aComponentID\)](#)
- [OsclComponentFactory GetFactory \(OSCL\\_String &aComponent\)](#)
- void [GetFactories \(OSCL\\_String &aRegistry, Oscl\\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &aVec\)](#)

### Friends

- class [OsclRegistryClient](#)
- class [OsclRegistryAccessClient](#)

### 7.203.1 Constructor & Destructor Documentation

7.203.1.1 `OsclRegistryServTlsImpl::OsclRegistryServTlsImpl ()` [protected]

7.203.1.2 `virtual OsclRegistryServTlsImpl::~OsclRegistryServTlsImpl ()` [protected, virtual]

### 7.203.2 Member Function Documentation

7.203.2.1 `void OsclRegistryServTlsImpl::Close ()` [protected]

7.203.2.2 `int32 OsclRegistryServTlsImpl::Connect ()` [protected]

7.203.2.3 `void OsclRegistryServTlsImpl::GetFactories (OSCL_String & aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > & aVec)` [protected]

7.203.2.4 `OsclComponentFactory OsclRegistryServTlsImpl::GetFactory (OSCL_String & aComponent)` [protected]

7.203.2.5 `int32 OsclRegistryServTlsImpl::Register (OSCL_String & aComponentID, OsclComponentFactory aFactory)` [protected]

7.203.2.6 `int32 OsclRegistryServTlsImpl::UnRegister (OSCL_String & aComponentID)` [protected]

### 7.203.3 Friends And Related Function Documentation

7.203.3.1 `friend class OsclRegistryAccessClient` [friend]

7.203.3.2 `friend class OsclRegistryClient` [friend]

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

- [oscl\\_registry\\_serv\\_impl\\_tls.h](#)

## 7.204 OsclScheduler Class Reference

```
#include <oscl_scheduler.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [Init](#) (const char \*name, [Oscl\\_DefAlloc](#) \*alloc=NULL, int nreserve=20)
- OSCL\_IMPORT\_REF void [Cleanup](#) ()

#### 7.204.1 Detailed Description

Per-thread scheduler initialization and cleanup.

#### 7.204.2 Member Function Documentation

##### 7.204.2.1 OSCL\_IMPORT\_REF void OsclScheduler::Cleanup () [static]

This routine uninstalls and destroys Oscl scheduler for the calling thread.

##### 7.204.2.2 OSCL\_IMPORT\_REF void OsclScheduler::Init (const char \* *name*, [Oscl\\_DefAlloc](#) \* *alloc* = NULL, int *nreserve* = 20) [static]

This routine creates and installs a scheduler in the calling thread.

#### Parameters:

- name*: (input param) scheduler name.  
*alloc*: (input param) optional allocator to use for the internal implementation.  
*nreserve*: (input param) optional value for ready queue reserve size.

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

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

## 7.205 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](#) ()

### 7.205.1 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.

### 7.205.2 Constructor & Destructor Documentation

**7.205.2.1** virtual [OsclSchedulerObserver::~OsclSchedulerObserver](#) () [inline, virtual]

### 7.205.3 Member Function Documentation

**7.205.3.1** 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.

**7.205.3.2** 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](#)

## 7.206 OsclScopedLock< LockClass > Class Template Reference

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

```
#include <oscl_lock_base.h>
```

### Public Methods

- [OsclScopedLock \(LockClass &inLock\)](#)  
*Default constructor Initializes the pointer and takes ownership.*
- [~OsclScopedLock \(\)](#)  
*Destructor.*

### 7.206.1 Detailed Description

**template<class LockClass> class OsclScopedLock< LockClass >**

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

The purpose of this class is to provide a way to prevent accidental resource leaks in a class or a method, due to "not remembering to unlock" variables which might lead to deadlock conditions.

### 7.206.2 Constructor & Destructor Documentation

#### 7.206.2.1 **template<class LockClass> OsclScopedLock< LockClass >::OsclScopedLock (LockClass & inLock) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

#### 7.206.2.2 **template<class LockClass> OsclScopedLock< LockClass >::~OsclScopedLock () [inline]**

Destructor.

The pointer is deleted in case this class still has ownership

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

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

## 7.207 OsclSelect Class Reference

```
#include <oscl_init.h>
```

### Public Methods

- [OsclSelect \(\)](#)
- [OsclSelect \(Oscl\\_DefAlloc \\*erralloc, Oscl\\_DefAlloc \\*schedalloc, const char \\*name, int32 reserve=10, bool heapcheck=false, FILE \\*output=NULL\)](#)

### Data Fields

- bool [iOsclBase](#)
- bool [iOsclMemory](#)
- bool [iOsclErrorTrap](#)
- bool [iOsclLogger](#)
- bool [iOsclScheduler](#)
- [Oscl\\_DefAlloc \\* iErrAlloc](#)
- [Oscl\\_DefAlloc \\* iSchedulerAlloc](#)
- const char \* [iSchedulerName](#)
- int32 [iSchedulerReserve](#)
- bool [iHeapCheck](#)
- FILE \* [iOutputFile](#)

### 7.207.1 Detailed Description

Oscl Module selection and Init/Cleanup options.

## 7.207.2 Constructor & Destructor Documentation

**7.207.2.1 OsclSelect::OsclSelect () [inline]**

**7.207.2.2 OsclSelect::OsclSelect ([Oscl\\_DefAlloc](#) \* *erralloc*, [Oscl\\_DefAlloc](#) \* *schedalloc*, const char \* *name*, int32 *reserve* = 10, bool *heapcheck* = false, FILE \* *output* = NULL) [inline]**

## 7.207.3 Field Documentation

**7.207.3.1 [Oscl\\_DefAlloc](#)\* OsclSelect::iErrAlloc**

**7.207.3.2 bool OsclSelect::iHeapCheck**

**7.207.3.3 bool OsclSelect::iOsclBase**

**7.207.3.4 bool OsclSelect::iOsclErrorTrap**

**7.207.3.5 bool OsclSelect::iOsclLogger**

**7.207.3.6 bool OsclSelect::iOsclMemory**

**7.207.3.7 bool OsclSelect::iOsclScheduler**

**7.207.3.8 FILE\* OsclSelect::iOutputFile**

**7.207.3.9 [Oscl\\_DefAlloc](#)\* OsclSelect::iSchedulerAlloc**

**7.207.3.10 const char\* OsclSelect::iSchedulerName**

**7.207.3.11 int32 OsclSelect::iSchedulerReserve**

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

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

## 7.208 OsclSemaphore Class Reference

```
#include <oscl_semaphore.h>
```

### Public Methods

- OSCL\_IMPORT\_REF OsclSemaphore ()
- OSCL\_IMPORT\_REF ~OsclSemaphore ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Create (uint32 initVal=0)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Close ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Wait ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Wait (uint32 timeout\_msec)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError TryWait ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Signal ()

### 7.208.1 Detailed Description

Class Semaphore

### 7.208.2 Constructor & Destructor Documentation

#### 7.208.2.1 OSCL\_IMPORT\_REF OsclSemaphore::OsclSemaphore ()

Class constructor

#### 7.208.2.2 OSCL\_IMPORT\_REF OsclSemaphore::~OsclSemaphore ()

Class destructor

### 7.208.3 Member Function Documentation

#### 7.208.3.1 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Close ()

Closes the Semaphore

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

#### 7.208.3.2 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Create (uint32 initVal = 0)

Creates the Semaphore

**Parameters:**

*Intialcount*

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

**7.208.3.3 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Signal ()**

Signals that the thread is finished with the Semaphore

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

**7.208.3.4 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::TryWait ()**

Try to acquire semaphore ,if the semaphore is already acquired by another thread, calling thread immediately returns with out blocking

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns SUCCESS\_ERROR if the semaphore was acquired, SEM\_LOCKED\_ERROR if the semaphore cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.208.3.5 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Wait (uint32 timeout\_msec)**

Makes the thread to wait on the Semaphore, with a timeout.

**Parameters:**

*timeout* in milliseconds.

**Returns:**

Returns SUCCESS\_ERROR if the semaphore was aquired, WAIT\_TIMEOUT\_ERROR if the timeout expired without acquiring the semaphore, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.208.3.6 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Wait ()**

Makes the thread to wait on the Semaphore

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

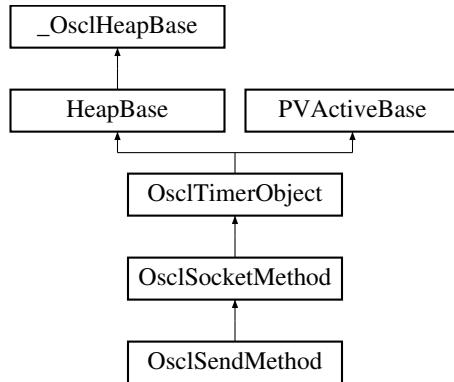
The documentation for this class was generated from the following file:

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

## 7.209 OsclSendMethod Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OsclSendMethod::



### Public Methods

- [~OsclSendMethod \(\)](#)
- [TPVSocketEvent Send \(const uint8 \\*aPtr, uint32 aLen, int32 aTimeout\)](#)
- [uint8 \\* GetSendData \(int32 \\*aLength\)](#)
- [OsclSendRequest \\* SendRequest \(\)](#)

### Static Public Methods

- [OsclSendMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 7.209.1 Constructor & Destructor Documentation

##### 7.209.1.1 OsclSendMethod::~OsclSendMethod ()

#### 7.209.2 Member Function Documentation

##### 7.209.2.1 uint8\* OsclSendMethod::GetSendData (int32 \* aLength)

##### 7.209.2.2 OsclSendMethod\* OsclSendMethod::NewL (OsclIPSocketI & c) [static]

##### 7.209.2.3 TPVSocketEvent OsclSendMethod::Send (const uint8 \*& aPtr, uint32 aLen, int32 aTimeout)

##### 7.209.2.4 OsclSendRequest\* OsclSendMethod::SendRequest () [inline]

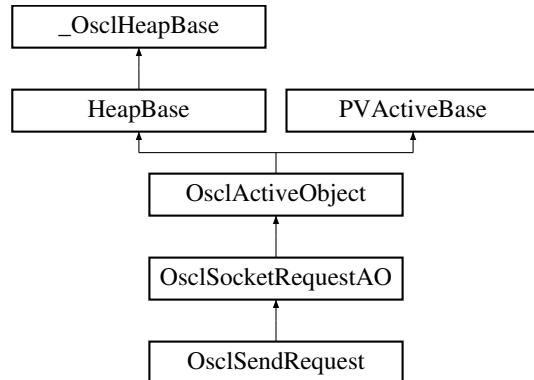
The documentation for this class was generated from the following file:

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

## 7.210 OsclSendRequest Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OsclSendRequest::



### Public Methods

- [OsclSendRequest \(OsclSocketMethod &c\)](#)
- void [Send \(const uint8 \\*&aPtr, uint32 aLen\)](#)
- void [Success \(\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)

#### 7.210.1 Constructor & Destructor Documentation

**7.210.1.1 OsclSendRequest::OsclSendRequest (OsclSocketMethod & c) [inline]**

#### 7.210.2 Member Function Documentation

**7.210.2.1 uint8\* OsclSendRequest::GetSendData (int32 \* aLength)**

**7.210.2.2 void OsclSendRequest::Send (const uint8 \*& aPtr, uint32 aLen)**

**7.210.2.3 void OsclSendRequest::Success () [virtual]**

Reimplemented from [OsclSocketRequestAO](#).

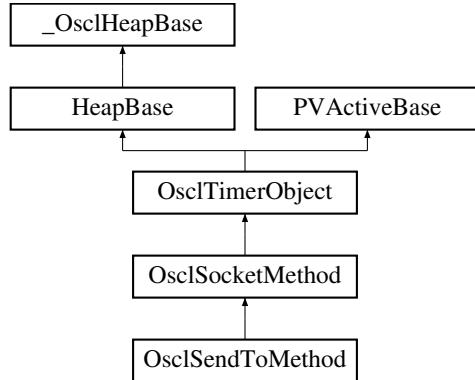
The documentation for this class was generated from the following file:

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

## 7.211 OsclSendToMethod Class Reference

```
#include <oscl_socket_send_to.h>
```

Inheritance diagram for OsclSendToMethod:::



### Public Methods

- [~OsclSendToMethod \(\)](#)
- [TPVSocketEvent SendTo \(const uint8 \\*&aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [uint8 \\* GetSendData \(int32 \\*aLength\)](#)
- [OsclSendToRequest \\* SendToRequest \(\)](#)

### Static Public Methods

- [OsclSendToMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 7.211.1 Constructor & Destructor Documentation

##### 7.211.1.1 OsclSendToMethod::~OsclSendToMethod ()

#### 7.211.2 Member Function Documentation

##### 7.211.2.1 uint8\* OsclSendToMethod::GetSendData (int32 \* aLength)

##### 7.211.2.2 OsclSendToMethod\* OsclSendToMethod::NewL (OsclIPSocketI & c) [static]

##### 7.211.2.3 TPVSocketEvent OsclSendToMethod::SendTo (const uint8 \*& aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeout)

##### 7.211.2.4 OsclSendToRequest\* OsclSendToMethod::SendToRequest () [inline]

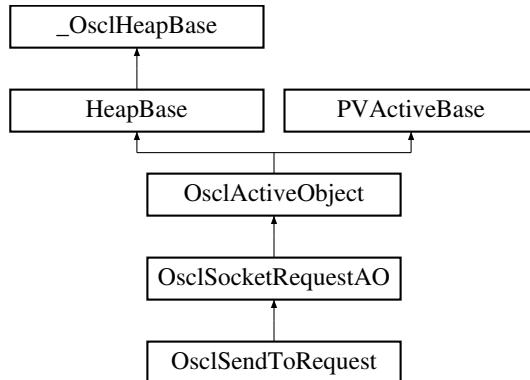
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_send\\_to.h](#)

## 7.212 OsclSendToRequest Class Reference

```
#include <oscl_socket_send_to.h>
```

Inheritance diagram for OsclSendToRequest::



### Public Methods

- [OsclSendToRequest \(OsclSocketMethod &c\)](#)
- void [SendTo \(const uint8 \\*&aPtr, uint32 aLen, OsclNetworkAddress &aAddress\)](#)
- void [Success \(\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)

#### 7.212.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.212.2 Constructor & Destructor Documentation

**7.212.2.1 OsclSendToRequest::OsclSendToRequest (OsclSocketMethod & c) [inline]**

#### 7.212.3 Member Function Documentation

**7.212.3.1 uint8\* OsclSendToRequest::GetSendData (int32 \* aLength)**

**7.212.3.2 void OsclSendToRequest::SendTo (const uint8 \*& aPtr, uint32 aLen, OsclNetworkAddress & aAddress)**

**7.212.3.3 void OsclSendToRequest::Success () [virtual]**

Reimplemented from [OsclSocketRequestAO](#).

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

- [oscl\\_socket\\_send\\_to.h](#)

## 7.213 OsclSharedPtr< TheClass > Class Template Reference

A parameterized smart pointer class.

```
#include <oscl_shared_ptr.h>
```

### Public Methods

- **OsclSharedPtr ()**  
*Constructor.*
- **OsclSharedPtr (TheClass \*inClassPtr, OsclRefCounter \*in\_refcnt)**  
*Constructor.*
- **OsclSharedPtr (const OsclSharedPtr &inSharedPtr)**  
*Copy constructor.*
- **virtual ~OsclSharedPtr ()**  
*Destructor.*
- **TheClass \* operator → ()**  
• **TheClass & operator \* ()**  
*The indirection operator returns a reference to an object of the parameterized type.*
- **operator TheClass \* ()**  
*Casting operator.*
- **TheClass \* GetRep ()**  
*Use this function to get a pointer to the wrapped object.*
- **OsclRefCounter \* GetRefCounter ()**  
*Get the refcount pointer. This should primarily be used for conversion operations.*
- **int get\_count ()**  
*Get a count of how many references to the object exist.*
- **void Bind (const OsclSharedPtr &inHandle)**  
*Use this function to bind an existing OsclSharedPtr to a already-wrapped object.*
- **void Bind (TheClass \*ptr, OsclRefCounter \*in\_refcnt)**  
*Use this function to bind an existing OsclSharedPtr to a new (unwrapped) object.*
- **void Unbind ()**  
*Use this function of unbind an existing OsclSharedPtr.*
- **OsclSharedPtr & operator= (const OsclSharedPtr &inSharedPtr)**  
*Assignment operator.*
- **bool operator== (const OsclSharedPtr &b) const**  
*Test for equality to see if two PVHandles wrap the same object.*

### 7.213.1 Detailed Description

**template<class TheClass> class OsclSharedPtr< TheClass >**

A parameterized smart pointer class.

### 7.213.2 Constructor & Destructor Documentation

**7.213.2.1 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr () [inline]**

Constructor.

**7.213.2.2 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr (TheClass \* *inClassPtr*, OsclRefCounter \* *in\_refcnt*) [inline]**

Constructor.

**Parameters:**

*inClassPtr* A pointer to an instance of the parameterized type that the new OsclSharedPtr will wrap.

**7.213.2.3 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr (const OsclSharedPtr< TheClass > & *inSharedPtr*) [inline]**

Copy constructor.

**7.213.2.4 template<class TheClass> virtual OsclSharedPtr< TheClass >::~OsclSharedPtr () [inline, virtual]**

Destructor.

### 7.213.3 Member Function Documentation

**7.213.3.1 template<class TheClass> int OsclSharedPtr< TheClass >::get\_count () [inline]**

Get a count of how many references to the object exist.

**7.213.3.2 template<class TheClass> OsclRefCounter\* OsclSharedPtr< TheClass >::GetRefCounter () [inline]**

Get the refcount pointer. This should primarily be used for conversion operations.

**7.213.3.3 template<class TheClass> TheClass\* OsclSharedPtr< TheClass >::GetRep () [inline]**

Use this function to get a pointer to the wrapped object.

**7.213.3.4 template<class TheClass> TheClass& OsclSharedPtr< TheClass >::operator \* () [inline]**

The indirection operator returns a reference to an object of the parameterized type.

**7.213.3.5 template<class TheClass> OsclSharedPtr< TheClass >::operator TheClass \* () [inline]**

Casting operator.

**7.213.3.6 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.

**7.213.3.7 template<class TheClass> OsclSharedPtr& OsclSharedPtr< TheClass >::operator=(const OsclSharedPtr< TheClass > & *inSharedPtr*) [inline]**

Assignment operator.

**7.213.3.8 template<class TheClass> void OsclSharedPtr< TheClass >::Unbind () [inline]**

Use this function of unbind an existing OsclSharedPtr.

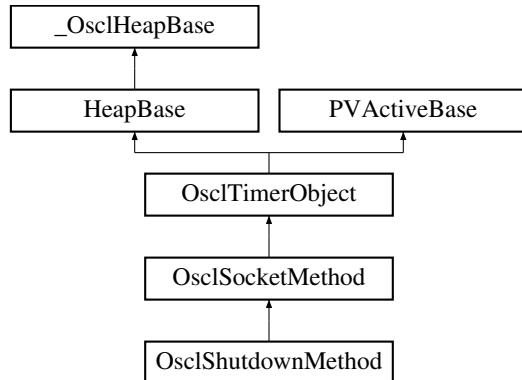
The documentation for this class was generated from the following file:

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

## 7.214 OsclShutdownMethod Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OsclShutdownMethod::



### Public Methods

- `~OsclShutdownMethod ()`
- `TPVSocketEvent Shutdown (TPVSocketShutdown aHow, int32 aTimeout)`
- `OsclShutdownRequest * ShutdownRequest ()`

### Static Public Methods

- `OsclShutdownMethod * NewL (OsclIPSocketI &c)`

#### 7.214.1 Constructor & Destructor Documentation

##### 7.214.1.1 OsclShutdownMethod::~OsclShutdownMethod ()

#### 7.214.2 Member Function Documentation

##### 7.214.2.1 OsclShutdownMethod\* OsclShutdownMethod::NewL (OsclIPSocketI &c) [static]

##### 7.214.2.2 TPVSocketEvent OsclShutdownMethod::Shutdown (TPVSocketShutdown aHow, int32 aTimeout)

##### 7.214.2.3 OsclShutdownRequest\* OsclShutdownMethod::ShutdownRequest () [inline]

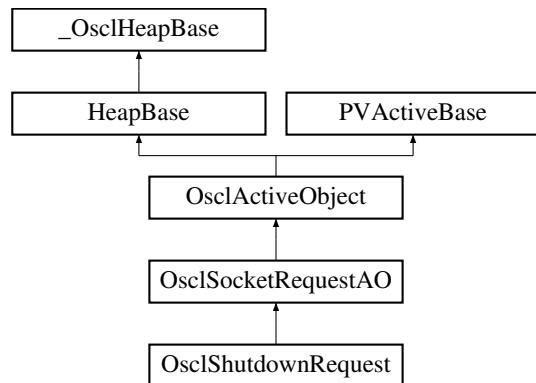
The documentation for this class was generated from the following file:

- `oscl_socket_shutdown.h`

## 7.215 OsclShutdownRequest Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OsclShutdownRequest::



### Public Methods

- [OsclShutdownRequest \(OsclSocketMethod &c\)](#)
- [void Shutdown \(TPVSocketShutdown aHow\)](#)

#### 7.215.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.215.2 Constructor & Destructor Documentation

**7.215.2.1 OsclShutdownRequest::OsclShutdownRequest ([OsclSocketMethod & c](#)) [inline]**

#### 7.215.3 Member Function Documentation

**7.215.3.1 void OsclShutdownRequest::Shutdown ([TPVSocketShutdown aHow](#))**

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

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

## 7.216 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 >
```

#### 7.216.1 Constructor & Destructor Documentation

**7.216.1.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> OsclSingleton< T, ID, Registry >::OsclSingleton () [inline]**

**7.216.1.2 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> OsclSingleton< T, ID, Registry >::~OsclSingleton () [inline]**

#### 7.216.2 Member Function Documentation

**7.216.2.1 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.

**7.216.2.2 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.

**7.216.2.3 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> bool  
OsclSingleton< T, ID, Registry >::set () [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

### 7.216.3 Field Documentation

**7.216.3.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> T\*  
OsclSingleton< T, ID, Registry >::\_Ptr [protected]**

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

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

## 7.217 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](#)

#### 7.217.1 Member Function Documentation

**7.217.1.1 OSCL\_IMPORT\_REF [OsclAny](#)\* OsclSingletonRegistry::getInstance (uint32 *ID*, int32 & *error*) [static]**

**7.217.1.2 OSCL\_IMPORT\_REF [OsclAny](#)\* OsclSingletonRegistry::lockAndGetInstance (uint32 *ID*, int32 & *error*) [static]**

**7.217.1.3 OSCL\_IMPORT\_REF void OsclSingletonRegistry::registerInstance ([OsclAny](#) \* *ptr*, uint32 *ID*, int32 & *error*) [static]**

**7.217.1.4 OSCL\_IMPORT\_REF void OsclSingletonRegistry::registerInstanceAndUnlock ([OsclAny](#) \* *ptr*, uint32 *ID*, int32 & *error*) [static]**

#### 7.217.2 Friends And Related Function Documentation

**7.217.2.1 friend class OsclBase [friend]**

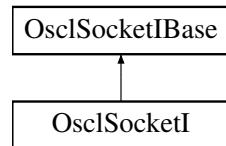
The documentation for this class was generated from the following file:

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

## 7.218 OsclSocketI Class Reference

```
#include <oscl_socket_imp_pv.h>
```

Inheritance diagram for OsclSocketI::



### Public Methods

- `~OsclSocketI ()`
- `int32 Open (OsclSocketServI &aServer, uint32 addrFamily, uint32 sockType, uint32 protocol)`
- `int32 Open (OsclSocketServI &aServer)`
- `int32 Bind (OsclNetworkAddress &anAddr)`
- `int32 SetSockOpt (TPVSocketOptionLevel aOptionLevel, TPVSocketOptionName aOptionName, OsclAny *aOptionValue, int32 aOptionLen)`
- `int32 GetPeerName (OsclNetworkAddress &peerName)`
- `int32 Join (OsclNetworkAddress &anAddr)`
- `int32 Close ()`
- `int32 Listen (uint32 qSize)`
- `int32 SetRecvBufferSize (uint32 size)`
- `TPVSocketEvent ThreadLogoff ()`
- `TPVSocketEvent ThreadLogon (OsclSocketServI *aServ)`
- `void Connect (ConnectParam &, OsclSocketRequestAO &)`
- `void Accept (AcceptParam &, OsclSocketRequestAO &)`
- `void Shutdown (ShutdownParam &, OsclSocketRequestAO &)`
- `void Send (SendParam &, OsclSocketRequestAO &)`
- `void SendSuccess (SendParam &)`
- `void SendTo (SendToParam &, OsclSocketRequestAO &)`
- `void SendToSuccess (SendToParam &)`
- `void Recv (RecvParam &, OsclSocketRequestAO &)`
- `void RecvSuccess (RecvParam &)`
- `void RecvFrom (RecvFromParam &, OsclSocketRequestAO &)`
- `void RecvFromSuccess (RecvFromParam &)`
- `TOsclSocket Socket ()`
- `void ProcessConnect (OsclSocketServRequestQElem *)`
- `void ProcessShutdown (OsclSocketServRequestQElem *)`
- `void ProcessAccept (OsclSocketServRequestQElem *)`
- `void ProcessSendTo (OsclSocketServRequestQElem *)`
- `void ProcessRecvFrom (OsclSocketServRequestQElem *)`
- `void ProcessSend (OsclSocketServRequestQElem *)`
- `void ProcessRecv (OsclSocketServRequestQElem *)`
- `PVLogger * Logger ()`

## Static Public Methods

- `OsclSocketI * NewL (Oscl_DefAlloc &a)`
- `bool MakeAddr (OsclNetworkAddress &in, TOsclSockAddr &addr)`
- `void MakeAddr (TOsclSockAddr &in, OsclNetworkAddress &addr)`
- `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`

### 7.218.1 Detailed Description

Socket implementation class

### 7.218.2 Constructor & Destructor Documentation

#### 7.218.2.1 `OsclSocketI::~OsclSocketI ()`

### 7.218.3 Member Function Documentation

#### 7.218.3.1 `void OsclSocketI::Accept (AcceptParam &, OsclSocketRequestAO &) [virtual]`

Implements `OsclSocketIBase`.

#### 7.218.3.2 `int32 OsclSocketI::Bind (OsclNetworkAddress & anAddr) [virtual]`

Implements `OsclSocketIBase`.

#### 7.218.3.3 `int32 OsclSocketI::Close () [virtual]`

Implements `OsclSocketIBase`.

#### 7.218.3.4 `void OsclSocketI::Connect (ConnectParam &, OsclSocketRequestAO &) [virtual]`

Implements `OsclSocketIBase`.

**7.218.3.5 int32 OsclSocketI::GetPeerName ([OsclNetworkAddress](#) & *peerName*)**

**7.218.3.6 int32 OsclSocketI::Join ([OsclNetworkAddress](#) & *anAddr*) [virtual]**

Implements [OsclSocketIBase](#).

**7.218.3.7 int32 OsclSocketI::Listen (uint32 *qSize*) [virtual]**

Implements [OsclSocketIBase](#).

**7.218.3.8 [PVLogger](#)\* OsclSocketI::Logger () [inline]**

**7.218.3.9 void OsclSocketI::MakeAddr ([TOscIAddr](#) & *in*, [OsclNetworkAddress](#) & *addr*) [static]**

**7.218.3.10 bool OsclSocketI::MakeAddr ([OsclNetworkAddress](#) & *in*, [TOscIAddr](#) & *addr*) [static]**

**7.218.3.11 void OsclSocketI::MakeMulticastGroupInformation ([TIpMReq](#) & *in*, [OsclIpMReq](#) & *addr*) [static]**

**7.218.3.12 bool OsclSocketI::MakeMulticastGroupInformation ([OsclIpMReq](#) & *in*, [TIpMReq](#) & *addr*) [static]**

**7.218.3.13 OsclSocketI\* OsclSocketI::NewL ([Oscl\\_DefAlloc](#) & *a*) [static]**

**7.218.3.14 int32 OsclSocketI::Open ([OsclSocketServI](#) & *aServer*) [virtual]**

Implements [OsclSocketIBase](#).

**7.218.3.15 int32 OsclSocketI::Open ([OsclSocketServI](#) & *aServer*, uint32 *addrFamily*, uint32 *sockType*, uint32 *protocol*) [virtual]**

Implements [OsclSocketIBase](#).

- 7.218.3.16 void OsclSocketI::ProcessAccept ([OsclSocketServRequestQElem](#) \*)
- 7.218.3.17 void OsclSocketI::ProcessConnect ([OsclSocketServRequestQElem](#) \*)
- 7.218.3.18 void OsclSocketI::ProcessRecv ([OsclSocketServRequestQElem](#) \*)
- 7.218.3.19 void OsclSocketI::ProcessRecvFrom ([OsclSocketServRequestQElem](#) \*)
- 7.218.3.20 void OsclSocketI::ProcessSend ([OsclSocketServRequestQElem](#) \*)
- 7.218.3.21 void OsclSocketI::ProcessSendTo ([OsclSocketServRequestQElem](#) \*)
- 7.218.3.22 void OsclSocketI::ProcessShutdown ([OsclSocketServRequestQElem](#) \*)
- 7.218.3.23 void OsclSocketI::Recv ([RecvParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

- 7.218.3.24 void OsclSocketI::RecvFrom ([RecvFromParam](#) &, [OsclSocketRequestAO](#) && [virtual])

Implements [OsclSocketIBase](#).

- 7.218.3.25 void OsclSocketI::RecvFromSuccess ([RecvFromParam](#) &) [virtual]

Implements [OsclSocketIBase](#).

- 7.218.3.26 void OsclSocketI::RecvSuccess ([RecvParam](#) &) [virtual]

Implements [OsclSocketIBase](#).

- 7.218.3.27 void OsclSocketI::Send ([SendParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

- 7.218.3.28 void OsclSocketI::SendSuccess ([SendParam](#) &) [virtual]

Implements [OsclSocketIBase](#).

- 7.218.3.29 void OsclSocketI::SendTo ([SendToParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

- 7.218.3.30 void OsclSocketI::SendToSuccess ([SendToParam](#) &) [virtual]

Implements [OsclSocketIBase](#).

**7.218.3.31** `int32 OsclSocketI::SetRecvBufferSize (uint32 size)`

**7.218.3.32** `int32 OsclSocketI::SetSockOpt (TPVSocketOptionLevel aOptionLevel,  
TPVSocketOptionName aOptionName, OsclAny * aOptionValue, int32 aOptionLen)`

**7.218.3.33** `void OsclSocketI::Shutdown (ShutdownParam &, OsclSocketRequestAO &)`  
[virtual]

Implements [OsclSocketIBase](#).

**7.218.3.34** `TOsclSocket OsclSocketI::Socket () [inline]`

**7.218.3.35** `TPVSocketEvent OsclSocketI::ThreadLogoff ()`

**7.218.3.36** `TPVSocketEvent OsclSocketI::ThreadLogon (OsclSocketServI * aServ)`

## 7.218.4 Friends And Related Function Documentation

**7.218.4.1** `friend class OsclAcceptRequest [friend]`

**7.218.4.2** `friend class OsclConnectRequest [friend]`

**7.218.4.3** `friend class OsclRecvFromRequest [friend]`

**7.218.4.4** `friend class OsclRecvRequest [friend]`

**7.218.4.5** `friend class OsclSendRequest [friend]`

**7.218.4.6** `friend class OsclSendToRequest [friend]`

**7.218.4.7** `friend class OsclShutdownRequest [friend]`

**7.218.4.8** `friend class OsclTCPSocket [friend]`

Reimplemented from [OsclSocketIBase](#).

**7.218.4.9** `friend class OsclUDPSocket [friend]`

Reimplemented from [OsclSocketIBase](#).

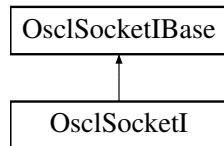
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_imp\\_pv.h](#)

## 7.219 OsclSocketIBase Class Reference

```
#include <oscl_socket_imp_base.h>
```

Inheritance diagram for OsclSocketIBase::



### Public Methods

- virtual ~OsclSocketIBase ()
- virtual int32 Open (OsclSocketServI &aServer, uint32 addrFamily, uint32 sockType, uint32 proto-col)=0
- virtual int32 Open (OsclSocketServI &aServer)=0
- virtual int32 Bind (OsclNetworkAddress &anAddr)=0
- virtual int32 Join (OsclNetworkAddress &anAddr)=0
- virtual int32 Close ()=0
- virtual int32 Listen (uint32 qSize)=0
- virtual void Connect (ConnectParam &, OsclSocketRequestAO &)=0
- virtual void Accept (AcceptParam &, OsclSocketRequestAO &)=0
- virtual void Shutdown (ShutdownParam &, OsclSocketRequestAO &)=0
- virtual void Send (SendParam &, OsclSocketRequestAO &)=0
- virtual void SendSuccess (SendParam &)=0
- virtual void SendTo (SendToParam &, OsclSocketRequestAO &)=0
- virtual void SendToSuccess (SendToParam &)=0
- virtual void Recv (RecvParam &, OsclSocketRequestAO &)=0
- virtual void RecvSuccess (RecvParam &)=0
- virtual void RecvFrom (RecvFromParam &, OsclSocketRequestAO &)=0
- virtual void RecvFromSuccess (RecvFromParam &)=0
- virtual void BindAsync (BindParam &, OsclSocketRequestAO &)
- virtual void ListenAsync (ListenParam &, OsclSocketRequestAO &)
- void CancelFxn (TPVSocketFxn)

### Static Public Methods

- bool HasAsyncBind ()
- bool HasAsyncListen ()

### Protected Methods

- OsclSocketIBase (Oscl\_DefAlloc &a)
- virtual void CancelConnect ()=0
- virtual void CancelAccept ()=0
- virtual void CancelShutdown ()=0
- virtual void CancelSend ()=0

- virtual void [CancelSendTo \(\)=0](#)
- virtual void [CancelRecv \(\)=0](#)
- virtual void [CancelRecvFrom \(\)=0](#)
- virtual void [CancelBind \(\)](#)
- virtual void [CancelListen \(\)](#)
- virtual bool [IsOpen \(\)=0](#)

## Static Protected Methods

- int [GetShutdown \(TPVSocketShutdown aOsclVal\)](#)

## Protected Attributes

- [Oscl\\_DefAlloc & iAlloc](#)
- [OsclSocketServI \\* iSocketServ](#)

## Friends

- class [OsclSocketRequest](#)
- class [OsclSocketMethod](#)
- class [OsclSocketRequestAO](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

### 7.219.1 Detailed Description

Socket implementation base class

### 7.219.2 Constructor & Destructor Documentation

**7.219.2.1** virtual OsclSocketIBase::~OsclSocketIBase () [virtual]

**7.219.2.2** OsclSocketIBase::OsclSocketIBase ([Oscl\\_DefAlloc & a](#)) [protected]

### 7.219.3 Member Function Documentation

**7.219.3.1** virtual void OsclSocketIBase::Accept ([AcceptParam &](#), [OsclSocketRequestAO &](#)) [pure virtual]

Implemented in [OsclSocketI](#).

**7.219.3.2** virtual int32 OsclSocketIBase::Bind ([OsclNetworkAddress & anAddr](#)) [pure virtual]

Implemented in [OsclSocketI](#).

- 7.219.3.3 **virtual void OsclSocketIBase::BindAsync (BindParam &, OsclSocketRequestAO &)**  
[inline, virtual]
- 7.219.3.4 **virtual void OsclSocketIBase::CancelAccept ()** [protected, pure virtual]
- 7.219.3.5 **virtual void OsclSocketIBase::CancelBind ()** [inline, protected, virtual]
- 7.219.3.6 **virtual void OsclSocketIBase::CancelConnect ()** [protected, pure virtual]
- 7.219.3.7 **void OsclSocketIBase::CancelFxn (TPVSocketFxn)**
- 7.219.3.8 **virtual void OsclSocketIBase::CancelListen ()** [inline, protected, virtual]
- 7.219.3.9 **virtual void OsclSocketIBase::CancelRecv ()** [protected, pure virtual]
- 7.219.3.10 **virtual void OsclSocketIBase::CancelRecvFrom ()** [protected, pure virtual]
- 7.219.3.11 **virtual void OsclSocketIBase::CancelSend ()** [protected, pure virtual]
- 7.219.3.12 **virtual void OsclSocketIBase::CancelSendTo ()** [protected, pure virtual]
- 7.219.3.13 **virtual void OsclSocketIBase::CancelShutdown ()** [protected, pure virtual]
- 7.219.3.14 **virtual int32 OsclSocketIBase::Close ()** [pure virtual]

Implemented in [OsclSocketI](#).

- 7.219.3.15 **virtual void OsclSocketIBase::Connect (ConnectParam &, OsclSocketRequestAO &)**  
[pure virtual]

Implemented in [OsclSocketI](#).

- 7.219.3.16 **int OsclSocketIBase::GetShutdown (TPVSocketShutdown aOsclVal)** [static,  
protected]
- 7.219.3.17 **bool OsclSocketIBase::HasAsyncBind ()** [static]
- 7.219.3.18 **bool OsclSocketIBase::HasAsyncListen ()** [static]
- 7.219.3.19 **virtual bool OsclSocketIBase::IsOpen ()** [protected, pure virtual]
- 7.219.3.20 **virtual int32 OsclSocketIBase::Join (OsclNetworkAddress & anAddr)** [pure  
virtual]

Implemented in [OsclSocketI](#).

- 7.219.3.21 **virtual int32 OsclSocketIBase::Listen (uint32 qSize)** [pure virtual]

Implemented in [OsclSocketI](#).

**7.219.3.22** `virtual void OsclSocketIBase::ListenAsync (ListenParam &, OsclSocketRequestAO &)`  
[inline, virtual]

**7.219.3.23** `virtual int32 OsclSocketIBase::Open (OsclSocketServI & aServer)` [pure  
virtual]

Implemented in [OsclSocketI](#).

**7.219.3.24** `virtual int32 OsclSocketIBase::Open (OsclSocketServI & aServer, uint32 addrFamily,  
uint32 sockType, uint32 protocol)` [pure virtual]

Implemented in [OsclSocketI](#).

**7.219.3.25** `virtual void OsclSocketIBase::Recv (RecvParam &, OsclSocketRequestAO &)` [pure  
virtual]

Implemented in [OsclSocketI](#).

**7.219.3.26** `virtual void OsclSocketIBase::RecvFrom (RecvFromParam &, OsclSocketRequestAO  
&)` [pure virtual]

Implemented in [OsclSocketI](#).

**7.219.3.27** `virtual void OsclSocketIBase::RecvFromSuccess (RecvFromParam &)` [pure  
virtual]

Implemented in [OsclSocketI](#).

**7.219.3.28** `virtual void OsclSocketIBase::RecvSuccess (RecvParam &)` [pure virtual]

Implemented in [OsclSocketI](#).

**7.219.3.29** `virtual void OsclSocketIBase::Send (SendParam &, OsclSocketRequestAO &)` [pure  
virtual]

Implemented in [OsclSocketI](#).

**7.219.3.30** `virtual void OsclSocketIBase::SendSuccess (SendParam &)` [pure virtual]

Implemented in [OsclSocketI](#).

**7.219.3.31** `virtual void OsclSocketIBase::SendTo (SendToParam &, OsclSocketRequestAO &)`  
[pure virtual]

Implemented in [OsclSocketI](#).

**7.219.3.32 virtual void OsclSocketIBase::SendToSuccess ([SendToParam](#) &)** [pure virtual]

Implemented in [OsclSocketI](#).

**7.219.3.33 virtual void OsclSocketIBase::Shutdown ([ShutdownParam](#) &, [OsclSocketRequestAO](#) &)** [pure virtual]

Implemented in [OsclSocketI](#).

## 7.219.4 Friends And Related Function Documentation

**7.219.4.1 friend class OsclSocketMethod** [friend]

**7.219.4.2 friend class OsclSocketRequest** [friend]

**7.219.4.3 friend class OsclSocketRequestAO** [friend]

**7.219.4.4 friend class OsclTCPSocket** [friend]

Reimplemented in [OsclSocketI](#).

**7.219.4.5 friend class OsclUDPSocket** [friend]

Reimplemented in [OsclSocketI](#).

## 7.219.5 Field Documentation

**7.219.5.1 [Oscl\\_DefAlloc](#)& OsclSocketIBase::iAlloc** [protected]

**7.219.5.2 [OsclSocketServI](#)\* OsclSocketIBase::iSocketServ** [protected]

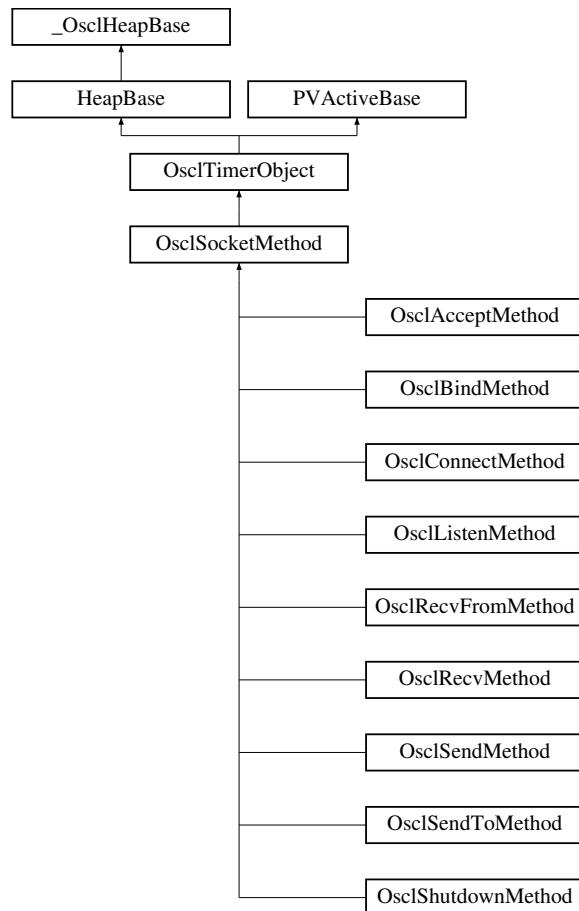
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_imp\\_base.h](#)

## 7.220 OsclSocketMethod Class Reference

```
#include <oscl_socket_method.h>
```

Inheritance diagram for OsclSocketMethod::



### Public Methods

- [OsclSocketMethod \(OsclIPSocketI &aContainer, const char \\*name, TPVSocketFxn fxn\)](#)
- virtual [~OsclSocketMethod \(\)](#)
- void [Abort \(\)](#)
- void [AbortAll \(\)](#)
- void [CancelMethod \(\)](#)
- [Oscl\\_DefAlloc & Alloc \(\)](#)
- [TPVSocketEvent ThreadLogon \(\)](#)
- [TPVSocketEvent ThreadLogoff \(\)](#)

### Data Fields

- [OsclIPSocketI & iContainer](#)
- [TPVSocketFxn iSocketFxn](#)

## Protected Methods

- void [ConstructL \(OsclSocketRequestAO \\*aAO\)](#)
- bool [StartMethod \(int32 aTimeoutMsec\)](#)
- void [MethodDone \(\)](#)
- void [Run \(\)](#)

## Protected Attributes

- [OsclSocketRequestAO \\* iSocketRequestAO](#)

### 7.220.1 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.

### 7.220.2 Constructor & Destructor Documentation

**7.220.2.1 OsclSocketMethod::OsclSocketMethod** ([OsclIPSocketI & aContainer](#), [const char \\* name](#), [TPVSocketFxn ffn](#)) [inline]

**7.220.2.2 virtual OsclSocketMethod::~OsclSocketMethod ()** [inline, virtual]

### 7.220.3 Member Function Documentation

**7.220.3.1 void OsclSocketMethod::Abort ()** [inline]

**7.220.3.2 void OsclSocketMethod::AbortAll ()** [inline]

**7.220.3.3 Oscl\_DefAlloc& OsclSocketMethod::Alloc ()** [inline]

**7.220.3.4 void OsclSocketMethod::CancelMethod ()** [inline]

**7.220.3.5 void OsclSocketMethod::ConstructL (OsclSocketRequestAO \* aAO)** [inline, protected]

**7.220.3.6 void OsclSocketMethod::MethodDone ()** [inline, protected]

**7.220.3.7 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](#).

**7.220.3.8 bool OsclSocketMethod::StartMethod (int32 *aTimeoutMsec*) [protected]**

**7.220.3.9 TPVSocketEvent OsclSocketMethod::ThreadLogoff ()**

**7.220.3.10 TPVSocketEvent OsclSocketMethod::ThreadLogon ()**

## 7.220.4 Field Documentation

**7.220.4.1 OsclIPSocketI& OsclSocketMethod::iContainer**

**7.220.4.2 TPVSocketFxn OsclSocketMethod::iSocketFxn**

**7.220.4.3 OsclSocketRequestAO\* OsclSocketMethod::iSocketRequestAO [protected]**

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

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

## 7.221 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](#) ()

### 7.221.1 Detailed Description

Socket event observer. The client implements this to get asynchronous command completion.

### 7.221.2 Constructor & Destructor Documentation

7.221.2.1 virtual [OsclSocketObserver::~OsclSocketObserver](#) () [inline, virtual]

### 7.221.3 Member Function Documentation

7.221.3.1 virtual OSCL\_IMPORT\_REF void [OsclSocketObserver::HandleSocketEvent](#) (int32 *aId*, [TPVSocketFxn](#) *aFxn*, [TPVSocketEvent](#) *aEvent*, int32 *aError*) [pure virtual]

Socket Event callback.

#### Parameters:

*aId*: The ID that was supplied when the socket was created.

*aFxn*: Type of socket function call.

*aEvent*: Function completion event. Will be EPVSocketSuccess, EPVSocketTimeout, or EPVSocketFailure.

*aError*: When the event is EPVSocketFailure, this may contain a platform-specific error code, or zero if none is available.

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

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

## 7.222 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](#)

#### 7.222.1 Detailed Description

This class defines the request interface to the PV socket server.

#### 7.222.2 Constructor & Destructor Documentation

##### 7.222.2.1 [OsclSocketRequest::OsclSocketRequest \(\) \[inline\]](#)

#### 7.222.3 Member Function Documentation

##### 7.222.3.1 [void OsclSocketRequest::Activate \(SocketRequestParam \\* iParam, OsclSocketRequestAO & a\)](#)

##### 7.222.3.2 [void OsclSocketRequest::CancelRequest \(\)](#)

##### 7.222.3.3 [void OsclSocketRequest::Complete \(OsclSocketServRequestQElem \\*, int32 aStatus, int32 aSockErr = 0\)](#)

##### 7.222.3.4 [TPVSocketFxn OsclSocketRequest::Fxn \(\) \[inline\]](#)

#### 7.222.4 Field Documentation

##### 7.222.4.1 [SocketRequestParam\\* OsclSocketRequest::iParam](#)

##### 7.222.4.2 [OsclSocketI\\* OsclSocketRequest::iSocketI](#)

##### 7.222.4.3 [OsclSocketRequestAO\\* OsclSocketRequest::iSocketRequestAO](#)

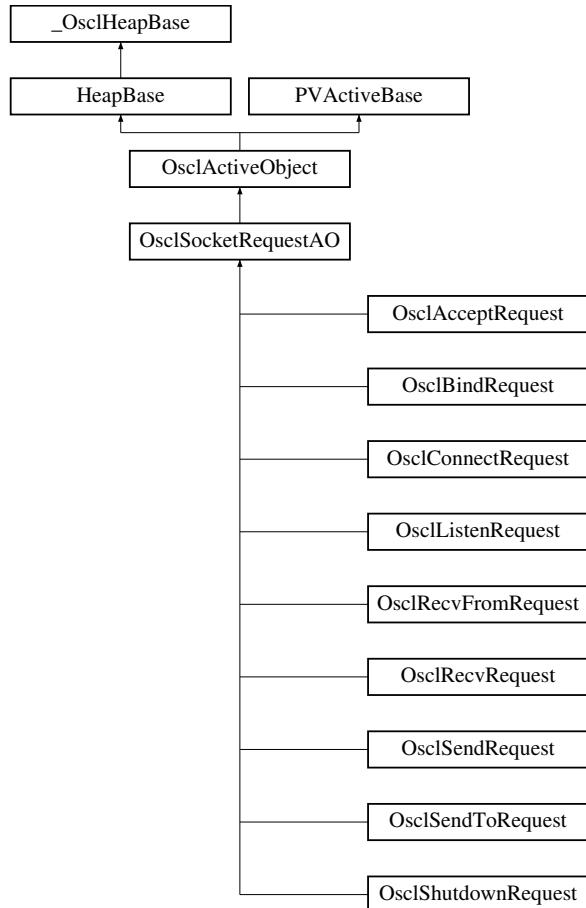
The documentation for this class was generated from the following file:

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

## 7.223 OsclSocketRequestAO Class Reference

```
#include <oscl_socket_method.h>
```

Inheritance diagram for OsclSocketRequestAO::



### Public Methods

- void [ConstructL \(\)](#)

### Protected Methods

- [OsclSocketRequestAO \(OsclSocketMethod &aContainer, const char \\*name\)](#)
- virtual [~OsclSocketRequestAO \(\)](#)
- [OsclAny \\* NewRequest \(const uint32 size\)](#)
- void [CleanupParam \(bool deallocate=false\)](#)
- void [Abort \(\)](#)
- void [RequestDone \(\)](#)
- int [GetSocketError \(\)](#)
- void [DoCancel \(\)](#)
- void [Run \(\)](#)

- virtual void [Success \(\)](#)
- [OsclSocketI \\* SocketI \(\)](#)
- [OsclSocketObserver \\* SocketObserver \(\)](#)
- uint32 [Id \(\)](#)
- [Oscl\\_DefAlloc & Alloc \(\)](#)

## Protected Attributes

- [OsclSocketMethod & iContainer](#)
- int32 [iSocketError](#)
- [SocketRequestParam \\* iParam](#)
- uint32 [iParamSize](#)

## Friends

- class [OsclSocketI](#)
- class [OsclSocketMethod](#)
- class [OsclSocketRequest](#)

### 7.223.1 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

### 7.223.2 Constructor & Destructor Documentation

**7.223.2.1 OsclSocketRequestAO::OsclSocketRequestAO ([OsclSocketMethod & aContainer](#), const char \* *name*) [inline, protected]**

**7.223.2.2 virtual OsclSocketRequestAO::~OsclSocketRequestAO () [inline, protected, virtual]**

### 7.223.3 Member Function Documentation

**7.223.3.1 void OsclSocketRequestAO::Abort () [inline, protected]**

**7.223.3.2 [Oscl\\_DefAlloc& OsclSocketRequestAO::Alloc \(\) \[inline, protected\]](#)**

**7.223.3.3 void OsclSocketRequestAO::CleanupParam (bool *deallocate* = false) [protected]**

**7.223.3.4 void OsclSocketRequestAO::ConstructL () [inline]**

**7.223.3.5 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](#).

7.223.3.6 **int OsclSocketRequestAO::GetSocketError ()** [inline, protected]

7.223.3.7 **uint32 OsclSocketRequestAO::Id ()** [inline, protected]

7.223.3.8 **OsclAny\* OsclSocketRequestAO::NewRequest (const uint32 size)** [protected]

7.223.3.9 **void OsclSocketRequestAO::RequestDone ()** [inline, protected]

7.223.3.10 **void OsclSocketRequestAO::Run ()** [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's Run() function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

Run() runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's Run() or RunError() functions.

Implements [PVActiveBase](#).

7.223.3.11 **OsclSocketI\* OsclSocketRequestAO::SocketI ()** [inline, protected]

7.223.3.12 **OsclSocketObserver\* OsclSocketRequestAO::SocketObserver ()** [inline, protected]

7.223.3.13 **virtual void OsclSocketRequestAO::Success ()** [inline, protected, virtual]

Reimplemented in [OsclRecvRequest](#), [OsclRecvFromRequest](#), [OsclSendRequest](#), and [OsclSendToRequest](#).

## 7.223.4 Friends And Related Function Documentation

7.223.4.1 **friend class OsclSocketI** [friend]

7.223.4.2 **friend class OsclSocketMethod** [friend]

7.223.4.3 **friend class OsclSocketRequest** [friend]

## 7.223.5 Field Documentation

7.223.5.1 **OsclSocketMethod& OsclSocketRequestAO::iContainer** [protected]

7.223.5.2 **SocketRequestParam\* OsclSocketRequestAO::iParam** [protected]

7.223.5.3 **uint32 OsclSocketRequestAO::iParamSize** [protected]

7.223.5.4 **int32 OsclSocketRequestAO::iSocketError** [protected]

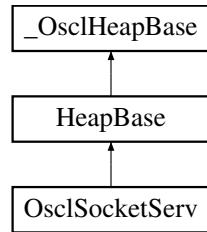
The documentation for this class was generated from the following file:

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

## 7.224 OsclSocketServ Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclSocketServ::



### Public Methods

- OSCL\_IMPORT\_REF ~OsclSocketServ ()
- OSCL\_IMPORT\_REF int32 Connect (uint32 aMessageSlots=8, bool aShareSession=false)
- OSCL\_IMPORT\_REF void Close (bool aCleanup=true)

### Static Public Methods

- OSCL\_IMPORT\_REF OsclSocketServ \* NewL (Oscl\_DefAlloc &alloc)

### Friends

- class OsclTCPSocket
- class OsclUDPSocket
- class OsclDNS

#### 7.224.1 Constructor & Destructor Documentation

##### 7.224.1.1 OSCL\_IMPORT\_REF OsclSocketServ::~OsclSocketServ ()

Destructor. The server object must be deleted using the same allocator used in the NewL call.

#### 7.224.2 Member Function Documentation

##### 7.224.2.1 OSCL\_IMPORT\_REF void OsclSocketServ::Close (bool aCleanup = true)

Close socket server. This is a synchronous method.

#### Parameters:

*aCleanup*: cleanup the socket system? the socket cleanup should only be done when all parts of the application are done using sockets.

**7.224.2.2 OSCL\_IMPORT\_REF int32 OsclSocketServ::Connect (uint32 *aMessageSlots* = 8, bool *aShareSession* = false)**

Connect to socket server. This is a synchronous method.

**Parameters:**

*Number* of message slots.

**Returns:**

Returns OsclErrNone for success, or a platform-specific code.

**7.224.2.3 OSCL\_IMPORT\_REF OsclSocketServ\* OsclSocketServ::NewL (Oscl\_DefAlloc & *alloc*)  
[static]**

Create a socket server. May leave if failure.

**Parameters:**

*alloc*: Memory allocator.

**Returns:**

Returns pointer to socket server

**7.224.3 Friends And Related Function Documentation****7.224.3.1 friend class OsclDNS [friend]****7.224.3.2 friend class OsclTCPSocket [friend]****7.224.3.3 friend class OsclUDPSocket [friend]**

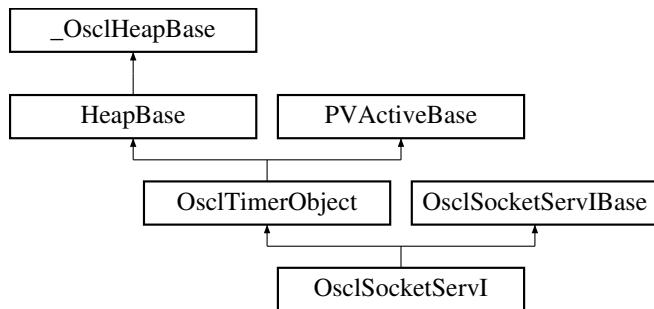
The documentation for this class was generated from the following file:

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

## 7.225 OsclSocketServI Class Reference

```
#include <oscl_socket_serv_imp_pv.h>
```

Inheritance diagram for OsclSocketServI::



### Public Methods

- int32 [Connect](#) (uint32 aMessageSlots, bool aShareSession)
- void [Close](#) (bool)
- bool [IsServerThread](#) ()

### Static Public Methods

- OsclSocketServI \* [NewL](#) (Oscl\_DefAlloc &a)

### Friends

- class [OsclSocketServRequestList](#)
- class [LoopbackSocket](#)
- class [OsclTCPSocketI](#)
- class [OsclUDPSocketI](#)
- class [OsclSocketI](#)
- class [OsclDNSI](#)
- class [OsclSocketRequest](#)
- class [OsclSocketServ](#)

### 7.225.1 Detailed Description

PV socket server implementation

### 7.225.2 Member Function Documentation

#### 7.225.2.1 void OsclSocketServI::Close (bool) [virtual]

Implements [OsclSocketServIBase](#).

**7.225.2.2 int32 OsclSocketServI::Connect (uint32 *aMessageSlots*, bool *aShareSession*)**  
[virtual]

Implements [OsclSocketServIBase](#).

**7.225.2.3 bool OsclSocketServI::IsServerThread ()**

**7.225.2.4 OsclSocketServI\* OsclSocketServI::NewL (Oscl\_DefAlloc & *a*) [static]**

### 7.225.3 Friends And Related Function Documentation

**7.225.3.1 friend class LoopbackSocket [friend]**

**7.225.3.2 friend class OsclDNSI [friend]**

**7.225.3.3 friend class OsclSocketI [friend]**

**7.225.3.4 friend class OsclSocketRequest [friend]**

**7.225.3.5 friend class OsclSocketServ [friend]**

**7.225.3.6 friend class OsclSocketServRequestList [friend]**

**7.225.3.7 friend class OsclTCPSocketI [friend]**

**7.225.3.8 friend class OsclUDPSocketI [friend]**

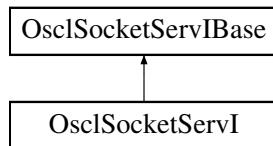
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_serv\\_imp\\_pv.h](#)

## 7.226 OsclSocketServIBase Class Reference

```
#include <oscl_socket_serv_imp_base.h>
```

Inheritance diagram for OsclSocketServIBase::



### Public Methods

- virtual ~[OsclSocketServIBase](#) ()
- 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

- [OsclSocketServIBase \(Oscl\\_DefAlloc &a\)](#)
- [TSocketServState State \(\) const](#)
- [bool IsServConnected \(\) const](#)

### Protected Attributes

- [Oscl\\_DefAlloc & iAlloc](#)
- [TSocketServState iServState](#)
- [int iServError](#)

### 7.226.1 Detailed Description

Base class common to all implementations

### 7.226.2 Member Enumeration Documentation

#### 7.226.2.1 enum OsclSocketServIBase::TSocketServState [protected]

Enumeration values:

[ESocketServ\\_Idle](#)

**ESocketServ\_Connected**

**ESocketServ\_Error**

### 7.226.3 Constructor & Destructor Documentation

**7.226.3.1 virtual OsclSocketServIBase::~OsclSocketServIBase () [inline, virtual]**

**7.226.3.2 OsclSocketServIBase::OsclSocketServIBase ([Oscl\\_DefAlloc](#) & *a*) [inline, protected]**

### 7.226.4 Member Function Documentation

**7.226.4.1 virtual void OsclSocketServIBase::Close (bool) [pure virtual]**

Implemented in [OsclSocketServI](#).

**7.226.4.2 virtual int32 OsclSocketServIBase::Connect (uint32 *aMessageSlots*, bool *aShareSession*) [pure virtual]**

Implemented in [OsclSocketServI](#).

**7.226.4.3 bool OsclSocketServIBase::IsServConnected () const [inline, protected]**

**7.226.4.4 [TSocketServState](#) OsclSocketServIBase::State () const [inline, protected]**

### 7.226.5 Field Documentation

**7.226.5.1 [Oscl\\_DefAlloc](#)& OsclSocketServIBase::iAlloc [protected]**

**7.226.5.2 [PVLogger](#)\* OsclSocketServIBase::iLogger**

**7.226.5.3 int OsclSocketServIBase::iServerError [protected]**

**7.226.5.4 [TSocketServState](#) OsclSocketServIBase::iServState [protected]**

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

- [oscl\\_socket\\_serv\\_imp\\_base.h](#)

## 7.227 OsclSocketServRequestList Class Reference

```
#include <oscl_socket_serv_imp_reqlist.h>
```

### Public Methods

- [OsclSocketServRequestList \(\)](#)
- [void Add \(OsclSocketRequest \\*\)](#)
- [void StartCancel \(OsclSocketRequest \\*\)](#)
- [void Open \(OsclSocketServI \\*s\)](#)
- [void Close \(\)](#)
- [void Wakeup \(\)](#)
- [void WaitOnRequests \(\)](#)
- [void Remove \(OsclSocketServRequestQElem \\*aElem\)](#)

### Friends

- class [OsclSocketServI](#)

### 7.227.1 Detailed Description

PV socket server request queue

### 7.227.2 Constructor & Destructor Documentation

#### 7.227.2.1 OsclSocketServRequestList::OsclSocketServRequestList ()

### 7.227.3 Member Function Documentation

#### 7.227.3.1 void OsclSocketServRequestList::Add ([OsclSocketRequest \\*](#))

#### 7.227.3.2 void OsclSocketServRequestList::Close ()

#### 7.227.3.3 void OsclSocketServRequestList::Open ([OsclSocketServI \\* s](#))

#### 7.227.3.4 void OsclSocketServRequestList::Remove ([OsclSocketServRequestQElem \\* aElem](#)) [inline]

#### 7.227.3.5 void OsclSocketServRequestList::StartCancel ([OsclSocketRequest \\*](#))

#### 7.227.3.6 void OsclSocketServRequestList::WaitOnRequests ()

#### 7.227.3.7 void OsclSocketServRequestList::Wakeup ()

### 7.227.4 Friends And Related Function Documentation

#### 7.227.4.1 friend class OsclSocketServI [friend]

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

- [oscl\\_socket\\_serv\\_imp\\_reqlist.h](#)

## 7.228 OsclSocketServRequestQElem Class Reference

```
#include <oscl_socket_serv_imp_reqlist.h>
```

### Public Methods

- [OsclSocketServRequestQElem \(OsclSocketRequest \\*r\)](#)

### Data Fields

- [OsclSocketRequest \\* iSocketRequest](#)
- [uint8 iSelect](#)
- [bool iCancel](#)

#### 7.228.1 Constructor & Destructor Documentation

7.228.1.1 [OsclSocketServRequestQElem::OsclSocketServRequestQElem \(OsclSocketRequest \\* r\)](#)  
[inline]

#### 7.228.2 Field Documentation

7.228.2.1 [bool OsclSocketServRequestQElem::iCancel](#)

7.228.2.2 [uint8 OsclSocketServRequestQElem::iSelect](#)

7.228.2.3 [OsclSocketRequest\\* OsclSocketServRequestQElem::iSocketRequest](#)

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

- [oscl\\_socket\\_serv\\_imp\\_reqlist.h](#)

## 7.229 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

#### 7.229.1 Member Enumeration Documentation

##### 7.229.1.1 enum OsclSocketTOS::TPVServicePrecedence

Enumeration values:

**EPVRoutine**  
**EPVPriority**  
**EPVImmediate**  
**EPVFlash**  
**EPVOverrideFlash**  
**EPVCritic\_Ecp**  
**EPVInetControl**  
**EPVNetControl**

##### 7.229.1.2 enum OsclSocketTOS::TPVServicePriority

Enumeration values:

**EPVNoTOS**  
**EPVLDelay**  
**EPVHiThrpt**  
**EPVHiRel**

## 7.229.2 Constructor & Destructor Documentation

7.229.2.1 `OsclSocketTOS::OsclSocketTOS () [inline]`

## 7.229.3 Member Function Documentation

7.229.3.1 `void OsclSocketTOS::ClearTOS () [inline]`

7.229.3.2 `uint8 OsclSocketTOS::GetTOS () const [inline]`

7.229.3.3 `void OsclSocketTOS::SetPrecedence (TPVServicePrecedence aPrecedence) [inline]`

7.229.3.4 `void OsclSocketTOS::SetPriority (bool aMinimizeDelay, bool aMaximizeThroughput, bool MaximizeReliability) [inline]`

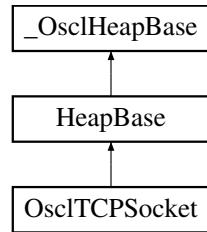
The documentation for this class was generated from the following file:

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

## 7.230 OsclTCPSocket Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclTCPSocket::



### Public Methods

- OSCL\_IMPORT\_REF ~OsclTCPSocket ()
- OSCL\_IMPORT\_REF TPVSocketEvent ThreadLogoff ()
- OSCL\_IMPORT\_REF TPVSocketEvent ThreadLogon (OsclSocketServ &aServ, OsclSocketObserver \*aObserver)
- OSCL\_IMPORT\_REF int32 Close ()
- OSCL\_IMPORT\_REF int32 Bind (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF TPVSocketEvent BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void CancelBind ()
- OSCL\_IMPORT\_REF int32 SetOptionToReuseAddress ()
- OSCL\_IMPORT\_REF int32 SetTOS (const OsclSocketTOS &aTOS)
- OSCL\_IMPORT\_REF int32 GetPeerName (OsclNetworkAddress &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 OsclTCPSocket \* GetAcceptedSocketL (uint32 aId)
- OSCL\_IMPORT\_REF uint8 \* GetRecvData (int32 \*aLength)
- OSCL\_IMPORT\_REF uint8 \* GetSendData (int32 \*aLength)
- OSCL\_IMPORT\_REF TPVSocketEvent Connect (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelConnect ()
- OSCL\_IMPORT\_REF TPVSocketEvent Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelShutdown ()
- OSCL\_IMPORT\_REF TPVSocketEvent Accept (int32 aTimeout=-1)
- OSCL\_IMPORT\_REF void CancelAccept ()
- OSCL\_IMPORT\_REF TPVSocketEvent Send (const uint8 \*aPtr, uint32 aLen, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelSend ()
- OSCL\_IMPORT\_REF TPVSocketEvent Recv (uint8 \*aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelRecv ()

## Static Public Methods

- OSCL\_IMPORT\_REF OsclTCPSocket \* **NewL** (Oscl\_DefAlloc &alloc, OsclSocketServ &aServ, OsclSocketObserver \*aObserver, uint32 aId)

### 7.230.1 Detailed Description

The TCP Socket class

### 7.230.2 Constructor & Destructor Documentation

#### 7.230.2.1 OSCL\_IMPORT\_REF OsclTCPSocket::~OsclTCPSocket ()

Destructor. The object must be deleted using the same allocator used in the NewL call.

### 7.230.3 Member Function Documentation

#### 7.230.3.1 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Accept (int32 aTimeout = -1)

Accept incoming connections. This is an asynchronous method.

**Parameters:**

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

#### 7.230.3.2 OSCL\_IMPORT\_REF int32 OsclTCPSocket::Bind (OsclNetworkAddress & aAddress)

Bind a TCP socket to an address. This is a synchronous method.

**Parameters:**

*aAddress*: Bind address.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code.

#### 7.230.3.3 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::BindAsync (OsclNetworkAddress & aAddress, int32 aTimeoutMsec = (-1))

Bind a TCP socket to an address. This is an asynchronous method.

**Parameters:**

*aAddress*: Bind address.

*aTimeoutMsec*: Optional timeout. Use a negative value for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**7.230.3.4 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelAccept ()**

Cancel Accept

This method will cancel any pending Accept operation on the current socket, causing the Accept to complete with error EPVSocketCancel. If there is no pending Accept operation, this method will have no effect.

**7.230.3.5 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelBind ()**

Cancel Bind

This method will cancel any pending BindAsync operation on the current socket, causing the BindAsync to complete with error EPVSocketCancel. If there is no pending BindAsync operation, this method will have no effect.

**7.230.3.6 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelConnect ()**

Cancel Connect

This method will cancel any pending Connect operation on the current socket, causing the Connect to complete with error EPVSocketCancel. If there is no pending Connect operation, this method will have no effect.

**7.230.3.7 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelListen ()**

Cancel Async Listen

This method will cancel any pending ListenAsync operation on the current socket, causing the Listen to complete with error EPVSocketCancel. If there is no pending Listen operation, this method will have no effect.

**7.230.3.8 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelRecv ()**

Cancel Recv

This method will cancel any pending Recv operation on the current socket, causing the Recv to complete with error EPVSocketCancel. If there is no pending Recv operation, this method will have no effect.

**7.230.3.9 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelSend ()**

Cancel Send

This method will cancel any pending Send operation on the current socket, causing the Send to complete with error EPVSocketCancel. If there is no pending Send operation, this method will have no effect.

**7.230.3.10 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelShutdown ()**

Cancel Shutdown

This method will cancel any pending Shutdown operation on the current socket, causing the Shutdown to complete with error EPVSocketCancel. If there is no pending Shutdown operation, this method will have no effect.

**7.230.3.11 OSCL\_IMPORT\_REF int32 OsclTCPSocket::Close ()**

Close a TCP socket. This is a synchronous method.

Once it is closed a socket cannot be re-opened. Sockets are automatically closed when they are deleted. This method may be used to see any error code returned from the platform's socket close call.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code.

**7.230.3.12 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Connect  
(OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1)**

Connect to an address. This is an asynchronous method.

**Parameters:**

*aAddress*: a network address.

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**7.230.3.13 OSCL\_IMPORT\_REF OsclTCPSocket\* OsclTCPSocket::GetAcceptedSocketL (uint32 aId)**

Retrieve the accept socket after a successful Accept operation. This is a synchronous method.

**Parameters:**

*aId*: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

**Returns:**

Returns pointer to socket, or NULL if error. Note: The caller is responsible for deleting any accepted socket that it retrieves.

**7.230.3.14 OSCL\_IMPORT\_REF int32 OsclTCPSocket::GetPeerName (OsclNetworkAddress & aPeerName)**

Retrieves the peer address of the socket

**Parameters:**

*aPeerName*: This will store the peer address when API returns successfully.

**Returns:**

Returns OsclErrNone 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

**7.230.3.15 OSCL\_IMPORT\_REF uint8\* OsclTCPSocket::GetRecvData (int32 \* aLength)**

Retrieve the received data after a successful Recv operation. This is a synchronous method.

**Parameters:**

*aLength*: (output) number of bytes of data received.

**Returns:**

Returns pointer to received data, or NULL if none.

**7.230.3.16 OSCL\_IMPORT\_REF uint8\* OsclTCPSocket::GetSendData (int32 \* aLength)**

Retrieve the sent data after a successful Send operation. This is a synchronous method.

**Parameters:**

*aLength*: (output) number of bytes of data sent.

**Returns:**

Returns pointer to sent data, or NULL if none.

**7.230.3.17 OSCL\_IMPORT\_REF int32 OsclTCPSocket::Listen (int32 aQueueSize)**

Listen. This is a synchronous method.

**Parameters:**

*aQueueSize*: Queue size.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code.

**7.230.3.18 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::ListenAsync (int32 aQueueSize, int32 aTimeoutMsec = (-1))**

ListenAsync This is an asynchronous method.

**Parameters:**

*aQueueSize*: Queue size.

*aTimeoutMsec*: Optional timeout. Use a negative value for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**7.230.3.19 OSCL\_IMPORT\_REF OsclTCPSocket\* OsclTCPSocket::NewL (Oscl\_DefAlloc & alloc, OsclSocketServ & aServ, OsclSocketObserver \* aObserver, uint32 aId) [static]**

Create a TCP Socket. May leave if failure.

**Parameters:**

*alloc*: Memory allocator.

*aServ*: Socket server. Must be connected.

*aObserver*: Socket observer.

*aId*: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

**Returns:**

Returns pointer to socket.

**7.230.3.20 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Recv (uint8 \* aPtr, uint32 aMaxLen, int32 aTimeoutMsec = -1)**

Receive Data. This is an asynchronous method.

**Parameters:**

*aPtr*: Buffer for received data.

*aMaxLen*: Length of buffer.

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**7.230.3.21 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Send (const uint8 \* aPtr, uint32 aLen, int32 aTimeoutMsec = -1)**

Send Data. This is an asynchronous method.

**Parameters:**

*aPtr*: Data to send.

*aLen*: Length of data to send.

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**7.230.3.22 OSCL\_IMPORT\_REF int32 OsclTCPSocket::SetOptionToReuseAddress ()**

Allows the server to bind to an address which is in a TIME\_WAIT state.

**Returns:**

Returns: OsclErrNone 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

**7.230.3.23 OSCL\_IMPORT\_REF int32 OsclTCPSocket::SetTOS (const OsclSocketTOS & aTOS)**

Sets the Type of Service field of each outgoing IP datagram.

**Parameters:**

*aTOS*: Specifies the type of service requested.

**Returns:**

Returns: OsclErrNone 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

**7.230.3.24 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Shutdown  
(TPVSocketShutdown aHow, int32 aTimeoutMsec = -1)**

Shutdown a socket. This is an asynchronous method.

**Parameters:**

*aHow*: type of shutdown

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**7.230.3.25 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::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;

**7.230.3.26 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::ThreadLogon  
(OsclSocketServ & aServ, OsclSocketObserver \* 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.

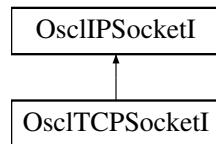
The documentation for this class was generated from the following file:

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

## 7.231 OsclTCPSocketI Class Reference

```
#include <oscl_tcp_socket.h>
```

Inheritance diagram for OsclTCPSocketI::



### Public Methods

- virtual ~OsclTCPSocketI ()
- [TPVSocketEvent ThreadLogoff \(\)](#)
- [TPVSocketEvent ThreadLogon \(OsclSocketServI \\*aServ, OsclSocketObserver \\*aObserver\)](#)
- int32 [Close \(\)](#)
- int32 [Listen \(int aQueueSize\)](#)
- OsclTCPSocketI \* [GetAcceptedSocketL \(uint32 aId\)](#)
- uint8 \* [GetRecvData \(int32 \\*aLength\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)
- [TPVSocketEvent BindAsync \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelBind \(\)](#)
- [TPVSocketEvent ListenAsync \(uint32 qsize, int32 aTimeoutMsec=-1\)](#)
- void [CancelListen \(\)](#)
- [TPVSocketEvent Connect \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelConnect \(\)](#)
- [TPVSocketEvent Shutdown \(TPVSocketShutdown aHow, int32 aTimeoutMsec=-1\)](#)
- void [CancelShutdown \(\)](#)
- [TPVSocketEvent Accept \(int32 aTimeout=-1\)](#)
- void [CancelAccept \(\)](#)
- [TPVSocketEvent Send \(const uint8 \\*&aPtr, uint32 aLen, int32 aTimeoutMsec=-1\)](#)
- void [CancelSend \(\)](#)
- [TPVSocketEvent Recv \(uint8 \\*&aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1\)](#)
- void [CancelRecv \(\)](#)

### Static Public Methods

- OsclTCPSocketI \* [NewL \(Oscl\\_DefAlloc &a, OsclSocketServI \\*aServ, OsclSocketObserver \\*aObserver, uint32 aId\)](#)

#### 7.231.1 Detailed Description

Internal implementation class for [OsclTCPSocket](#)

## 7.231.2 Constructor & Destructor Documentation

7.231.2.1 **virtual OsclTCPSocketI::~OsclTCPSocketI () [virtual]**

## 7.231.3 Member Function Documentation

7.231.3.1 **TPVSocketEvent OsclTCPSocketI::Accept (int32 *aTimeout* = -1) [inline]**

7.231.3.2 **TPVSocketEvent OsclTCPSocketI::BindAsync (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1) [inline]**

7.231.3.3 **void OsclTCPSocketI::CancelAccept () [inline]**

7.231.3.4 **void OsclTCPSocketI::CancelBind () [inline]**

7.231.3.5 **void OsclTCPSocketI::CancelConnect () [inline]**

7.231.3.6 **void OsclTCPSocketI::CancelListen () [inline]**

7.231.3.7 **void OsclTCPSocketI::CancelRecv () [inline]**

7.231.3.8 **void OsclTCPSocketI::CancelSend () [inline]**

7.231.3.9 **void OsclTCPSocketI::CancelShutdown () [inline]**

7.231.3.10 **int32 OsclTCPSocketI::Close () [virtual]**

Implements [OsclIPSocketI](#).

7.231.3.11 **TPVSocketEvent OsclTCPSocketI::Connect (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1) [inline]**

7.231.3.12 **OsclTCPSocketI\* OsclTCPSocketI::GetAcceptedSocketL (uint32 *aId*)**

7.231.3.13 **uint8 \* OsclTCPSocketI::GetRecvData (int32 \* *aLength*) [inline, virtual]**

Implements [OsclIPSocketI](#).

7.231.3.14 **uint8 \* OsclTCPSocketI::GetSendData (int32 \* *aLength*) [inline, virtual]**

Implements [OsclIPSocketI](#).

- 7.231.3.15 **int32** OsclTCPSocketI::Listen (*int aQueueSize*) [inline]
- 7.231.3.16 **TPVSocketEvent** OsclTCPSocketI::ListenAsync (*uint32 qsize, int32 aTimeoutMsec = -1*) [inline]
- 7.231.3.17 OsclTCPSocketI\* OsclTCPSocketI::NewL (**Oscl\_DefAlloc** & *a*, **OsclSocketServI** \* *aServ*, **OsclSocketObserver** \* *aObserver*, *uint32 aId*) [static]
- 7.231.3.18 **TPVSocketEvent** OsclTCPSocketI::Recv (*uint8 \*& aPtr, uint32 aMaxLen, int32 aTimeoutMsec = -1*) [inline]
- 7.231.3.19 **TPVSocketEvent** OsclTCPSocketI::Send (*const uint8 \*& aPtr, uint32 aLen, int32 aTimeoutMsec = -1*) [inline]
- 7.231.3.20 **TPVSocketEvent** OsclTCPSocketI::Shutdown (**TPVSocketShutdown** *aHow, int32 aTimeoutMsec = -1*) [inline]
- 7.231.3.21 **TPVSocketEvent** OsclTCPSocketI::ThreadLogoff ()

Reimplemented from **OsclIPSocketI**.

- 7.231.3.22 **TPVSocketEvent** OsclTCPSocketI::ThreadLogon (**OsclSocketServI** \* *aServ*, **OsclSocketObserver** \* *aObserver*)

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

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

## 7.232 OsclThread Class Reference

```
#include <oscl_thread.h>
```

### Public Methods

- OSCL\_IMPORT\_REF OsclThread ()
- OSCL\_IMPORT\_REF ~OsclThread ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Create (TOsclThreadFuncPtr func, int32 stack\_size, TOsclThreadFuncArg argument, OsclThread\_State state=Start\_on\_creation, bool oIsJoinable=false)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError GetPriority (OsclThreadPriority &refThreadPriority)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError SetPriority (OsclThreadPriority ePriority)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Suspend ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Resume ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Terminate (OsclAny \*exitcode)
- OSCL\_IMPORT\_REF TOsclThreadTerminate CanTerminate ()

### Static Public Methods

- OSCL\_IMPORT\_REF void Exit (OsclAny \*exitcode)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError GetId (TOsclThreadId &refThreadId)
- OSCL\_IMPORT\_REF bool CompareId (TOsclThreadId &t1, TOsclThreadId &t2)
- OSCL\_IMPORT\_REF void SleepMillisec (const int32 msec)

### 7.232.1 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.

### 7.232.2 Constructor & Destructor Documentation

#### 7.232.2.1 OSCL\_IMPORT\_REF OsclThread::OsclThread ()

Class constructor

#### 7.232.2.2 OSCL\_IMPORT\_REF OsclThread::~OsclThread ()

Class destructor

### 7.232.3 Member Function Documentation

#### 7.232.3.1 OSCL\_IMPORT\_REF TOsclThreadTerminate OsclThread::CanTerminate ()

Tell if thread terminate will do join, immediate hard kill, or NOP.

**Returns:**

Terminate behavior.

**7.232.3.2 OSCL\_IMPORT\_REF bool OsclThread::CompareId ([TOsclThreadId & t1](#), [TOsclThreadId & t2](#)) [static]**

Static routine to compare whether two thread ID's are equal.

**Parameters:**

*t1, t2*: thread ID passed by the application

**Returns:**

true if equal.

**7.232.3.3 OSCL\_IMPORT\_REF [OsclProcStatus::eOsclProcError](#) OsclThread::Create ([TOsclThreadFuncPtr func](#), [int32 stack\\_size](#), [TOsclThreadFuncArg argument](#), [OsclThread\\_State state = Start\\_on\\_creation](#), [bool oIsJoinable = false](#))**

This routine will create a thread. The thread may be launched immediately or may be created in a suspended state and launched with a Resume call.

**Parameters:**

*func* = Name of the thread Function  
*stack\_size* = Size of the thread stack. If zero, then the platform-specific default stack size will be used.  
*argument* = Argument to be passed to thread function  
*state* = Enumeration which specifies the state of the thread on creation with values Running and Suspend. Note: the Suspend option may not be available on all platforms. If it is not supported, the Create call will return INVALID\_PARAM\_ERROR.  
*oIsJoinable* = A boolean, which when set to true, creates a Joinable thread. The default value for this is false, which creates a Detached thread.  
Note 1: When a joinable thread is created, it is imperative to call thread Terminate. Otherwise, it would cause a memory leak.  
Note 2: This is currently available only for platforms that have support for pthreads.

**Returns:**

[eOsclProcError](#)

**7.232.3.4 OSCL\_IMPORT\_REF void OsclThread::Exit ([OsclAny \\* exitcode](#)) [static]**

Exit is a static function which is used to end the current thread. When called it just ends the execution of the current thread. Note: on some platforms this may be a NOP.

**Parameters:**

*exitcode* = Exitcode of the thread. This can be used by other threads to know the exit status of this thread.

**Returns:**

None

**7.232.3.5 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::GetId  
(TOsclThreadId & refThreadId) [static]**

Static routine to retrieve ID of calling thread.

**Parameters:**

*Thread* ID passed by the application

**Returns:**

Error code

**7.232.3.6 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::GetPriority  
(OsclThreadPriority & refThreadPriority)**

GetThreadPriority gets the priority of the thread. It takes reference of the input argument and assigns priority to it from one of the already defined priorities.

**Parameters:**

*int16&* refThreadPriority : Output Priority value

**Returns:**

Error code

**7.232.3.7 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Resume ()**

ResumeThread resumes the suspended thread and brings it into execution.

**Parameters:**

*None*

**Returns:**

Error code Note: this function may not be supported on all platforms, and may return NOT\_ - IMPLEMENTED.

**7.232.3.8 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::SetPriority  
(OsclThreadPriority ePriority)**

SetThreadPriority sets the priority of the thread. It takes priority as the input argument and assigns it to the thread referred.

**Parameters:**

*ePriorityLevel* : Input Priority value

**Returns:**

Error code Note: this function may not be supported on all platforms, and may return NOT\_ - IMPLEMENTED.

**7.232.3.9 OSCL\_IMPORT\_REF void OsclThread::SleepMillisec (const int32 msec) [static]**

Suspend current thread execution for specified time.

**Parameters:**

*msec*, *t2*: sleep time in milliseconds.

**7.232.3.10 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Suspend ()**

This API suspends the thread being referred. The thread can later be brought into execution by calling OSCL\_ResumeThread() on it.

**Parameters:**

*None*

**Returns:**

Error code Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.232.3.11 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Terminate  
(OsclAny \* *exitcode*)**

Terminate a thread other than the calling thread.

This API may have multiple behaviors. It may do a hard kill, a "join" operation, or a do-nothing. Caller can use CanTerminate option to tell the behavior in advance.

**Parameters:**

*exitcode* = Exitcode of the thread.

**Returns:**

Error code

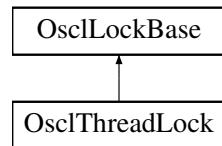
The documentation for this class was generated from the following file:

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

## 7.233 OsclThreadLock Class Reference

```
#include <oscl_mutex.h>
```

Inheritance diagram for OsclThreadLock::



### Public Methods

- OSCL\_IMPORT\_REF OsclThreadLock ()
- virtual OSCL\_IMPORT\_REF ~OsclThreadLock ()
- OSCL\_IMPORT\_REF void Lock ()
- OSCL\_IMPORT\_REF void Unlock ()

#### 7.233.1 Detailed Description

An implementation of [OsclLockBase](#) using a mutex

#### 7.233.2 Constructor & Destructor Documentation

##### 7.233.2.1 OSCL\_IMPORT\_REF OsclThreadLock::OsclThreadLock ()

7.233.2.2 virtual OSCL\_IMPORT\_REF OsclThreadLock::~OsclThreadLock () [virtual]

#### 7.233.3 Member Function Documentation

##### 7.233.3.1 OSCL\_IMPORT\_REF void OsclThreadLock::Lock () [virtual]

Implements [OsclLockBase](#).

##### 7.233.3.2 OSCL\_IMPORT\_REF void OsclThreadLock::Unlock () [virtual]

Implements [OsclLockBase](#).

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

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

## 7.234 OsclTickCount Class Reference

```
#include <oscl_tickcount.h>
```

### Static Public Methods

- uint32 [TickCount \(\)](#)
- uint32 [TickCountFrequency \(\)](#)
- uint32 [TickCountPeriod \(\)](#)
- uint32 [TicksToMsec \(uint32 ticks\)](#)
- uint32 [MsecToTicks \(uint32 msec\)](#)

#### 7.234.1 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.

#### 7.234.2 Member Function Documentation

##### 7.234.2.1 uint32 OsclTickCount::MsecToTicks (uint32 *msec*) [static]

This function converts milliseconds to ticks

**Returns:**

ticks

##### 7.234.2.2 uint32 OsclTickCount::TickCount () [static]

This function returns the current system tick count

**Returns:**

returns the tick count

##### 7.234.2.3 uint32 OsclTickCount::TickCountFrequency () [static]

This function returns the tick frequency in ticks per second

**Returns:**

ticks per second

##### 7.234.2.4 uint32 OsclTickCount::TickCountPeriod () [static]

This function returns the tick period in microseconds per tick

**Returns:**

microseconds per tick

**7.234.2.5 uint32 OsclTickCount::TicksToMsec (uint32 *ticks*) [static]**

This function converts ticks to milliseconds

**Returns:**

milliseconds

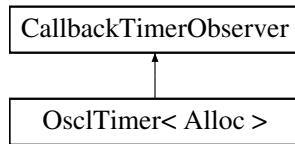
The documentation for this class was generated from the following file:

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

## 7.235 OsclTimer< Alloc > Class Template Reference

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

Inheritance diagram for OsclTimer< Alloc >::



### Public Types

- `typedef CallbackTimer< Alloc > callback_timer_type`

### Public Methods

- `OsclTimer (const char *name, uint32 frequency=1, int32 priority=OsclActiveObject::EPriorityNominal)`
- `virtual ~OsclTimer ()`
- `void SetObserver (OsclTimerObserver *obs)`
- `void SetFrequency (uint32 frequency)`
- `void SetExactFrequency (uint32 frequency)`
- `void Request (int32 timerID, int32 timeoutInfo, int32 cycles, OsclTimerObserver *obs=0, bool recurring=0)`
- `void Cancel (int32 timerID, int32 timeoutInfo=-1)`
- `void Clear ()`

### Protected Methods

- `void TimerBaseElapsed ()`

### Friends

- `class CallbackTimer< Alloc >`

template<class Alloc> class OsclTimer< Alloc >

### 7.235.1 Member Typedef Documentation

7.235.1.1 template<class Alloc> typedef CallbackTimer<Alloc> OsclTimer< Alloc >::callback\_timer\_type

### 7.235.2 Constructor & Destructor Documentation

7.235.2.1 template<class Alloc> OsclTimer< Alloc >::OsclTimer (const char \* *name*, uint32 *frequency* = 1, int32 *priority* = OsclActiveObject::EPriorityNominal)

Constructor

**Parameters:**

*frequency* The frequency of the timer in cycles/second. A value of 1 means the timer will cycle in 1 second intervals.

7.235.2.2 template<class Alloc> OsclTimer< Alloc >::~OsclTimer () [virtual]

### 7.235.3 Member Function Documentation

7.235.3.1 template<class Alloc> void OsclTimer< Alloc >::Cancel (int32 *timerID*, int32 *timeoutInfo* = -1)

Cancel a timer

**Parameters:**

*timerID* used to identify the timer to cancel.

*timeoutInfo* if not set to -1, this value will be used as additional matching criteria to cancel a timer.

7.235.3.2 template<class Alloc> void OsclTimer< Alloc >::Clear ()

Cancel all pending timers.

7.235.3.3 template<class Alloc> void OsclTimer< Alloc >::Request (int32 *timerID*, int32 *timeoutInfo*, int32 *cycles*, OsclTimerObserver \* *obs* = 0, bool *recurring* = 0)

Request a timer

**Parameters:**

*timerID* used to identify the timer for cancellation. This value will be returned as part of the timeout event.

*timeoutInfo* for user info. Returned to the observer on a timeout event

*cycles* the number of cycles to wait before a timeout event. If the timer frequency is 1 and the cycles are set to 2, then the timeout event will occur in 2 seconds.

*obs* a local observer object to be called on a timeout event. This observer overrides the global observer if set.

**7.235.3.4 template<class Alloc> void OsclTimer< Alloc >::SetExactFrequency (uint32 *frequency*)**

Set the exact frequency of the timer in microsecond.

**Parameters:**

*frequency* A value of 1 means the timer will cycle in one microsecond intervals, 1000 means millisecond intervals, etc.

**7.235.3.5 template<class Alloc> void OsclTimer< Alloc >::SetFrequency (uint32 *frequency*)**

Set the frequency of the timer in cycles/second.

**Parameters:**

*frequency* A value of 1 means the timer will cycle in one second intervals, 1000 means millisecond intervals, etc.

**7.235.3.6 template<class Alloc> void OsclTimer< Alloc >::SetObserver ([OsclTimerObserver](#) \* *obs*) [inline]**

Set the global observer. Each timer can request a local observer, which if set overrides the global observer.

**Parameters:**

*obs* observer object.

**7.235.3.7 template<class Alloc> void OsclTimer< Alloc >::TimerBaseElapsed () [protected, virtual]**

Implements [CallbackTimerObserver](#).

## 7.235.4 Friends And Related Function Documentation

**7.235.4.1 template<class Alloc> friend class [CallbackTimer](#)< Alloc > [friend]**

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

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

## 7.236 OsclTimerCompare Class Reference

```
#include <oscl_scheduler_readyq.h>
```

### Static Public Methods

- int [compare \(TOsclReady &a, TOsclReady &b\)](#)

#### 7.236.1 Member Function Documentation

##### 7.236.1.1 int OsclTimerCompare::compare (TOsclReady &*a*, TOsclReady &*b*) [static]

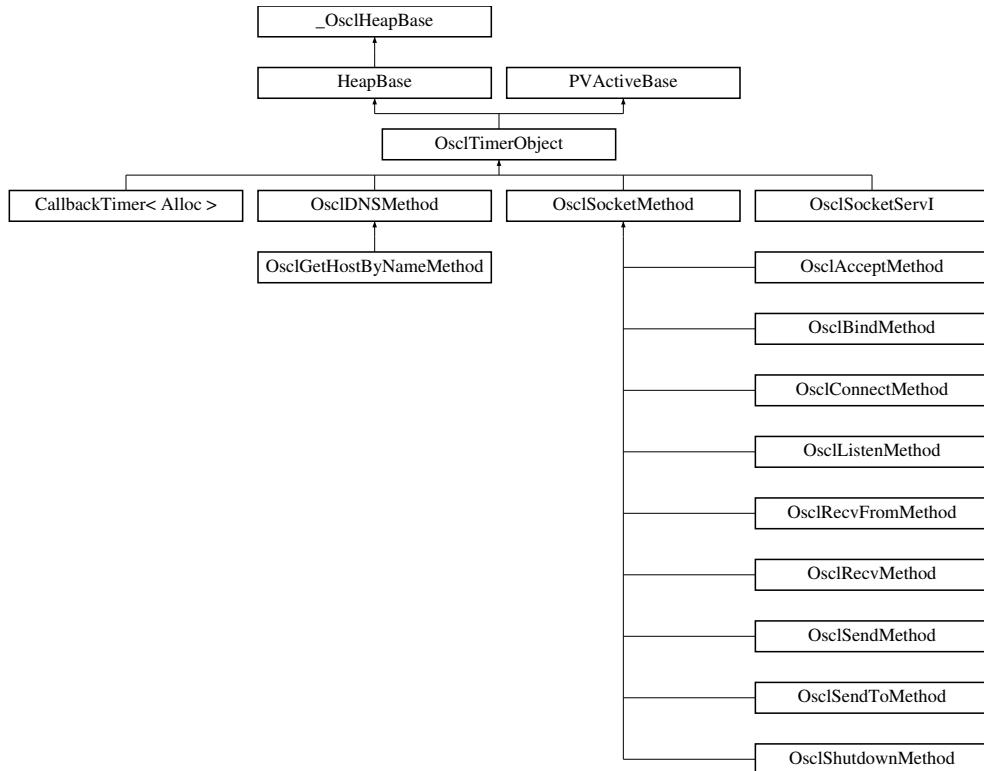
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 7.237 OsclTimerObject Class Reference

```
#include <oscl_scheduler_ao.h>
```

Inheritance diagram for OsclTimerObject::



### Public Methods

- OSCL\_IMPORT\_REF [OsclTimerObject](#) (int32 aPriority, const char name[ ])
- virtual OSCL\_IMPORT\_REF [~OsclTimerObject](#) ()
- OSCL\_IMPORT\_REF void [AddToScheduler](#) ()
- OSCL\_IMPORT\_REF void [RemoveFromScheduler](#) ()
- OSCL\_IMPORT\_REF void [After](#) (int32 aDelayMicrosec)
- OSCL\_IMPORT\_REF void [RunIfNotReady](#) (uint32 aDelayMicrosec=0)
- OSCL\_IMPORT\_REF void [SetBusy](#) ()
- OSCL\_IMPORT\_REF bool [IsBusy](#) () const
- OSCL\_IMPORT\_REF void [Cancel](#) ()
- OSCL\_IMPORT\_REF int32 [Priority](#) () const
- OSCL\_IMPORT\_REF int32 [Status](#) () const
- OSCL\_IMPORT\_REF void [SetStatus](#) (int32)
- OSCL\_IMPORT\_REF [OsclAOStatus](#) & [StatusRef](#) ()

### Protected Methods

- virtual OSCL\_IMPORT\_REF void [DoCancel](#) ()
- virtual OSCL\_IMPORT\_REF int32 [RunError](#) (int32 aError)

### 7.237.1 Detailed Description

User base class for execution objects. OsclTimerObject defines an exec object with a timer.

### 7.237.2 Constructor & Destructor Documentation

#### 7.237.2.1 OSCL\_IMPORT\_REF OsclTimerObject::OsclTimerObject (int32 *aPriority*, const char *name*[ ])

Constructor.

**Parameters:**

*aPriority* (input param): scheduling priority

*name* (input param): optional name for this AO.

#### 7.237.2.2 virtual OSCL\_IMPORT\_REF OsclTimerObject::~OsclTimerObject () [virtual]

Destructor.

### 7.237.3 Member Function Documentation

#### 7.237.3.1 OSCL\_IMPORT\_REF void OsclTimerObject::AddToScheduler ()

Add this AO to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

#### 7.237.3.2 OSCL\_IMPORT\_REF void OsclTimerObject::After (int32 *aDelayMicrosec*)

'After' sets the request ready, with request status OSCL\_REQUEST\_STATUS\_PENDING, and starts a timer. When the timer expires, the request will complete with status OSCL\_REQUEST\_ERR\_NONE. Must be called from the same thread in which the active object is scheduled. Will leave if the request is already readied, the object is not added to any scheduler, or the calling thread does not match the scheduling thread.

**Parameters:**

*anInterval*: timeout interval in microseconds.

#### 7.237.3.3 OSCL\_IMPORT\_REF void OsclTimerObject::Cancel ()

Cancel any active request. If the request is pending, this will call the DoCancel routine, wait for the request to cancel, then set the request idle. The AO will not run. If the request is not active, it does nothing. Request must be canceled from the same thread in which it is scheduled.

Reimplemented from [PVActiveBase](#).

**7.237.3.4 virtual OSCL\_IMPORT\_REF void OsclTimerObject::DoCancel () [protected, virtual]**

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will cancel the timer. If any additional action is needed, the derived class may override this. If the derived class does override this, it should explicitly call [OsclTimerObject::DoCancel](#) in its own DoCancel routine.

Implements [PVActiveBase](#).

**7.237.3.5 OSCL\_IMPORT\_REF bool OsclTimerObject::IsBusy ()**

Return true if this AO is active, false otherwise.

**7.237.3.6 OSCL\_IMPORT\_REF int32 OsclTimerObject::Priority ()**

Return scheduling priority of this exec object.

**7.237.3.7 OSCL\_IMPORT\_REF void OsclTimerObject::RemoveFromScheduler ()**

Remove this AO from its scheduler. Will leave if the calling thread context does not match the scheduling thread. Cancels any pending request before removing.

Reimplemented from [PVActiveBase](#).

**7.237.3.8 virtual OSCL\_IMPORT\_REF int32 OsclTimerObject::RunError (int32 *aError*) [protected, virtual]**

Run Leave handler. This gets called by scheduler when the Run routine leaves. The default implementation simply returns the leave code. If the derived class wants to handle errors from Run, it may override this. The ExecError should return OsclErrNone if it handles the error, otherwise it should return the input error code.

**Parameters:**

*aError*: the leave code generated by the Run.

Implements [PVActiveBase](#).

**7.237.3.9 OSCL\_IMPORT\_REF void OsclTimerObject::RunIfNotReady (uint32 *aDelayMicrosec* = 0)**

Complete the request after a time interval. RunIfNotReady is identical to [After\(\)](#) except that it first checks the request status, and if it is already readied, it does nothing.

**Parameters:**

*aDelayMicrosec* (input param): delay in microseconds.

**7.237.3.10 OSCL\_IMPORT\_REF void OsclTimerObject::SetBusy ()**

Set request ready for this AO. Will leave if the request is already readied, or the exec object is not added to any scheduler, or the calling thread context does not match the scheduler thread.

**7.237.3.11 OSCL\_IMPORT\_REF void OsclTimerObject::SetStatus (int32)**

**7.237.3.12 OSCL\_IMPORT\_REF int32 OsclTimerObject::Status ()**

Request status access

**7.237.3.13 OSCL\_IMPORT\_REF OsclAOStatus& OsclTimerObject::StatusRef ()**

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

- [oscl\\_scheduler\\_ao.h](#)

## 7.238 OsclTimerObserver Class Reference

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

### Public Methods

- virtual void [TimeoutOccurred](#) (int32 timerID, int32 timeoutInfo)=0
- virtual [~OsclTimerObserver](#) ()

#### 7.238.1 Detailed Description

The observer class to receive timeout callbacks

#### 7.238.2 Constructor & Destructor Documentation

**7.238.2.1** virtual OsclTimerObserver::[~OsclTimerObserver](#) () [inline, virtual]

#### 7.238.3 Member Function Documentation

**7.238.3.1** virtual void OsclTimerObserver::[TimeoutOccurred](#) (int32 *timerID*, int32 *timeoutInfo*) [pure virtual]

This function will be called when the timer associated with this observer is executed

##### Parameters:

*timerID* The ID given at timer request.

*timeoutInfo* Any extra info given at timer request.

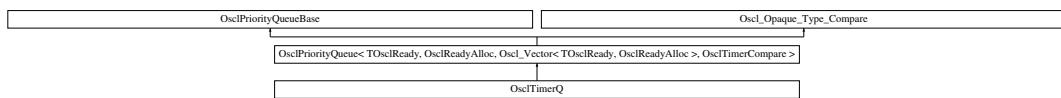
The documentation for this class was generated from the following file:

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

## 7.239 OsclTimerQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclTimerQ::



### Public Methods

- void [Construct](#) (int)
- void [Add](#) ([TOsclReady](#))
- void [Remove](#) ([TOsclReady](#))
- [TOsclReady PopTop](#) ()
- [TOsclReady Top](#) ()
- void [Pop](#) ([TOsclReady](#))
- bool [IsIn](#) ([TOsclReady](#))

#### 7.239.1 Member Function Documentation

**7.239.1.1 void OsclTimerQ::Add ([TOsclReady](#))**

**7.239.1.2 void OsclTimerQ::Construct (int)**

**7.239.1.3 bool OsclTimerQ::IsIn ([TOsclReady](#))**

**7.239.1.4 void OsclTimerQ::Pop ([TOsclReady](#))**

**7.239.1.5 [TOsclReady](#) OsclTimerQ::PopTop ()**

**7.239.1.6 void OsclTimerQ::Remove ([TOsclReady](#))**

**7.239.1.7 [TOsclReady](#) OsclTimerQ::Top ()**

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

- [oscl\\_scheduler\\_readyq.h](#)

## 7.240 OsclTLS< T, ID, Registry > Class Template Reference

```
#include <oscl_tls.h>
```

### Public Methods

- `OsclTLS ()`
- `~OsclTLS ()`
- `T & operator * () const`

*The indirection operator (\*) accesses a value indirectly, through a pointer.*

- `T * operator -> () const`

*The indirection operator (->) accesses a value indirectly, through a pointer.*

- `bool set ()`

*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- `T * _Ptr`

```
template<class T, uint32 ID, class Registry = OsclTLSRegistry> class OsclTLS< T, ID, Registry >
```

#### 7.240.1 Constructor & Destructor Documentation

**7.240.1.1 template<class T, uint32 ID, class Registry = OsclTLSRegistry> OsclTLS< T, ID, Registry >::OsclTLS () [inline]**

**7.240.1.2 template<class T, uint32 ID, class Registry = OsclTLSRegistry> OsclTLS< T, ID, Registry >::~OsclTLS () [inline]**

#### 7.240.2 Member Function Documentation

**7.240.2.1 template<class T, uint32 ID, class Registry = OsclTLSRegistry> T& OsclTLS< T, ID, Registry >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclTLS can be used like the regular pointer that it was initialized with.

**7.240.2.2 template<class T, uint32 ID, class Registry = OsclTLSRegistry> T\* OsclTLS< T, ID, Registry >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclTLS can be used like the regular pointer that it was initialized with.

**7.240.2.3 template<class T, uint32 ID, class Registry = OsclTLSRegistry> bool OsclTLS< T, ID, Registry >::set () [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

**7.240.3 Field Documentation****7.240.3.1 template<class T, uint32 ID, class Registry = OsclTLSRegistry> T\* OsclTLS< T, ID, Registry >::\_Ptr [protected]**

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

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

## 7.241 OsclTLSEEx< T, ID, Registry > Class Template Reference

```
#include <oscl_error.h>
```

### Public Methods

- `OsclTLSEEx ()`
- `~OsclTLSEEx ()`
- `T & operator * () const`

*The indirection operator (\*) accesses a value indirectly, through a pointer.*

- `T * operator -> () const`

*The indirection operator (->) accesses a value indirectly, through a pointer.*

- `bool set ()`

*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- `T * _Ptr`

```
template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> class OsclTLSEEx< T, ID, Registry >
```

#### 7.241.1 Constructor & Destructor Documentation

**7.241.1.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> OsclTLSEEx< T, ID, Registry >::OsclTLSEEx () [inline]**

**7.241.1.2 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> OsclTLSEEx< T, ID, Registry >::~OsclTLSEEx () [inline]**

#### 7.241.2 Member Function Documentation

**7.241.2.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T& OsclTLSEEx< T, ID, Registry >::operator \* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the `OsclTLS` can be used like the regular pointer that it was initialized with.

**7.241.2.2 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T\* OsclTLSEEx< T, ID, Registry >::operator -> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the `OsclTLS` can be used like the regular pointer that it was initialized with.

### 7.241.2.3 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> bool OsclTLSE< 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 = OsclTLSRegistryEx> T\* OsclTLSE< T, ID, Registry >::\_Ptr [protected]

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

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

## 7.242 OsclTLSRegistry Class Reference

```
#include <oscl_tls.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF [OsclAny](#) \* getInstance (uint32 ID, int32 &error)
- OSCL\_IMPORT\_REF void [registerInstance](#) ([OsclAny](#) \*ptr, uint32 ID, int32 &error)

### Friends

- class [OsclBase](#)

#### 7.242.1 Member Function Documentation

**7.242.1.1 OSCL\_IMPORT\_REF [OsclAny](#)\* OsclTLSRegistry::getInstance (uint32 *ID*, int32 & *error*) [static]**

**7.242.1.2 OSCL\_IMPORT\_REF void OsclTLSRegistry::registerInstance ([OsclAny](#) \* *ptr*, uint32 *ID*, int32 & *error*) [static]**

#### 7.242.2 Friends And Related Function Documentation

**7.242.2.1 friend class [OsclBase](#) [friend]**

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

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

## 7.243 OsclTLSRegistryEx Class Reference

```
#include <oscl_error.h>
```

### Static Public Methods

- [OsclAny \\* getInstance \(uint32 ID\)](#)
- [void registerInstance \(OsclAny \\*ptr, uint32 ID\)](#)

#### 7.243.1 Member Function Documentation

**7.243.1.1 [OsclAny\\* OsclTLSRegistryEx::getInstance \(uint32 ID\)](#) [inline, static]**

**7.243.1.2 [void OsclTLSRegistryEx::registerInstance \(OsclAny \\*ptr, uint32 ID\)](#) [inline, static]**

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

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

## 7.244 OsclTrapItem Class Reference

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

### Public Methods

- OSCL\_INLINE [OsclTrapItem \(OsclTrapOperation anOperation\)](#)
- OSCL\_INLINE [OsclTrapItem \(OsclTrapOperation anOperation, OsclAny \\*aPtr\)](#)

### Friends

- class [OsclTrapStackItem](#)
- class [OsclTrapStack](#)

#### 7.244.1 Constructor & Destructor Documentation

**7.244.1.1 OSCL\_INLINE OsclTrapItem::OsclTrapItem ([OsclTrapOperation anOperation](#))**

**7.244.1.2 OSCL\_INLINE OsclTrapItem::OsclTrapItem ([OsclTrapOperation anOperation, OsclAny \\* aPtr](#))**

#### 7.244.2 Friends And Related Function Documentation

**7.244.2.1 friend class OsclTrapStack [friend]**

**7.244.2.2 friend class OsclTrapStackItem [friend]**

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

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

## 7.245 OsclTrapStack Class Reference

```
#include <oscl_error_trapcleanup.h>
```

### Friends

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

### 7.245.1 Detailed Description

A common type for cleanup stack and trap mark stack. for internal use only.

### 7.245.2 Friends And Related Function Documentation

**7.245.2.1 friend class OsclError [friend]**

**7.245.2.2 friend class OsclErrorTrap [friend]**

**7.245.2.3 friend class OsclErrorTrapImp [friend]**

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

- [oscl\\_error\\_trapcleanup.h](#)

## 7.246 OsclTrapStackItem Class Reference

```
#include <oscl_error_trapcleanup.h>
```

### Public Methods

- [OsclTrapStackItem \(\)](#)
- [OsclTrapStackItem \(\\_OsclHeapBase \\*aCBase\)](#)
- [OsclTrapStackItem \(OsclAny \\*aTAny\)](#)
- [OsclTrapStackItem \(OsclTrapItem aItem\)](#)

### Data Fields

- [\\_OsclHeapBase \\* iCBase](#)
- [OsclAny \\* iTAny](#)
- [OsclTrapOperation iTrapOperation](#)
- [OsclTrapStackItem \\* iNext](#)

#### 7.246.1 Detailed Description

Internal cleanup stack item type.

#### 7.246.2 Constructor & Destructor Documentation

**7.246.2.1 OsclTrapStackItem::OsclTrapStackItem () [inline]**

**7.246.2.2 OsclTrapStackItem::OsclTrapStackItem (\_OsclHeapBase \* aCBase) [inline]**

**7.246.2.3 OsclTrapStackItem::OsclTrapStackItem (OsclAny \* aTAny) [inline]**

**7.246.2.4 OsclTrapStackItem::OsclTrapStackItem (OsclTrapItem aItem) [inline]**

#### 7.246.3 Field Documentation

**7.246.3.1 \_OsclHeapBase\* OsclTrapStackItem::iCBase**

**7.246.3.2 OsclTrapStackItem\* OsclTrapStackItem::iNext**

**7.246.3.3 OsclAny\* OsclTrapStackItem::iTAny**

**7.246.3.4 OsclTrapOperation OsclTrapStackItem::iTrapOperation**

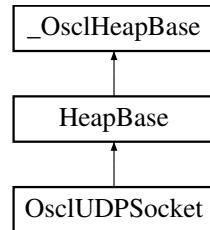
The documentation for this class was generated from the following file:

- [oscl\\_error\\_trapcleanup.h](#)

## 7.247 OsclUDPSocket Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclUDPSocket::



### Public Methods

- OSCL\_IMPORT\_REF ~OsclUDPSocket ()
- OSCL\_IMPORT\_REF TPVSocketEvent ThreadLogoff ()
- OSCL\_IMPORT\_REF TPVSocketEvent ThreadLogon (OsclSocketServ &aServ, OsclSocketObserver \*aObserver)
- OSCL\_IMPORT\_REF int32 Close ()
- OSCL\_IMPORT\_REF int32 Bind (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF int32 Join (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF int32 JoinMulticastGroup (OsclIpMReq &aMReq)
- OSCL\_IMPORT\_REF int32 SetMulticastTTL (int32 aTTL)
- OSCL\_IMPORT\_REF int32 SetOptionToReuseAddress ()
- OSCL\_IMPORT\_REF int32 SetTOS (const OsclSocketTOS &aTOS)
- OSCL\_IMPORT\_REF int32 GetPeerName (OsclNetworkAddress &aPeerName)
- OSCL\_IMPORT\_REF TPVSocketEvent BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void CancelBind ()
- OSCL\_IMPORT\_REF uint8 \* GetRecvData (int32 \*aLength)
- OSCL\_IMPORT\_REF uint8 \* GetSendData (int32 \*aLength)
- OSCL\_IMPORT\_REF TPVSocketEvent SendTo (const uint8 \*aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelSendTo ()
- OSCL\_IMPORT\_REF TPVSocketEvent RecvFrom (uint8 \*aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiRecvLimit=0, Oscl\_Vector< uint32, OsclMemAllocator > \*aPacketLen=NULL, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \*aPacketSource=NULL)
- OSCL\_IMPORT\_REF void CancelRecvFrom ()
- OSCL\_IMPORT\_REF int32 SetRecvBufferSize (uint32 size)

### Static Public Methods

- OSCL\_IMPORT\_REF OsclUDPSocket \* NewL (Oscl\_DefAlloc &alloc, OsclSocketServ &aServ, OsclSocketObserver \*aObserver, uint32 aId)

### 7.247.1 Detailed Description

The UDP Socket class

### 7.247.2 Constructor & Destructor Documentation

#### 7.247.2.1 OSCL\_IMPORT\_REF OsclUDPSocket::~OsclUDPSocket ()

Destructor. The object must be deleted using the same allocator used in the NewL call.

### 7.247.3 Member Function Documentation

#### 7.247.3.1 OSCL\_IMPORT\_REF int32 OsclUDPSocket::Bind ([OsclNetworkAddress](#) & *aAddress*)

Bind a UDP socket to an address. This is a synchronous method.

**Parameters:**

*aAddress*: Bind address.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code.

#### 7.247.3.2 OSCL\_IMPORT\_REF [TPVSocketEvent](#) OsclUDPSocket::BindAsync ([OsclNetworkAddress](#) & *aAddress*, int32 *aTimeoutMsec* = (-1))

Bind a UDP socket to an address. This is an asynchronous method.

**Parameters:**

*aAddress*: Bind address.

*aTimeoutMsec*: Optional timeout. Use a negative value for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

#### 7.247.3.3 OSCL\_IMPORT\_REF void OsclUDPSocket::CancelBind ()

Cancel Bind

This method will cancel any pending BindAsync operation on the current socket, causing the BindAsync to complete with error EPVSocketCancel. If there is no pending BindAsync operation, this method will have no effect.

#### 7.247.3.4 OSCL\_IMPORT\_REF void OsclUDPSocket::CancelRecvFrom ()

Cancel RecvFrom

This method will cancel any pending RecvFrom operation on the current socket, causing the RecvFrom to complete with error EPVSocketCancel. If there is no pending RecvFrom operation, this method will have no effect.

**7.247.3.5 OSCL\_IMPORT\_REF void OsclUDPSocket::CancelSendTo ()**

Cancel SendTo

This method will cancel any pending SendTo operation on the current socket, causing the SendTo to complete with error EPVSocketCancel. If there is no pending SendTo operation, this method will have no effect.

**7.247.3.6 OSCL\_IMPORT\_REF int32 OsclUDPSocket::Close ()**

Close a UDP socket. This is a synchronous method.

Once it is closed a socket cannot be re-opened. Sockets are automatically closed when they are deleted. This method may be used to see any error code returned from the platform's socket close call.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code.

**7.247.3.7 OSCL\_IMPORT\_REF int32 OsclUDPSocket::GetPeerName ([OsclNetworkAddress](#) & *aPeerName*)**

Retrieves the peer address of the socket

**Parameters:**

*aPeerName*: This will store the peer address when API returns successfully.

**Returns:**

Returns OsclErrNone 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

**7.247.3.8 OSCL\_IMPORT\_REF uint8\* OsclUDPSocket::GetRecvData (int32 \* *aLength*)**

Retrieve the received data after a successful RecvFrom operation. This is a synchronous method.

**Parameters:**

*aLength*: (output) number of bytes of data received.

**Returns:**

Returns pointer to received data, or NULL if none.

**7.247.3.9 OSCL\_IMPORT\_REF uint8\* OsclUDPSocket::GetSendData (int32 \* *aLength*)**

Retrieve the sent data after a successful SendTo operation. This is a synchronous method.

**Parameters:**

*aLength*: (output) number of bytes of data sent.

**Returns:**

Returns pointer to sent data, or NULL if none.

### 7.247.3.10 OSCL\_IMPORT\_REF int32 OsclUDPSocket::Join ([OsclNetworkAddress](#) & *aAddress*)

Bind a UDP socket to an address and Join the multicast group. This is a synchronous method.

**Parameters:**

*aAddress*: Bind address.

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code. May throw an OsclErrNotSupported Exception

### 7.247.3.11 OSCL\_IMPORT\_REF int32 OsclUDPSocket::JoinMulticastGroup ([OsclIpMReq](#) & *aMReq*)

Join the multicast group.

**Parameters:**

*aMReq*: Multicast group information.

**Returns:**

Returns: OsclErrNone 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

### 7.247.3.12 OSCL\_IMPORT\_REF OsclUDPSocket\* OsclUDPSocket::NewL ([Oscl\\_DefAlloc](#) & *alloc*, [OsclSocketServ](#) & *aServ*, [OsclSocketObserver](#) \* *aObserver*, uint32 *aId*) [static]

Create a UDP Socket. May leave if failure.

**Parameters:**

*alloc*: Memory allocator.

*aServ*: Socket server. Must be connected.

*aObserver*: Socket observer.

*aId*: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

**Returns:**

Returns pointer to socket.

### 7.247.3.13 OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::RecvFrom (uint8 \* *aPtr*, uint32 *aMaxLen*, [OsclNetworkAddress](#) & *aAddress*, int32 *aTimeoutMsec* = -1, uint32 *aMultiRecvLimit* = 0, [Oscl\\_Vector](#)< uint32, [OsclMemAllocator](#) > \* *aPacketLen* = NULL, [Oscl\\_Vector](#)< [OsclNetworkAddress](#), [OsclMemAllocator](#) > \* *aPacketSource* = NULL)

Receive Data. This is an asynchronous method.

**Parameters:**

**aPtr:** Buffer to receive incoming data

**aMaxLen:** Length of buffer.

**aAddress:** (output) Source address.

**aTimeoutMsec:** Timeout in milliseconds, or (-1) for infinite wait.

**aMultiRecvLimit** (optional input): Configures multiple packet receive mode. As long as there are packets queued at the socket and at least aMultiRecvLimit bytes are available in the buffer, recvfrom operations will continue. A value of zero disabled multiple packet mode. The individual packet lengths can be retrieved in the aPacketLen parameter; and the individual packet source addresses can be retrieved in the aPacketSource parameter.

**aPacketLen:** (optional output) a vector of packet lengths, in case multiple packets were received.

**aPacketSource:** (optional output) a vector of source addresses, in case multiple packets were received.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

#### 7.247.3.14 OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::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.

#### 7.247.3.15 OSCL\_IMPORT\_REF int32 OsclUDPSocket::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: OsclErrNone 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

**7.247.3.16 OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetOptionToReuseAddress ()**

Allows the server to bind to an address which is in a TIME\_WAIT state.

**Returns:**

Returns: OsclErrNone 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

**7.247.3.17 OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetRecvBufferSize (uint32 *size*)**

Set the buffer size of the socket This is a synchronous method.

**Parameters:**

*size*: buffer size

**Returns:**

Returns OsclErrNone for success, or a platform-specific error code. May throw an OsclErrNotSupported Exception.

**7.247.3.18 OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetTOS (const OsclSocketTOS & *aTOS*)**

Sets the Type of Service field of each outgoing IP datagram.

**Parameters:**

*aTOS*: Specifies the type of service requested.

**Returns:**

Returns: OsclErrNone 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

**7.247.3.19 OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::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;

**7.247.3.20 OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::ThreadLogon  
(OsclSocketServ & *aServ*, OsclSocketObserver \* *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.

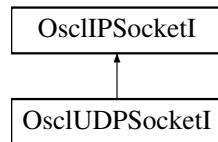
The documentation for this class was generated from the following file:

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

## 7.248 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 \(Oscl\\_DefAlloc &a, OsclSocketServI \\*aServ, OsclSocketObserver \\*aObserver, uint32 aId\)](#)

#### 7.248.1 Detailed Description

Internal implementation class for [OsclUDPSocket](#)

## 7.248.2 Constructor & Destructor Documentation

**7.248.2.1** `virtual OsclUDPSocketI::~OsclUDPSocketI () [virtual]`

## 7.248.3 Member Function Documentation

**7.248.3.1** `TPVSocketEvent OsclUDPSocketI::BindAsync (OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]`

**7.248.3.2** `void OsclUDPSocketI::CancelBind () [inline]`

**7.248.3.3** `void OsclUDPSocketI::CancelRecvFrom () [inline]`

**7.248.3.4** `void OsclUDPSocketI::CancelSendTo () [inline]`

**7.248.3.5** `int32 OsclUDPSocketI::Close () [virtual]`

Implements [OsclIPSocketI](#).

**7.248.3.6** `uint8 * OsclUDPSocketI::GetRecvData (int32 * aLength) [inline, virtual]`

Implements [OsclIPSocketI](#).

**7.248.3.7** `uint8 * OsclUDPSocketI::GetSendData (int32 * aLength) [inline, virtual]`

Implements [OsclIPSocketI](#).

**7.248.3.8** `int32 OsclUDPSocketI::JoinMulticastGroup (OsclIpMReq & aMReq)`

**7.248.3.9** `OsclUDPSocketI* OsclUDPSocketI::NewL (Oscl_DefAlloc & a, OsclSocketServI * aServ, OsclSocketObserver * aObserver, uint32 aId) [static]`

**7.248.3.10** `TPVSocketEvent OsclUDPSocketI::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiMaxLen = 0, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen = NULL, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource = NULL) [inline]`

**7.248.3.11** `TPVSocketEvent OsclUDPSocketI::SendTo (const uint8 *& aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]`

**7.248.3.12** `int32 OsclUDPSocketI::SetMulticastTTL (int32 aTTL)`

**7.248.3.13** `TPVSocketEvent OsclUDPSocketI::ThreadLogoff ()`

Reimplemented from [OsclIPSocketI](#).

**7.248.3.14** `TPVSocketEvent OsclUDPSocketI::ThreadLogon (OsclSocketServI * aServ, OsclSocketObserver * aObserver)`

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

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

## 7.249 OsclUuid Struct Reference

```
#include <oscl_uuid.h>
```

### Public Methods

- [OsclUuid \(\)](#)
- [OsclUuid \(uint32 l, uint16 w1, uint16 w2, uint8 b1, uint8 b2, uint8 b3, uint8 b4, uint8 b5, uint8 b6, uint8 b7, uint8 b8\)](#)
- [OsclUuid \(const OsclUuid &uuid\)](#)
- [OsclUuid & operator= \(const OsclUuid &src\)](#)
- [bool operator== \(const OsclUuid &src\) const](#)
- [bool operator!= \(const OsclUuid &src\) const](#)

### Data Fields

- uint32 [data1](#)
- uint16 [data2](#)
- uint16 [data3](#)
- uint8 [data4 \[BYTES\\_IN\\_UUID\\_ARRAY\]](#)

#### 7.249.1 Detailed Description

OSCL UUID structure used for unique identification of modules and interfaces.

## 7.249.2 Constructor & Destructor Documentation

**7.249.2.1 OsclUuid::OsclUuid () [inline]**

**7.249.2.2 OsclUuid::OsclUuid (uint32 *l*, uint16 *w1*, uint16 *w2*, uint8 *b1*, uint8 *b2*, uint8 *b3*, uint8 *b4*, uint8 *b5*, uint8 *b6*, uint8 *b7*, uint8 *b8*) [inline]**

**7.249.2.3 OsclUuid::OsclUuid (const OsclUuid & *uuid*) [inline]**

## 7.249.3 Member Function Documentation

**7.249.3.1 bool OsclUuid::operator!= (const OsclUuid & *src*) const [inline]**

**7.249.3.2 OsclUuid& OsclUuid::operator= (const OsclUuid & *src*) [inline]**

**7.249.3.3 bool OsclUuid::operator== (const OsclUuid & *src*) const [inline]**

## 7.249.4 Field Documentation

**7.249.4.1 uint32 OsclUuid::data1**

**7.249.4.2 uint16 OsclUuid::data2**

**7.249.4.3 uint16 OsclUuid::data3**

**7.249.4.4 uint8 OsclUuid::data4[BYTES\_IN\_UUID\_ARRAY]**

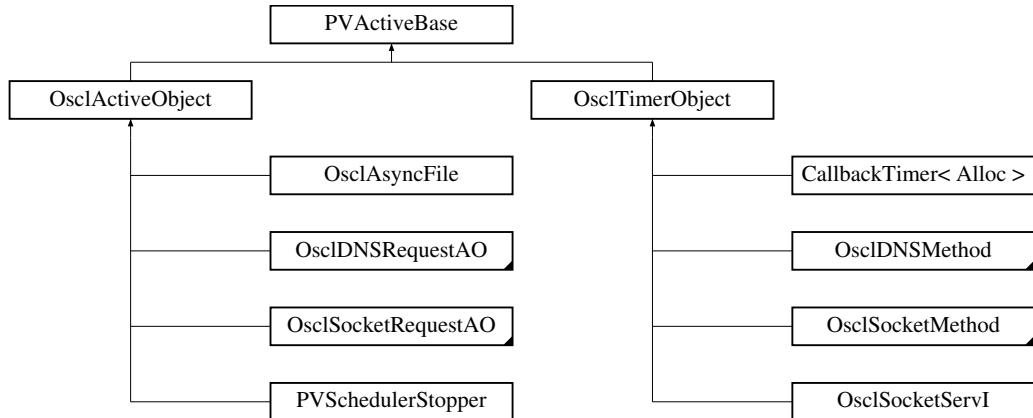
The documentation for this struct was generated from the following file:

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

## 7.250 PVActiveBase Class Reference

```
#include <oscl_scheduler_aobase.h>
```

Inheritance diagram for PVActiveBase::



### Public Methods

- [PVActiveBase](#) (const char name[ ], int32 pri)
- virtual [~PVActiveBase](#) ()
- bool [IsInAnyQ](#) ()
- virtual int32 [RunError](#) (int32 aError)=0
- virtual void [Run](#) ()=0
- virtual void [DoCancel](#) ()=0
- void [AddToScheduler](#) ()
- void [RemoveFromScheduler](#) ()
- void [Destroy](#) ()
- void [Activate](#) ()
- OSCL\_IMPORT\_REF bool [IsAdded](#) () const
- void [Cancel](#) ()

### Data Fields

- uint32 [iAddedNum](#)
- [OsclNameString< PVEXECNAMELEN > iName](#)
- [PVThreadContext iThreadContext](#)
- [PVActiveStats \\* iPVAstats](#)
- [TReadyQueLink iPVReadyQLink](#)
- bool [iBusy](#)
- [OsclAOStatus iStatus](#)

### Friends

- class [PVActiveStats](#)
- class [OsclSchedulerCommonBase](#)

- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [OsclReadyQ](#)
- class [OsclReadyCompare](#)
- class [OsclReadySetPosition](#)
- class [OsclExecScheduler](#)

## 7.250.1 Detailed Description

PV Scheduler internal AO base class. Both [OsclActiveObject](#) and [OsclTimerObject](#) derive from this class. For Symbian, this just container has the desired additions to the basic CTimer or OsclActiveObj functionality. For non-Symbian, this class contains the entire AO implementation.

## 7.250.2 Constructor & Destructor Documentation

**7.250.2.1 PVActiveBase::PVActiveBase (const char *name*[ ], int32 *pri*)**

**7.250.2.2 virtual PVActiveBase::~PVActiveBase () [virtual]**

## 7.250.3 Member Function Documentation

**7.250.3.1 void PVActiveBase::Activate ()**

**7.250.3.2 void PVActiveBase::AddToScheduler ()**

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

**7.250.3.3 void PVActiveBase::Cancel ()**

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

**7.250.3.4 void PVActiveBase::Destroy ()**

**7.250.3.5 virtual void PVActiveBase::DoCancel () [pure virtual]**

Implements cancellation of an outstanding request.

This function is called as part of the active object's [Cancel\(\)](#).

It must call the appropriate cancel function offered by the active object's asynchronous service provider. The asynchronous service provider's cancel is expected to act immediately.

[DoCancel\(\)](#) must not wait for event completion; this is handled by [Cancel\(\)](#).

Implemented in [OsclDNSRequestAO](#), [OsclSocketRequestAO](#), [OsclActiveObject](#), and [OsclTimerObject](#).

**7.250.3.6 OSCL\_IMPORT\_REF bool PVActiveBase::IsAdded ()**

**7.250.3.7 bool PVActiveBase::IsInAnyQ () [inline]**

**7.250.3.8 void PVActiveBase::RemoveFromScheduler ()**

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

**7.250.3.9 virtual void PVActiveBase::Run () [pure virtual]**

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implemented in [OsclDNSMethod](#), [OsclDNSRequestAO](#), [OsclSocketMethod](#), [OsclSocketRequestAO](#), and [CallbackTimer< Alloc >](#).

**7.250.3.10 virtual int32 PVActiveBase::RunError (int32 *aError*) [pure virtual]**

Virtual routine that gets called if the active object's Run routine leaves.

**Parameters:**

***aError*:** the leave code generated by the Run.

**Returns:**

:returns `OsclErrNone` if the error was handled, or returns the input *aError* value if not handled.

Implemented in [OsclActiveObject](#), and [OsclTimerObject](#).

## 7.250.4 Friends And Related Function Documentation

**7.250.4.1 friend class OsclActiveObject [friend]**

**7.250.4.2 friend class OsclExecScheduler [friend]**

**7.250.4.3 friend class OsclReadyCompare [friend]**

**7.250.4.4 friend class OsclReadyQ [friend]**

**7.250.4.5 friend class OsclReadySetPosition [friend]**

**7.250.4.6 friend class OsclSchedulerCommonBase [friend]**

**7.250.4.7 friend class OsclTimerObject [friend]**

**7.250.4.8 friend class PVActiveStats [friend]**

## 7.250.5 Field Documentation

**7.250.5.1 uint32 PVActiveBase::iAddedNum**

**7.250.5.2 bool PVActiveBase::iBusy**

**7.250.5.3 OsclNameString<PVEXECNAMELEN> PVActiveBase::iName**

**7.250.5.4 PVActiveStats\* PVActiveBase::iPVActiveStats**

**7.250.5.5 TReadyQueLink PVActiveBase::iPVReadyQLink**

**7.250.5.6 OsclAOStatus PVActiveBase::iStatus**

The request status associated with an asynchronous request.

This is passed as a parameter to all asynchronous service providers.

The active scheduler uses this to check whether the active object's request has completed.

The function can use the completion code to judge the success or otherwise of the request.

Request status contains one of the values OSCL\_REQUEST\_ERR\_NONE: request completed with no error, or request is not active. OSCL\_REQUEST\_PENDING: request is active & pending OSCL\_REQUEST\_ERR\_CANCEL: request was canceled before completion. or any user-defined value.

**7.250.5.7 PVThreadContext PVActiveBase::iThreadContext**

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

- [oscl\\_scheduler\\_aobase.h](#)

## 7.251 PVActiveStats Class Reference

```
#include <oscl_scheduler_aobase.h>
```

### Friends

- class [PVActiveBase](#)
- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [OsclReadyQ](#)

### 7.251.1 Detailed Description

PV AO statistics

### 7.251.2 Friends And Related Function Documentation

**7.251.2.1 friend class OsclActiveObject [friend]**

**7.251.2.2 friend class OsclExecScheduler [friend]**

**7.251.2.3 friend class OsclExecSchedulerCommonBase [friend]**

**7.251.2.4 friend class OsclReadyQ [friend]**

**7.251.2.5 friend class OsclTimerObject [friend]**

**7.251.2.6 friend class PVActiveBase [friend]**

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

- [oscl\\_scheduler\\_aobase.h](#)

## 7.252 PVLogger Class Reference

```
#include <pvlogger.h>
```

### Public Types

- `typedef int32 log_level_type`
- `typedef int32 message_id_type`
- `typedef int32 filter_status_type`
- `typedef _OsclBasicAllocator alloc_type`

### Public Methods

- `void SetLogLevel (log_level_type level)`
- `OSCL_IMPORT_REF void SetLogLevelAndPropagate (log_level_type level)`
- `log_level_type GetLogLevel ()`
- `void DisableAppenderInheritance ()`
- `void AddAppender (OsclSharedPtr< PVLoggerAppender > &appender)`
- `void RemoveAppender (OsclSharedPtr< PVLoggerAppender > &appender)`
- `void AddFilter (OsclSharedPtr< PVLoggerFilter > &filter)`
- `uint32 GetNumAppenders ()`
- `OSCL_IMPORT_REF bool IsActive (log_level_type level)`
- `OSCL_IMPORT_REF void LogMsgStringV (message_id_type msgID, const char *fmt, va_list arguments)`
- `OSCL_IMPORT_REF void LogMsgBuffersV (message_id_type msgID, int32 numPairs, va_list arguments)`
- `OSCL_IMPORT_REF void LogMsgString (message_id_type msgID, const char *fmt,...)`
- `OSCL_IMPORT_REF void LogMsgBuffers (message_id_type msgID, int32 numPairs,...)`
- `OSCL_IMPORT_REF PVLogger (const char *inputTag, log_level_type level, bool oAppenderInheritance)`
- `virtual ~PVLogger ()`

### Static Public Methods

- `OSCL_IMPORT_REF void Init ()`
- `OSCL_IMPORT_REF void Cleanup ()`
- `OSCL_IMPORT_REF PVLogger * GetLoggerObject (const char *inputTag)`

### Protected Methods

- `void SetParent (PVLogger *parentLogger)`
- `PVLogger * GetParent ()`

### Friends

- class `PVLoggerRegistry`

### 7.252.1 Member Typedef Documentation

7.252.1.1 `typedef _OsclBasicAllocator PVLogger::alloc_type`

7.252.1.2 `typedef int32 PVLogger::filter_status_type`

7.252.1.3 `typedef int32 PVLogger::log_level_type`

7.252.1.4 `typedef int32 PVLogger::message_id_type`

### 7.252.2 Constructor & Destructor Documentation

7.252.2.1 `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

7.252.2.2 `virtual PVLogger::~PVLogger () [inline, virtual]`

### 7.252.3 Member Function Documentation

7.252.3.1 `void PVLogger::AddAppender (OsclSharedPtr< PVLoggerAppender > & appender) [inline]`

This method adds an appender to the logging control point. Each logger maintains a list of appenders. Any msg to a logger if deemed active is logged to all the appenders.

**Parameters:**

*appender* pointer to the appender to add

**Returns:**

NONE

**Exceptions:**

*leaves* if out of memory

7.252.3.2 `void PVLogger::AddFilter (OsclSharedPtr< PVLoggerFilter > & filter) [inline]`

This method adds a message filter to the logging control point. Each logger maintains a list of filters. Any msg to a logger if deemed active is passed through the msg filters prior to logging.

**Parameters:**

*msgFilter* pointer to the filter to add

**Returns:**

NONE

**Exceptions:**

*leaves* if out of memory

**7.252.3.3 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:****7.252.3.4 void PVLogger::DisableAppenderInheritance () [inline]**

This method disables appender inheritance for the logging control point

**7.252.3.5 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

**7.252.3.6 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

**7.252.3.7 uint32 PVLogger::GetNumAppenders () [inline]**

This method returns the number of appenders attached to the logging control point.

**7.252.3.8 PVLogger\* PVLogger::GetParent () [inline, protected]****7.252.3.9 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

**7.252.3.10 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

**7.252.3.11 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

**7.252.3.12 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

**7.252.3.13 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

**7.252.3.14 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

**7.252.3.15 void PVLogger::RemoveAppender (*OsclSharedPtr< PVLoggerAppender > & appender*) [inline]**

This method removes an appender from the logging control point. Each logger maintains a list of appenders. Any msg to a logger if deemed active is logged to all the appenders.

**Parameters:**

*appender* pointer to the appender to delete

**Returns:**

NONE

**7.252.3.16 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

**7.252.3.17 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

**7.252.3.18 void PVLogger::SetParent (PVLogger \**parentLogger*) [inline, protected]****7.252.4 Friends And Related Function Documentation****7.252.4.1 friend class PVLoggerRegistry [friend]**

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

- [pvlogger.h](#)

## 7.253 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`

#### 7.253.1 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.

#### 7.253.2 Member Typedef Documentation

##### 7.253.2.1 `typedef PVLogger::message_id_type PVLoggerAppender::message_id_type`

#### 7.253.3 Constructor & Destructor Documentation

##### 7.253.3.1 `virtual PVLoggerAppender::~PVLoggerAppender () [inline, virtual]`

#### 7.253.4 Member Function Documentation

##### 7.253.4.1 `virtual void PVLoggerAppender::AppendBuffers (message_id_type msgID, int32 numPairs, va_list va) [pure virtual]`

##### 7.253.4.2 `virtual void PVLoggerAppender::AppendString (message_id_type msgID, const char *fmt, va_list va) [pure virtual]`

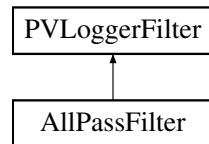
The documentation for this class was generated from the following file:

- `pvlogger_accessories.h`

## 7.254 PVLoggerFilter Class Reference

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

Inheritance diagram for PVLoggerFilter::



### Public Types

- `typedef PVLogger::message_id_type message_id_type`
- `typedef PVLogger::log_level_type log_level_type`
- `typedef PVLogger::filter_status_type filter_status_type`

### Public Methods

- `virtual ~PVLoggerFilter ()`
- `virtual filter_status_type FilterString (char *tag, message_id_type msgID, log_level_type level)=0`
- `virtual filter_status_type FilterOpaqueMessge (char *tag, message_id_type msgID, log_level_type level)=0`

#### 7.254.1 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.

#### 7.254.2 Member Typedef Documentation

##### 7.254.2.1 `typedef PVLogger::filter_status_type PVLoggerFilter::filter_status_type`

Reimplemented in [AllPassFilter](#).

##### 7.254.2.2 `typedef PVLogger::log_level_type PVLoggerFilter::log_level_type`

Reimplemented in [AllPassFilter](#).

##### 7.254.2.3 `typedef PVLogger::message_id_type PVLoggerFilter::message_id_type`

Reimplemented in [AllPassFilter](#).

### 7.254.3 Constructor & Destructor Documentation

7.254.3.1 `virtual PVLoggerFilter::~PVLoggerFilter () [inline, virtual]`

### 7.254.4 Member Function Documentation

7.254.4.1 `virtual filter_status_type PVLoggerFilter::FilterOpaqueMessge (char * tag, message_id_type msgID, log_level_type level) [pure virtual]`

Implemented in [AllPassFilter](#).

7.254.4.2 `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](#)

## 7.255 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`

#### 7.255.1 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.

#### 7.255.2 Member Typedef Documentation

##### 7.255.2.1 `typedef PVLogger::message_id_type PVLoggerLayout::message_id_type`

#### 7.255.3 Constructor & Destructor Documentation

##### 7.255.3.1 `virtual PVLoggerLayout::~PVLoggerLayout () [inline, virtual]`

#### 7.255.4 Member Function Documentation

##### 7.255.4.1 `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.

##### 7.255.4.2 `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](#)

## 7.256 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()`

#### 7.256.1 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.

#### 7.256.2 Member Typedef Documentation

##### 7.256.2.1 `typedef PVLogger::alloc_type PVLoggerRegistry::alloc_type`

##### 7.256.2.2 `typedef PVLogger::log_level_type PVLoggerRegistry::log_level_type`

#### 7.256.3 Constructor & Destructor Documentation

##### 7.256.3.1 `OSCL_IMPORT_REF PVLoggerRegistry::PVLoggerRegistry()`

PVLoggerRegistry Constructor

##### 7.256.3.2 `virtual OSCL_IMPORT_REF PVLoggerRegistry::~PVLoggerRegistry()` [virtual]

PVLoggerRegistry Destructor

## 7.256.4 Member Function Documentation

### 7.256.4.1 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

### 7.256.4.2 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

### 7.256.4.3 OSCL\_IMPORT\_REF PVLoggerRegistry\* PVLoggerRegistry::GetPVLoggerRegistry () [static]

Get the logger registry. There is only one logger registry instance per thread.

### 7.256.4.4 OSCL\_IMPORT\_REF void PVLoggerRegistry::SetNodeLogLevelExplicit (Oscl\_TagTree< PVLogger \*, alloc\_type >::node\_type \* *node*, log\_level\_type *level*)

This method recursively propagates the log level to all the descendants, of a node.

**Parameters:**

*node* Node ptr, associated with a logger, from the tag tree.  
*level* log level associated with the logging control point

**Returns:**

NONE

#### 7.256.4.5 OSCL\_IMPORT\_REF bool PVLoggerRegistry::SetNodeLogLevelExplicit (char \* *tagIn*, *log\_level\_type level*)

This method propagates the log level to all the descendants of the node, with a specified tag.

**Parameters:**

*tagIn* logger tag, viz. "x.y.z"

*level* log level associated with the logging control point

**Returns:**

true on success, else false.

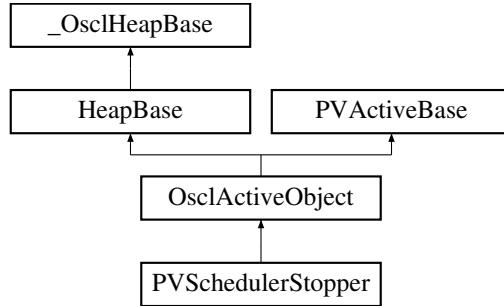
The documentation for this class was generated from the following file:

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

## 7.257 PVSchedulerStopper Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for PVSchedulerStopper::



### Public Methods

- [PVSchedulerStopper \(\)](#)
- [~PVSchedulerStopper \(\)](#)

#### 7.257.1 Detailed Description

Scheduler stopper AO class, for internal use by scheduler.

#### 7.257.2 Constructor & Destructor Documentation

##### 7.257.2.1 PVSchedulerStopper::PVSchedulerStopper ()

##### 7.257.2.2 PVSchedulerStopper::~PVSchedulerStopper ()

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

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

## 7.258 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](#)

#### 7.258.1 Constructor & Destructor Documentation

**7.258.1.1 PVSockBufRecv::PVSockBufRecv () [inline]**

**7.258.1.2 PVSockBufRecv::PVSockBufRecv (uint8 \* *aPtr*, uint32 *aLen*, uint32 *aMax*) [inline]**

**7.258.1.3 PVSockBufRecv::PVSockBufRecv (const PVSockBufRecv & *a*) [inline]**

#### 7.258.2 Field Documentation

**7.258.2.1 uint32 PVSockBufRecv::iLen**

**7.258.2.2 uint32 PVSockBufRecv::iMaxLen**

**7.258.2.3 uint8\* PVSockBufRecv::iPtr**

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

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

## 7.259 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](#)

#### 7.259.1 Constructor & Destructor Documentation

**7.259.1.1 PVSockBufSend::PVSockBufSend () [inline]**

**7.259.1.2 PVSockBufSend::PVSockBufSend (const uint8 \* *aPtr*, uint32 *aLen*) [inline]**

**7.259.1.3 PVSockBufSend::PVSockBufSend (const PVSockBufSend & *a*) [inline]**

#### 7.259.2 Field Documentation

**7.259.2.1 uint32 PVSockBufSend::iLen**

**7.259.2.2 const uint8\* PVSockBufSend::iPtr**

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

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

## 7.260 PVThreadContext Class Reference

```
#include <oscl_scheduler_threadcontext.h>
```

### Public Methods

- OSCL\_IMPORT\_REF PVThreadContext ()
- OSCL\_IMPORT\_REF ~PVThreadContext ()
- OSCL\_IMPORT\_REF bool IsSameThreadContext ()
- OSCL\_IMPORT\_REF void EnterThreadContext ()
- OSCL\_IMPORT\_REF void ExitThreadContext ()

### Static Public Methods

- OSCL\_IMPORT\_REF uint32 Id ()
- OSCL\_IMPORT\_REF bool ThreadHasScheduler ()

### Friends

- class PVActiveBase
- class OsclActiveObject
- class OsclTimerObject
- class OsclExecScheduler
- class OsclCoeActiveScheduler
- class OsclExecSchedulerCommonBase
- class OsclExecSchedulerBase
- class OsclCoeActiveSchedulerBase

#### 7.260.1 Constructor & Destructor Documentation

##### 7.260.1.1 OSCL\_IMPORT\_REF PVThreadContext::PVThreadContext ()

##### 7.260.1.2 OSCL\_IMPORT\_REF PVThreadContext::~PVThreadContext ()

#### 7.260.2 Member Function Documentation

##### 7.260.2.1 OSCL\_IMPORT\_REF void PVThreadContext::EnterThreadContext ()

enter and exit thread context.

##### 7.260.2.2 OSCL\_IMPORT\_REF void PVThreadContext::ExitThreadContext ()

##### 7.260.2.3 OSCL\_IMPORT\_REF uint32 PVThreadContext::Id () [static]

static routine to get a unique thread ID for caller's thread context.

**7.260.2.4 OSCL\_IMPORT\_REF bool PVThreadContext::IsSameThreadContext ()**

compare caller's thread context to this one.

**7.260.2.5 OSCL\_IMPORT\_REF bool PVThreadContext::ThreadHasScheduler () [static]**

a static utility to tell whether the calling thread has any scheduler— either Oscl scheduler or native scheduler.

**7.260.3 Friends And Related Function Documentation****7.260.3.1 friend class OsclActiveObject [friend]****7.260.3.2 friend class OsclCoeActiveScheduler [friend]****7.260.3.3 friend class OsclCoeActiveSchedulerBase [friend]****7.260.3.4 friend class OsclExecScheduler [friend]****7.260.3.5 friend class OsclExecSchedulerBase [friend]****7.260.3.6 friend class OsclExecSchedulerCommonBase [friend]****7.260.3.7 friend class OsclTimerObject [friend]****7.260.3.8 friend class PVActiveBase [friend]**

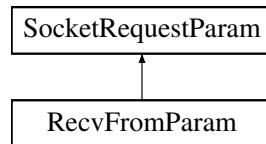
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_threadcontext.h](#)

## 7.261 RecvFromParam Class Reference

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

Inheritance diagram for RecvFromParam::



### Public Methods

- [RecvFromParam \(uint8 \\*aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, uint32 flags, uint32 aMultiMax, Oscl\\_Vector< uint32, OsclMemAllocator > \\*aPacketLen, Oscl\\_Vector< OsclNetworkAddress, OsclMemAllocator > \\*aPacketSource\)](#)

### Data Fields

- [PVSockBufRecv iBufRecv](#)
- [uint32 iFlags](#)
- [OsclNetworkAddress & iAddr](#)
- [uint32 iMultiMaxLen](#)
- [Oscl\\_Vector< uint32, OsclMemAllocator > \\* iPacketLen](#)
- [Oscl\\_Vector< OsclNetworkAddress, OsclMemAllocator > \\* iPacketSource](#)

#### 7.261.1 Constructor & Destructor Documentation

[7.261.1.1 RecvFromParam::RecvFromParam \(uint8 \\*& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, uint32 flags, uint32 aMultiMax, Oscl\\_Vector< uint32, OsclMemAllocator > \\* aPacketLen, Oscl\\_Vector< OsclNetworkAddress, OsclMemAllocator > \\* aPacketSource\) \[inline\]](#)

#### 7.261.2 Field Documentation

[7.261.2.1 OsclNetworkAddress& RecvFromParam::iAddr](#)

[7.261.2.2 PVSockBufRecv RecvFromParam::iBufRecv](#)

[7.261.2.3 uint32 RecvFromParam::iFlags](#)

[7.261.2.4 uint32 RecvFromParam::iMultiMaxLen](#)

[7.261.2.5 Oscl\\_Vector<uint32, OsclMemAllocator>\\* RecvFromParam::iPacketLen](#)

[7.261.2.6 Oscl\\_Vector<OsclNetworkAddress, OsclMemAllocator>\\*> RecvFromParam::iPacketSource](#)

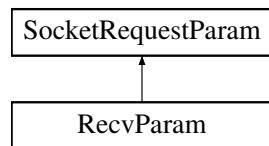
The documentation for this class was generated from the following file:

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

## 7.262 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](#)

#### 7.262.1 Constructor & Destructor Documentation

**7.262.1.1 RecvParam::RecvParam (uint8 \*& aPtr, uint32 aMaxLen, uint32 flags) [inline]**

#### 7.262.2 Field Documentation

**7.262.2.1 PVSockBufRecv RecvParam::iBufRecv**

**7.262.2.2 uint32 RecvParam::iFlags**

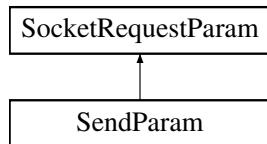
The documentation for this class was generated from the following file:

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

## 7.263 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](#)

#### 7.263.1 Detailed Description

Socket method parameter sets

#### 7.263.2 Constructor & Destructor Documentation

**7.263.2.1 SendParam::SendParam (const uint8 \*& aPtr, uint32 aLen, uint32 aFlags) [inline]**

#### 7.263.3 Field Documentation

**7.263.3.1 PVSockBufSend SendParam::iBufSend**

**7.263.3.2 uint32 SendParam::iFlags**

**7.263.3.3 uint32 SendParam::iXferLen**

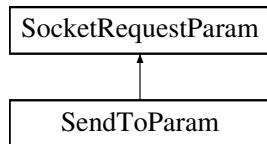
The documentation for this class was generated from the following file:

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

## 7.264 SendToParam Class Reference

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

Inheritance diagram for SendToParam::



### Public Methods

- [SendToParam \(const uint8 \\*aPtr, uint32 aLen, OsclNetworkAddress &aAddr, uint32 flags\)](#)
- [~SendToParam \(\)](#)

### Data Fields

- [PVSockBufSend iBufSend](#)
- [uint32 iFlags](#)
- [OsclNetworkAddress iAddr](#)
- [uint32 iXferLen](#)

#### 7.264.1 Constructor & Destructor Documentation

**7.264.1.1 SendToParam::SendToParam (const uint8 \*& aPtr, uint32 aLen, OsclNetworkAddress & aAddr, uint32 flags) [inline]**

**7.264.1.2 SendToParam::~SendToParam () [inline]**

#### 7.264.2 Field Documentation

**7.264.2.1 OsclNetworkAddress SendToParam::iAddr**

**7.264.2.2 PVSockBufSend SendToParam::iBufSend**

**7.264.2.3 uint32 SendToParam::iFlags**

**7.264.2.4 uint32 SendToParam::iXferLen**

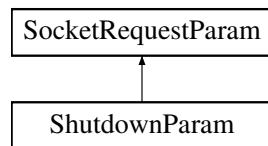
The documentation for this class was generated from the following file:

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

## 7.265 ShutdownParam Class Reference

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

Inheritance diagram for ShutdownParam::



### Public Methods

- [ShutdownParam \(TPVSocketShutdown aHow\)](#)

### Data Fields

- [TPVSocketShutdown iHow](#)

#### 7.265.1 Constructor & Destructor Documentation

**7.265.1.1 ShutdownParam::ShutdownParam ([TPVSocketShutdown aHow](#)) [inline]**

#### 7.265.2 Field Documentation

##### 7.265.2.1 [TPVSocketShutdown ShutdownParam::iHow](#)

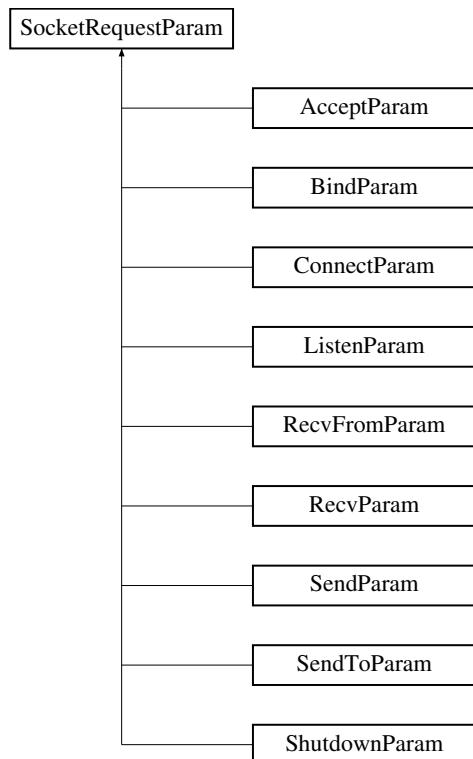
The documentation for this class was generated from the following file:

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

## 7.266 SocketRequestParam Class Reference

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

Inheritance diagram for SocketRequestParam::



### Public Methods

- [SocketRequestParam \(TPVSocketFxn aFxn\)](#)

### Data Fields

- [TPVSocketFxn iFxn](#)

#### 7.266.1 Detailed Description

Base class for all socket method parameter sets

## 7.266.2 Constructor & Destructor Documentation

7.266.2.1 `SocketRequestParam::SocketRequestParam (TPVSocketFxn aFxn)` [inline]

## 7.266.3 Field Documentation

7.266.3.1 `TPVSocketFxn SocketRequestParam::iFxn`

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

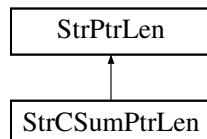
- [oscl\\_socket\\_request.h](#)

## 7.267 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](#)

#### 7.267.1 Detailed Description

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

## 7.267.2 Member Typedef Documentation

**7.267.2.1** `typedef int16 StrCSumPtrLen::CheckSumType`

## 7.267.3 Constructor & Destructor Documentation

**7.267.3.1** `StrCSumPtrLen::StrCSumPtrLen () [inline]`

**7.267.3.2** `StrCSumPtrLen::StrCSumPtrLen (const char * newPtr) [inline]`

**7.267.3.3** `StrCSumPtrLen::StrCSumPtrLen (const char * newPtr, uint32 newLen) [inline]`

**7.267.3.4** `StrCSumPtrLen::StrCSumPtrLen (const StrCSumPtrLen & rhs) [inline]`

**7.267.3.5** `StrCSumPtrLen::StrCSumPtrLen (const StrPtrLen & rhs) [inline]`

## 7.267.4 Member Function Documentation

**7.267.4.1** `CheckSumType StrCSumPtrLen::getCheckSum () const [inline]`

**7.267.4.2** `c_bool StrCSumPtrLen::isCIEquivalentTo (const StrCSumPtrLen & rhs) const [inline]`

**7.267.4.3** `c_bool StrCSumPtrLen::operator!= (const StrCSumPtrLen & rhs) const [inline]`

**7.267.4.4** `StrCSumPtrLen& StrCSumPtrLen::operator= (const char * rhs) [inline]`

Reimplemented from [StrPtrLen](#).

**7.267.4.5** `StrCSumPtrLen& StrCSumPtrLen::operator= (const StrPtrLen & rhs) [inline]`

Reimplemented from [StrPtrLen](#).

**7.267.4.6** `StrCSumPtrLen& StrCSumPtrLen::operator= (const StrCSumPtrLen & rhs) [inline]`

**7.267.4.7** `c_bool StrCSumPtrLen::operator== (const StrCSumPtrLen & rhs) const [inline]`

**7.267.4.8** `OSCL_IMPORT_REF void StrCSumPtrLen::setCheckSum ()`

**7.267.4.9** `void StrCSumPtrLen::setPtrLen (const char * newPtr, uint32 newLen) [inline]`

Reimplemented from [StrPtrLen](#).

## 7.267.5 Field Documentation

**7.267.5.1** `CheckSumType StrCSumPtrLen::checkSum [protected]`

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

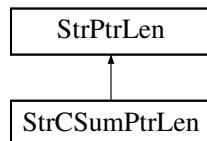
- [oscl\\_str\\_ptr\\_len.h](#)

## 7.268 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](#)

#### 7.268.1 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.

## 7.268.2 Constructor & Destructor Documentation

**7.268.2.1** `StrPtrLen::StrPtrLen (const char * newPtr) [inline]`

**7.268.2.2** `StrPtrLen::StrPtrLen (const char * newPtr, uint32 newLen) [inline]`

**7.268.2.3** `StrPtrLen::StrPtrLen () [inline]`

**7.268.2.4** `StrPtrLen::StrPtrLen (const StrPtrLen & rhs) [inline]`

## 7.268.3 Member Function Documentation

**7.268.3.1** `const char* StrPtrLen::c_str () const [inline]`

**7.268.3.2** `c_bool StrPtrLen::isCIEquivalentTo (const StrPtrLen & rhs) const [inline]`

**7.268.3.3** `c_bool StrPtrLen::isCIPrefixOf (const StrPtrLen & rhs) const [inline]`

**7.268.3.4** `bool StrPtrLen::isLetter (const char c) const [inline, protected]`

**7.268.3.5** `int32 StrPtrLen::length () const [inline]`

**7.268.3.6** `int32 StrPtrLen::operator!= (const StrPtrLen & rhs) const [inline]`

**7.268.3.7** `StrPtrLen& StrPtrLen::operator= (const char * rhs) [inline]`

Reimplemented in [StrCSumPtrLen](#).

**7.268.3.8** `StrPtrLen& StrPtrLen::operator= (const StrPtrLen & rhs) [inline]`

Reimplemented in [StrCSumPtrLen](#).

**7.268.3.9** `int32 StrPtrLen::operator== (const StrPtrLen & rhs) const [inline]`

**7.268.3.10** `void StrPtrLen::setPtrLen (const char * newPtr, uint32 newLen) [inline]`

Reimplemented in [StrCSumPtrLen](#).

**7.268.3.11** `int32 StrPtrLen::size () const [inline]`

## 7.268.4 Field Documentation

**7.268.4.1** `int32 StrPtrLen::len [protected]`

**7.268.4.2** `const char* StrPtrLen::ptr [protected]`

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

- [oscl\\_str\\_ptr\\_len.h](#)

## 7.269 TimeValue Class Reference

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

```
#include <oscl_time.h>
```

### Public Methods

- OSCL\_COND\_IMPORT\_REF [TimeValue](#) ()
 

*Create a TimeValue representing the current time.*
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) (const [TimeValue](#) &Tv)
 

*Copy constructor.*
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) (long tv, [TimeUnits](#) units)
 

*Create a TimeValue representing an interval of tv units.*
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) (const [OsclBasicTimeStruct](#) &in\_tv)
 

*Create a TimeValue representing the absolute time specified by the BasicTimeStruct.*
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) (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](#) ([OsclBasicDateTimeStruct](#) in\_ts)
 

*Create a TimeValue representing the absolute time specified by the BasicDateTimeStruct.*
- OSCL\_COND\_IMPORT\_REF int32 [get\\_local\\_time](#) ()
 

*Get the local time after having adjusted for daylight saving.*
- OSCL\_COND\_IMPORT\_REF void [set\\_to\\_zero](#) ()
 

*Set the time value to zero (represents a zero interval).*
- OSCL\_COND\_IMPORT\_REF void [set\\_to\\_current\\_time](#) ()
 

*Set the time value to the current system time.*
- OSCL\_COND\_IMPORT\_REF void [set\\_from\\_ntp\\_time](#) (const uint32 ntp\_offset)
 

*This method converts 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\_stdbuf)
 

*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\_stdbuf)
 

*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 OsclBasicTimeStruct \* [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)

## 7.269.1 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

## 7.269.2 Constructor & Destructor Documentation

### 7.269.2.1 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue ()

Create a TimeValue representing the current time.

### 7.269.2.2 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (const TimeValue & *Tv*)

Copy constructor.

### 7.269.2.3 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.

### 7.269.2.4 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (const OsclBasicTimeStruct & *in\_tv*)

Create a TimeValue representing the absolute time specified by the BasicTimeStruct.

**Parameters:**

- in\_tv* OsclBasicTimeStruct as defined for each platform.

### 7.269.2.5 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (const ISO8601timeStrBuf *time\_strbuf*)

### 7.269.2.6 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*

**7.269.2.7 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue ([OsclBasicDateTimeStruct in\\_ts](#))**

Create a TimeValue representing the absolute time specified by the BasicDateTimeStruct.

**Parameters:**

*in\_ts* OsclBasicDateTimeStruct as defined for each platform provides the date in a readable format (i.e. day, date, week etc.) Notes: Implementation incomplete (= not done) on Win32, Wince, Symbian

**7.269.3 Member Function Documentation****7.269.3.1 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".

**7.269.3.2 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

**7.269.3.3 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".

**7.269.3.4 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".

**7.269.3.5 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.)

**7.269.3.6 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".

**7.269.3.7 OSCL\_COND\_IMPORT\_REF OsclBasicTimeStruct\* TimeValue::get\_timeval\_ptr ()****7.269.3.8 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)

**7.269.3.9 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.

**7.269.3.10 OSCL\_COND\_IMPORT\_REF bool TimeValue::is\_zero ()**

Determine if the time value is zero.

**7.269.3.11 OSCL\_COND\_IMPORT\_REF bool TimeValue::is\_zulu ()**

Manipulate internal flags to mark the time value as being in "zulu" (GMT) time.

**7.269.3.12 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator \*= (const int scale)**

This operator scales the time value by a constant.

**7.269.3.13 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator+= (const int32 aSeconds)****7.269.3.14 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator+= (const TimeValue & a)**

+= operator

**7.269.3.15 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator-= (const int32 aSeconds)****7.269.3.16 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator-= (const TimeValue & a)**

-= operator

**7.269.3.17 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator= (const TimeValue & a)**

Assignment operator.

**7.269.3.18 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

**7.269.3.19 OSCL\_COND\_IMPORT\_REF void TimeValue::set\_to\_current\_time ()**

Set the time value to the current system time.

**7.269.3.20 OSCL\_COND\_IMPORT\_REF void TimeValue::set\_to\_zero ()**

Set the time value to zero (represents a zero interval).

**7.269.3.21 OSCL\_COND\_IMPORT\_REF void TimeValue::set\_zulu (bool *is\_zulu*)****7.269.3.22 OSCL\_COND\_IMPORT\_REF int32 TimeValue::to\_msec ()****7.269.4 Friends And Related Function Documentation****7.269.4.1 friend class NTPTime [friend]****7.269.4.2 OSCL\_COND\_IMPORT\_REF friend bool operator!= (const TimeValue & *a*, const TimeValue & *b*) [friend]****7.269.4.3 OSCL\_COND\_IMPORT\_REF friend bool operator< (const TimeValue & *a*, const TimeValue & *b*) [friend]****7.269.4.4 OSCL\_COND\_IMPORT\_REF friend bool operator<= (const TimeValue & *a*, const TimeValue & *b*) [friend]****7.269.4.5 OSCL\_COND\_IMPORT\_REF friend bool operator== (const TimeValue & *a*, const TimeValue & *b*) [friend]****7.269.4.6 OSCL\_COND\_IMPORT\_REF friend bool operator> (const TimeValue & *a*, const TimeValue & *b*) [friend]****7.269.4.7 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](#)

## 7.270 TLSStorageOps Class Reference

```
#include <oscl_tls.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [save\\_registry \(TOsclTlsKey \\*key, OsclAny \\*ptr, int32 &\)](#)
- OSCL\_IMPORT\_REF [OsclAny \\* get\\_registry \(TOsclTlsKey \\*key\)](#)

#### 7.270.1 Member Function Documentation

**7.270.1.1 OSCL\_IMPORT\_REF OsclAny\* TLSStorageOps::get\_registry (TOsclTlsKey \* *key*)  
[static]**

**7.270.1.2 OSCL\_IMPORT\_REF void TLSStorageOps::save\_registry (TOsclTlsKey \* *key*,  
OsclAny \* *ptr*, int32 &) [static]**

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

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

## 7.271 TReadyQueLink Class Reference

```
#include <oscl_scheduler_readyq.h>
```

### Public Methods

- [TReadyQueLink \(\)](#)

### Data Fields

- int32 [iAOPriority](#)
- uint32 [iTimeToRunTicks](#)
- uint32 [iTimeQueuedTicks](#)
- uint32 [iSeqNum](#)
- OsclAny \* [iIsIn](#)

#### 7.271.1 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.

#### 7.271.2 Constructor & Destructor Documentation

##### 7.271.2.1 [TReadyQueLink::TReadyQueLink \(\) \[inline\]](#)

#### 7.271.3 Field Documentation

##### 7.271.3.1 [int32 TReadyQueLink::iAOPriority](#)

##### 7.271.3.2 [OsclAny\\* TReadyQueLink::iIsIn](#)

##### 7.271.3.3 [uint32 TReadyQueLink::iSeqNum](#)

##### 7.271.3.4 [uint32 TReadyQueLink::iTimeQueuedTicks](#)

##### 7.271.3.5 [uint32 TReadyQueLink::iTimeToRunTicks](#)

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

- [oscl\\_scheduler\\_readyq.h](#)

## 7.272 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](#)

#### 7.272.1 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.

## 7.272.2 Constructor & Destructor Documentation

- 7.272.2.1 `WStrPtrLen::WStrPtrLen (const oscl_wchar * newPtr) [inline]`
- 7.272.2.2 `WStrPtrLen::WStrPtrLen (const oscl_wchar * newPtr, uint32 newLen) [inline]`
- 7.272.2.3 `WStrPtrLen::WStrPtrLen () [inline]`
- 7.272.2.4 `WStrPtrLen::WStrPtrLen (const WStrPtrLen & rhs) [inline]`

## 7.272.3 Member Function Documentation

- 7.272.3.1 `const oscl_wchar* WStrPtrLen::c_str () const [inline]`
- 7.272.3.2 `c_bool WStrPtrLen::isCIEquivalentTo (const WStrPtrLen & rhs) const [inline]`
- 7.272.3.3 `int32 WStrPtrLen::length () const [inline]`
- 7.272.3.4 `int32 WStrPtrLen::operator!= (const WStrPtrLen & rhs) const [inline]`
- 7.272.3.5 `WStrPtrLen& WStrPtrLen::operator= (const oscl_wchar * rhs) [inline]`
- 7.272.3.6 `WStrPtrLen& WStrPtrLen::operator= (const WStrPtrLen & rhs) [inline]`
- 7.272.3.7 `int32 WStrPtrLen::operator== (const WStrPtrLen & rhs) const [inline]`
- 7.272.3.8 `void WStrPtrLen::setPtrLen (const oscl_wchar * newPtr, uint32 newLen) [inline]`
- 7.272.3.9 `int32 WStrPtrLen::size () const [inline]`

## 7.272.4 Field Documentation

- 7.272.4.1 `int32 WStrPtrLen::len [protected]`
- 7.272.4.2 `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 [OsclAOStatus](#)

#### Variables

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

#### 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"`
- `#define EPV_ARM_GNUC 1`
- `#define EPV_ARM_RVCT 2`
- `#define EPV_ARM_MSEVC 3`

### 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 [OsclBinIStream](#)
- class [OsclBinIStreamBigEndian](#)
- class [OsclBinIStreamLittleEndian](#)
- class [OsclBinOStream](#)

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

- class [OsclBinOStreamBigEndian](#)

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

- class [OsclBinOStreamLittleEndian](#)

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

- class [OsclBinStream](#)

### 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 [OsclIDNS](#)
- class [OsclDNSObserver](#)

### 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 [OsclGetHostByNameMethod](#)
- class [OsclGetHostByNameRequest](#)

## **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 [OsclDNSIBase](#)

## **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 [OsclDNSI](#)

## **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 [OsclIDNSMethod](#)
- class [OsclDNSRequestAO](#)

## 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 [OsclMemAllocator TDNSRequestParamAllocator](#)

#### 8.16.1 Typedef Documentation

##### 8.16.1.1 typedef [OsclMemAllocator 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 [OsclError](#)
- class [OsclErrorTrap](#)
- class [OsclTLSEx](#)
- class [OsclTLSRegistryEx](#)

### Defines

- #define [OSCL\\_TRAPSTACK\\_PUSH\(a\)](#) OsclError::PushL(a)
- #define [OSCL\\_TRAPSTACK\\_POP\(\)](#) OsclError::Pop()
- #define [OSCL\\_TRAPSTACK\\_POPDEALLOC\(\)](#) OsclError::PopDealloc()

### 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 `OsclErrNone` 0
- #define `OsclErrGeneral` 100
- #define `OsclErrNoMemory` 101
- #define `OsclErrCancelled` 102
- #define `OsclErrNotSupported` 103
- #define `OsclErrArgument` 104
- #define `OsclErrBadHandle` 105
- #define `OsclErrAlreadyExists` 106
- #define `OsclErrBusy` 107
- #define `OsclErrNotReady` 108
- #define `OsclErrCorrupt` 109
- #define `OsclErrTimeout` 110
- #define `OsclErrOverflow` 111
- #define `OsclErrUnderflow` 112
- #define `OsclErrInvalidState` 113
- #define `OsclErrNoResources` 114
- #define `OsclErrNotInstalled` 115
- #define `OsclErrAlreadyInstalled` 116
- #define `OsclErrSystemCallFailed` 117
- #define `OsclErrNoHandler` 118
- #define `OsclErrThreadContextIncorrect` 119
- #define `OSCL_ERR_NONE` `OsclErrNone`
- #define `OSCL_BAD_ALLOC_EXCEPTION_CODE` `OsclErrNoMemory`
- #define `OsclSuccess` 0
- #define `OsclPending` 1
- #define `OsclFailure` -1

### Typedefs

- typedef int32 `OsclLeaveCode`
- typedef int32 `OsclReturnCode`

### 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 PVERROR\_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 PVError\_DoLeave() \_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=OsclErrNone;\  
{__s;}
```

##### 8.26.2.2 #define \_PV\_TRAP\_NO\_TLS(\_\_tr, \_\_r, \_\_s)

**Value:**

```
__r=OsclErrNone;\  
{__s;}
```

##### 8.26.2.3 #define PVError\_DoLeave() \_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 PVError\_DoLeave() 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 [OsclErrorTrapImp](#)
- class [OsclTrapStack](#)
- class [OsclTrapStackItem](#)

### Defines

- #define [OSCL\\_MAX\\_TRAP\\_LEVELS](#) 20
- #define [PVERRORTRAP\\_REGISTRY\\_ID](#) [OSCL\\_TLS\\_ID\\_PVERRORTRAP](#)
- #define [PVERRORTRAP\\_REGISTRY](#) [OsclTLSRegistry](#)

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

```
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclExclusiveArrayPtr](#)

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

- class [OsclExclusivePtr](#)

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

- class [OsclExclusivePtrA](#)

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

### 8.30.1 Detailed Description

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

## 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 [OsclFileCache](#).

```
#include "osclconfig_io.h"  
#include "oscl_base.h"  
#include "oscl_file_io.h"
```

### Data Structures

- class [OsclFileCache](#)
- class [OsclFileCacheBuffer](#)

#### 8.32.1 Detailed Description

The file [oscl\\_file\\_cache.h](#) defines the class [OsclFileCache](#).

## 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` (const `oscl_wchar` \*path, uint32 size)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE `oscl_getcwd` (const 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 [OsclFileHandle](#).

```
#include "osclconfig_io.h"  
#include "oscl_base.h"
```

### Data Structures

- class [OsclFileHandle](#)

### TypeDefs

- [typedef FILE \\* TOsclFileHandle](#)

#### 8.35.1 Detailed Description

The file [oscl\\_file\\_handle.h](#) defines the class [OsclFileHandle](#).

## 8.36 oscl\_file\_io.h File Reference

The file [oscl\\_file\\_io.h](#) defines the class [Oscl\\_File](#). This is the public API to the basic file I/O operations.

```
#include "osclconfig_io.h"  
#include "oscl_base.h"  
#include "oscl_mem.h"  
#include "oscl_vector.h"  
#include "oscl_file_server.h"  
#include "oscl_file_find.h"  
#include "oscl_file_dir_utils.h"  
#include "oscl_file_handle.h"
```

### Data Structures

- class [Oscl\\_File](#)
- class [OsclFixedCacheParam](#)
- class [OsclCacheObserver](#)

### Defines

- #define [TOsclFileOffsetInt32](#) int32

#### 8.36.1 Detailed Description

The file [oscl\\_file\\_io.h](#) defines the class [Oscl\\_File](#). This is the public API to the basic file I/O operations.

## 8.37 oscl\_file\_manager.h File Reference

File management class.

```
#include "oscl_base.h"
```

### Data Structures

- class [OsclFileManager](#)

#### 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 [OsclNativeFile](#). This is the porting layer for basic file I/O operations.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_aostatus.h"
#include "oscl_file_io.h"
```

### Data Structures

- class [OsclNativeFile](#)

#### 8.38.1 Detailed Description

The file [oscl\\_file\\_native.h](#) defines the class [OsclNativeFile](#). 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 [OsclFileStats](#)
- class [OsclFileStatsItem](#)

### Defines

- #define [OSCL\\_FILE\\_STATS\\_LOGGER\\_NODE](#) "OsclFileStats"

### Enumerations

- enum [TOsclFileOp](#) { [EOsclFileOp\\_Open](#), [EOsclFileOp\\_Close](#), [EOsclFileOp\\_Read](#), [EOsclFileOp\\_Write](#), [EOsclFileOp\\_Seek](#), [EOsclFileOp\\_Tell](#), [EOsclFileOp\\_Size](#), [EOsclFileOp\\_Flush](#), [EOsclFileOp\\_EndOfFile](#), [EOsclFileOp\\_SetSize](#), [EOsclFileOp\\_NativeOpen](#), [EOsclFileOp\\_NativeClose](#), [EOsclFileOp\\_NativeRead](#), [EOsclFileOp\\_NativeWrite](#), [EOsclFileOp\\_NativeSeek](#), [EOsclFileOp\\_NativeTell](#), [EOsclFileOp\\_NativeSize](#), [EOsclFileOp\\_NativeFlush](#), [EOsclFileOp\\_NativeEndOfFile](#), [EOsclFileOp\\_NativeSetSize](#), [EOsclFileOp\\_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 [OsclNativeFileParams](#)

### Defines

- #define [OSCL\\_IO\\_FILENAME\\_MAXLEN](#) 512
- #define [OSCL\\_IO\\_EXTENSION\\_MAXLEN](#) 512
- #define [OSCL\\_FILE\\_WCHAR\\_PATH\\_DELIMITER](#) \_STRLIT("/")
- #define [OSCL\\_FILE\\_CHAR\\_PATH\\_DELIMITER](#) \_STRLIT\_CHAR("/")

### 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 [\\_OsclHeapBase](#)
- class [OsclTrapItem](#)

### Typedefs

- [typedef void\(\\* OsclTrapOperation \)\(OsclAny \\*\)](#)

#### 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 [Oscl\\_Int64\\_Utils](#)  
*The Oscl\_Int64\_Utils class provides a wrapper for commonly used int64/uint64 operations.*
- struct [OsclInteger64Transport](#)

### Typedefs

- typedef [OsclInteger64Transport \\_OsclInteger64Transport](#)

#### 8.44.1 Typedef Documentation

##### 8.44.1.1 typedef struct [OsclInteger64Transport \\_OsclInteger64Transport](#)

###### [OsclInteger64Transport](#) Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

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

### Data Structures

- class [OsclLockBase](#)
- class [OsclNullLock](#)
- class [OsclScopedLock](#)

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

### 8.47.1 Detailed Description

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

## 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_Select1st`
- 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 [BuffFragGroup](#)
- 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 [OsclAuditCB](#)
- class [OsclMem](#)
- class [OsclMemAllocator](#)
- class [OsclMemAllocDestructDealloc](#)
- class [OsclMemBasicAllocator](#)
- class [OsclMemBasicAllocDestructDealloc](#)
- class [OsclMemGlobalAuditObject](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [OSCL\\_HAS\\_GLOBAL\\_NEW\\_DELETE](#) 1
- #define [OSCL\\_CLEANUP\\_BASE\\_CLASS\(T\)](#) \_OSCL\_CLEANUP\_BASE\_CLASS(T)
- #define [OSCL\\_ALLOC\\_NEW\(T\\_allocator, T, params\)](#) new(T\_allocator.allocate(1)) T params
- #define [OSCL\\_TRAP\\_ALLOC\\_NEW\(T\\_ptr, T\\_allocator, T, params\)](#) \_OSCL\_TRAP\_NEW(T\_allocator.allocate(1),T\_allocator.deallocate,T\_ptr,T,params)
- #define [OSCL\\_ALLOC\\_DELETE\(ptr, T\\_allocator, T\)](#)
- #define [OSCL\\_MALLOC\(count\)](#) \_oscl\_default\_audit\_malloc(count)
- #define [oscl\\_malloc\(a\)](#) OSCL\_MALLOC(a)
- #define [OSCL\\_DEFAULT\\_MALLOC\(x\)](#) OSCL\_MALLOC(x)
- #define [OSCL\\_AUDIT\\_MALLOC\(auditCB, count\)](#) \_oscl\_audit\_malloc(count, auditCB)
- #define [OSCL\\_CALLOC\(num, size\)](#) \_oscl\_default\_audit\_calloc(num,size)
- #define [oscl\\_calloc\(a, b\)](#) OSCL\_CALLOC(a,b)

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

## Functions

- **OSCL\_COND\_IMPORT\_REF** **uint** **oscl\_mem\_aligned\_size** (**uint** **size**)
- **OSCL\_IMPORT\_REF** **void** **OsclMemInit** (**OsclAuditCB** &auditCB)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_audit\_malloc** (**size\_t**, **OsclAuditCB** &, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_audit\_calloc** (**size\_t**, **size\_t**, **OsclAuditCB** &, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_audit\_realloc** (**void** \*, **size\_t**, **OsclAuditCB** &, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_audit\_new** (**size\_t**, **OsclAuditCB** &, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_default\_audit\_malloc** (**size\_t**, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_default\_audit\_calloc** (**size\_t**, **size\_t**, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_default\_audit\_realloc** (**void** \*, **size\_t**, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void \*** **\_oscl\_default\_audit\_new** (**size\_t**, **const char** \*f=NULL, **const int** l=0)
- **OSCL\_IMPORT\_REF** **void** **\_oscl\_audit\_free** (**void** \*)
- **void \*** **operator new** (**size\_t** aSize, **const char** \*aFile, **int** aLine)
- **void \*** **operator new** (**size\_t** aSize)
- **void operator delete** (**void** \*aPtr)
- **void \*** **operator new[]** (**size\_t** aSize, **const char** \*aFile, **int** aLine)
- **void \*** **operator new[]** (**size\_t** aSize)
- **void operator delete[]** (**void** \*aPtr)

### 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\_audit.h File Reference

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

```
#include "oscl_lock_base.h"
#include "oscl_base_alloc.h"
#include "oscl_tagtree.h"
#include "oscl_mem.h"
#include "oscl_mem_auto_ptr.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- struct [MM\\_AllocInfo](#)
- struct [MM\\_AllocNode](#)
- struct [MM\\_AllocQueryInfo](#)
- class [MM\\_Audit\\_Imp](#)
- struct [MM\\_AuditOverheadStats](#)
- struct [MM\\_FailInsertParam](#)
- struct [MM\\_Stats\\_CB](#)
- struct [MM\\_Stats\\_t](#)
- class [OsclMemAudit](#)
- class [OsclMemStatsNode](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [MM\\_ALLOC\\_MAX\\_QUERY\\_FILENAME\\_LEN](#) 128
- #define [MM\\_ALLOC\\_MAX\\_QUERY\\_TAG\\_LEN](#) 64
- #define [MM\\_AUDIT\\_VALIDATE\\_BLOCK](#) 1
- #define [MM\\_AUDIT\\_PREFILL\\_FLAG](#) 0x1
- #define [MM\\_AUDIT\\_POSTFILL\\_FLAG](#) 0x2
- #define [MM\\_AUDIT\\_VALIDATE\\_ALL\\_HEAP\\_FLAG](#) 0x4
- #define [MM\\_AUDIT\\_VALIDATE\\_ON\\_FREE\\_FLAG](#) 0x8
- #define [MM\\_AUDIT\\_ALLOC\\_NODE\\_ENABLE\\_FLAG](#) 0x10
- #define [MM\\_AUDIT\\_SUPPRESS\\_FILENAME\\_FLAG](#) 0x20
- #define [DEFAULT\\_MM\\_AUDIT\\_MODE](#) 0

### Typedefs

- typedef [OSCLMemAutoPtr< char, Oscl\\_TAlloc< char, OsclMemBasicAllocator > >](#) [MMAudit\\_CharAutoPtr](#)
- typedef [OSCLMemAutoPtr< uint8, Oscl\\_TAlloc< uint8, \\_OsclBasicAllocator > >](#) [MMAudit\\_Uint8AutoPtr](#)
- typedef [OSCLMemAutoPtr< MM\\_AllocNode, Oscl\\_TAlloc< MM\\_AllocNode, OsclMemBasicAllocator > >](#) [MM\\_AllocNodeAutoPtr](#)
- typedef [OSCLMemAutoPtr< OsclMemStatsNode, Oscl\\_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > >](#) [MM\\_StatsNodeTagTreeType](#)

- `typedef OSCLMemAutoPtr< OsclMemStatsNode, Oscl_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > > OsclMemStatsNodeAutoPtr`
- `typedef Oscl_TAlloc< MM_StatsNodeTagTreeType, OsclMemBasicAllocator > TagTree_Allocator`
- `typedef Oscl_TagTree< MM_StatsNodeTagTreeType, TagTree_Allocator > OsclTagTreeType`

### 8.53.1 Detailed Description

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

### 8.53.2 Define Documentation

#### 8.53.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

## 8.54 oscl\_mem\_audit\_internals.h File Reference

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

```
#include "oscl_base.h"
#include "oscl_mem_audit.h"
#include "oscl_mem_inst.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- struct [MM\\_AllocBlockFence](#)
- struct [MM\\_AllocBlockHdr](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [MM\\_AUDIT\\_ALLOC\\_NODE\\_SUPPORT](#) 1
- #define [MM\\_AUDIT\\_FENCE\\_SUPPORT](#) 0
- #define [MM\\_AUDIT\\_INCLUDE\\_ALL\\_HEAP\\_VALIDATION](#) 1
- #define [MM\\_AUDIT\\_FILL\\_SUPPORT](#) 0
- #define [MM\\_AUDIT\\_FAILURE\\_SIMULATION\\_SUPPORT](#) 1
- #define [FENCE\\_PATTERN](#) 0xAA
- #define [MIN\\_FENCE\\_SIZE](#) 4
- #define [MEM\\_ALIGN\\_SIZE](#) 8
- #define [COMPUTE\\_MEM\\_ALIGN\\_SIZE](#)(x, y, z) (y+(((x+y)%z) ? (z - (x+y)%z) : 0))
- #define [DEFAULT\\_PREFILL\\_PATTERN](#) 0x96
- #define [DEFAULT\\_POSTFILL\\_PATTERN](#) 0x5A

### 8.54.1 Detailed Description

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

### 8.54.2 Define Documentation

#### 8.54.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

## 8.55 oscl\_mem\_auto\_ptr.h File Reference

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

```
#include "osclconfig_memory.h"  
#include "osclconfig_compiler_warnings.h"  
#include "oscl_mem.h"
```

### Data Structures

- class [OSCLMemAutoPtr](#)

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

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [OSCL\\_DISABLE\\_WARNING\\_RETURN\\_TYPE\\_NOT\\_UDT](#)

#### 8.55.1 Detailed Description

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

#### 8.55.2 Define Documentation

##### 8.55.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

## 8.56 oscl\_mem\_basic\_functions.h File Reference

This file contains prototypes for the basic memory functions.

```
#include "oscl_base_macros.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)

### 8.56.1 Detailed Description

This file contains prototypes for the basic memory functions.

## **8.57 oscl\_mem\_inst.h File Reference**

The file defines default memory instrumentation level.

```
#include "osclconfig_memory.h"
```

### **Defines**

- #define **PVMEM\_INST\_LEVEL** 1

#### **8.57.1 Detailed Description**

The file defines default memory instrumentation level.

## 8.58 oscl\_mem\_mempool.h File Reference

This file contains the definition of memory pool allocators.

```
#include "oscl_mem.h"  
#include "oscl_defalloc.h"  
#include "oscl_vector.h"
```

### Data Structures

- struct [MemPoolBlockInfo](#)
- struct [MemPoolBufferInfo](#)
- class [OsclMemPoolFixedChunkAllocator](#)
- class [OsclMemPoolFixedChunkAllocatorObserver](#)
- class [OsclMemPoolResizableAllocator](#)
- class [OsclMemPoolResizableAllocatorMemoryObserver](#)
- class [OsclMemPoolResizableAllocatorObserver](#)

#### 8.58.1 Detailed Description

This file contains the definition of memory pool allocators.

## 8.59 oscl\_mutex.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"  
#include "oscl_types.h"  
#include "oscl_base.h"  
#include "oscl_thread.h"  
#include "oscl_lock_base.h"
```

### Data Structures

- class [OsclMutex](#)
- class [OsclThreadLock](#)

### Typedefs

- typedef [OsclMutex OsclNoYieldMutex](#)

#### 8.59.1 Detailed Description

This file provides implementation of mutex.

#### 8.59.2 Typedef Documentation

##### 8.59.2.1 typedef [OsclMutex OsclNoYieldMutex](#)

Class [OsclNoYieldMutex](#) can be used in use cases where there will be no CPU-yielding operation done while the Mutex is locked.

CPU-yielding operations include [OsclMutex::Lock](#), [OsclSemaphore::Wait](#), [OsclThread::Sleep](#), and [OsclBrewThreadUtil::BThreadYield](#).

The behavior of [OsclNoYieldMutex](#) depends on whether the threading model is pre-emptive or not. When threading is pre-emptive, it is identical to [OsclMutex](#). When threading is non-pre-emptive, it is a NO-OP.

An example of this type of use case is for simple data protection.

## 8.60 oscl\_namestring.h File Reference

Name string class include file.

```
#include "oscl_base.h"
```

### Data Structures

- class [OsclNameString](#)

#### 8.60.1 Detailed Description

Name string class include file.

## **8.61 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](#)

#### **8.61.1 Detailed Description**

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

## 8.62 oscl\_priqueue.h File Reference

Implements a priority queue data structure similar to STL.

```
#include "oscl_base.h"  
#include "oscl_vector.h"
```

### Data Structures

- class [OsclCompareLess](#)
- class [OsclPriorityQueue](#)
- class [OsclPriorityQueueBase](#)

#### 8.62.1 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.

## **8.63 oscl\_procstatus.h File Reference**

### **Data Structures**

- class [OsclProcStatus](#)

## 8.64 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](#)

#### 8.64.1 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.

## 8.65 oscl\_rand.h File Reference

Provides pseudo-random number generation.

```
#include "osclconfig_util.h"  
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.h"  
#include "oscl_rand.inl"
```

### Data Structures

- class [OsclRand](#)

#### 8.65.1 Detailed Description

Provides pseudo-random number generation.

## 8.66 oscl\_refcounter.h File Reference

A general purpose reference counter to object lifetimes.

```
#include "oscl_assert.h"  
#include "oscl_defalloc.h"
```

### Data Structures

- class [Oscl\\_DefAllocWithRefCounter](#)
- class [OsclRefCounter](#)
- class [OsclRefCounterDA](#)
- class [OsclRefCounterMTDA](#)
- class [OsclRefCounterMTSA](#)
- class [OsclRefCounterSA](#)

#### 8.66.1 Detailed Description

A general purpose reference counter to object lifetimes.

## 8.67 oscl\_refcounter\_memfrag.h File Reference

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

```
#include "oscl_base.h"  
#include "oscl_refcounter.h"
```

### Data Structures

- class [OsclRefCounterMemFrag](#)

#### 8.67.1 Detailed Description

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

## 8.68 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](#)

#### 8.68.1 Detailed Description

Client-side implementation Registry Access implementation.

## 8.69 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](#)

#### 8.69.1 Detailed Description

Client-side implementation of OsclRegistry.

## 8.70 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](#)

#### 8.70.1 Detailed Description

Client-side implementation of OsclRegistryInterface.

## 8.71 oscl\_registry\_serv\_impl.h File Reference

Server-side implementation of OsclRegistry interfaces.

```
#include "oscl_base.h"
#include "osclconfig_proc.h"
#include "oscl_registry_types.h"
#include "oscl_string.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
```

### Data Structures

- class [OsclComponentRegistry](#)
- class [OsclComponentRegistryData](#)
- class [OsclComponentRegistryElement](#)

#### 8.71.1 Detailed Description

Server-side implementation of OsclRegistry interfaces.

## **8.72 oscl\_registry\_serv\_impl\_global.h File Reference**

```
#include "osclconfig_proc.h"
#include "oscl_base.h"
```

## 8.73 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](#)

## 8.74 oscl\_registry\_types.h File Reference

Common types used in Oscl registry interfaces.

```
#include "oscl_types.h"  
#include "oscl_string_containers.h"
```

### Data Structures

- class [OsclRegistryAccessElement](#)

### TypeDefs

- typedef [OsclAny](#) \* [OsclComponentFactory](#)

#### 8.74.1 Detailed Description

Common types used in Oscl registry interfaces.

## 8.75 oscl\_scheduler.h File Reference

```
#include "oscl_scheduler_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_defalloc.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclScheduler](#)
- class [OsclSchedulerObserver](#)
- class [PVSchedulerStopper](#)

### Defines

- #define [PVSCHEDNAMELEN](#) 30

## **8.76 oscl\_scheduler\_ao.h File Reference**

Oscl Scheduler user execution object classes.

```
#include "oscl_scheduler_aobase.h"  
#include "oscl_mem.h"  
#include "oscl_scheduler_types.h"
```

### **Data Structures**

- class [OsclActiveObject](#)
- class [OsclTimerObject](#)

#### **8.76.1 Detailed Description**

Oscl Scheduler user execution object classes.

## 8.77 oscl\_scheduler\_aobase.h File Reference

Oscl Scheduler internal active object classes.

```
#include "oscl_namestring.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_scheduler_readyq.h"
#include "oscl_string_containers.h"
#include "oscl_scheduler_types.h"
```

### Data Structures

- class [PVActiveBase](#)
- class [PVActiveStats](#)

### Defines

- #define [OSCL\\_ZEROIZE](#)(ptr, size) oscl\_memset(ptr, 0, size)
- #define [PVEEXECNAMELEN](#) 30

### 8.77.1 Detailed Description

Oscl Scheduler internal active object classes.

## 8.78 oscl\_scheduler\_readyq.h File Reference

ready q types for oscl scheduler

```
#include "oscl_scheduler_tuneables.h"
#include "oscl_priqueue.h"
#include "oscl_base_alloc.h"
#include "oscl_semaphore.h"
#include "oscl_mem.h"
#include "oscl_string_containers.h"
#include "oscl_scheduler_types.h"
#include "oscl_mutex.h"
```

### Data Structures

- class [OsclReadyAlloc](#)
- class [OsclReadyCompare](#)
- class [OsclReadyQ](#)
- class [OsclTimerCompare](#)
- class [OsclTimerQ](#)
- class [TReadyQueLink](#)

### Typedefs

- typedef [PVActiveBase](#) \* TOsclReady

#### 8.78.1 Detailed Description

ready q types for oscl scheduler

## 8.79 oscl\_scheduler\_threadcontext.h File Reference

Thread context functions needed by oscl scheduler.

```
#include "oscl_double_list.h"
#include "oscl_mutex.h"
#include "oscl_aostatus.h"
```

### Data Structures

- class [PVThreadContext](#)

### Enumerations

- enum [TPVThreadContext](#) { [EPVThreadContext\\_InThread](#), [EPVThreadContext\\_OsclThread](#), [EPVThreadContext\\_NonOsclThread](#), [EPVThreadContext\\_Undetermined](#) }

### 8.79.1 Detailed Description

Thread context functions needed by oscl scheduler.

## 8.80 oscl\_scheduler\_tuneables.h File Reference

Tunable settings for Oscl Scheduler.

```
#include "osclconfig_proc.h"
```

### Defines

- #define PV\_SCHED\_ENABLE\_AO\_STATS 1
- #define PV\_SCHED\_ENABLE\_LOOP\_STATS 0
- #define PV\_SCHED\_ENABLE\_PERF\_LOGGING 1
- #define PV\_SCHED\_ENABLE\_THREAD\_CONTEXT\_CHECKS 1
- #define PV\_SCHED\_LOG\_Q 0
- #define PV\_SCHED\_CHECK\_Q 0
- #define PV\_SCHED\_FAIR\_SCHEDULING 1
- #define OSCL\_PERF\_SUMMARY\_LOGGING 0

### 8.80.1 Detailed Description

Tunable settings for Oscl Scheduler.

## 8.81 oscl\_scheduler\_types.h File Reference

Scheduler common types include file.

```
#include "osclconfig_proc.h"  
#include "oscl_aostatus.h"  
#include "oscl_heapbase.h"
```

### Data Structures

- class [OsclExecSchedulerBase](#)

#### 8.81.1 Detailed Description

Scheduler common types include file.

## 8.82 oscl\_semaphore.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"  
#include "oscl_thread.h"
```

### Data Structures

- class [OsclSemaphore](#)

#### 8.82.1 Detailed Description

This file provides implementation of mutex.

## 8.83 oscl\_shared\_ptr.h File Reference

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

```
#include "oscl_base.h"
#include "oscl_refcounter.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- class [OsclSharedPtr](#)  
*A parameterized smart pointer class.*

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_RETURN\\_TYPE\\_NOT\\_UDT](#)

#### 8.83.1 Detailed Description

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

## 8.84 oscl\_singleton.h File Reference

This file defines the [OsclSingleton](#) class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclSingleton](#)
- class [OsclSingletonRegistry](#)
- class [SingletonTable](#)

### Variables

- const uint32 [OSCL\\_SINGLETON\\_ID\\_TEST](#) = 0
- const uint32 [OSCL\\_SINGLETON\\_ID\\_OSCLMEM](#) = 1
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVLOGGER](#) = 2
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVSCHEDULER](#) = 3
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVERRORTRAP](#) = 4
- const uint32 [OSCL\\_SINGLETON\\_ID\\_SDPMEDIAPARSER](#) = 5
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PAYLOADPARSER](#) = 6
- const uint32 [OSCL\\_SINGLETON\\_ID\\_CPM\\_PLUGIN](#) = 7
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVMFRECOGNIZER](#) = 8
- const uint32 [OSCL\\_SINGLETON\\_ID\\_OSCLREGISTRY](#) = 9
- const uint32 [OSCL\\_SINGLETON\\_ID\\_OMX](#) = 10
- const uint32 [OSCL\\_SINGLETON\\_ID\\_OMXMASTERCORE](#) = 11
- const uint32 [OSCL\\_SINGLETON\\_ID\\_TICKCOUNT](#) = 12
- const uint32 [OSCL\\_SINGLETON\\_ID\\_WMDRMLOCK](#) = 13
- const uint32 [OSCL\\_SINGLETON\\_ID\\_LAST](#) = 14

### 8.84.1 Detailed Description

This file defines the [OsclSingleton](#) class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

[OsclSingleton](#) is initialized in [OsclBase::Init](#).

## 8.84.2 Variable Documentation

- 8.84.2.1 `const uint32 OSCL_SINGLETON_ID_CPM_PLUGIN = 7`
- 8.84.2.2 `const uint32 OSCL_SINGLETON_ID_LAST = 14`
- 8.84.2.3 `const uint32 OSCL_SINGLETON_ID_OMX = 10`
- 8.84.2.4 `const uint32 OSCL_SINGLETON_ID_OMXMASTERCORE = 11`
- 8.84.2.5 `const uint32 OSCL_SINGLETON_ID_OSCLMEM = 1`
- 8.84.2.6 `const uint32 OSCL_SINGLETON_ID_OSCLREGISTRY = 9`
- 8.84.2.7 `const uint32 OSCL_SINGLETON_ID_PAYLOADPARSER = 6`
- 8.84.2.8 `const uint32 OSCL_SINGLETON_ID_PVERRORTRAP = 4`
- 8.84.2.9 `const uint32 OSCL_SINGLETON_ID_PVLOGGER = 2`
- 8.84.2.10 `const uint32 OSCL_SINGLETON_ID_PVMFRECOGNIZER = 8`
- 8.84.2.11 `const uint32 OSCL_SINGLETON_ID_PVSCHEDULER = 3`
- 8.84.2.12 `const uint32 OSCL_SINGLETON_ID_SDPMEDIAPARSER = 5`
- 8.84.2.13 `const uint32 OSCL_SINGLETON_ID_TEST = 0`
- 8.84.2.14 `const uint32 OSCL_SINGLETON_ID_TICKCOUNT = 12`
- 8.84.2.15 `const uint32 OSCL_SINGLETON_ID_WMDRMLOCK = 13`

## 8.85 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)

### 8.85.1 Detailed Description

Provides a portable implementation of snprintf.

## 8.86 oscl\_socket.h File Reference

The file [oscl\\_socket.h](#) defines the OSCL Socket APIs.

```
#include "osclconfig_io.h"  
#include "oscl_heapbase.h"  
#include "oscl_defalloc.h"  
#include "oscl_vector.h"  
#include "oscl_mem.h"  
#include "oscl_socket_types.h"
```

### Data Structures

- class [OsclSocketServ](#)
- class [OsclTCPSocket](#)
- class [OsclUDPSocket](#)

#### 8.86.1 Detailed Description

The file [oscl\\_socket.h](#) defines the OSCL Socket APIs.

## 8.87 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](#)

## 8.88 oscl\_socket\_bind.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclBindMethod](#)
- class [OsclBindRequest](#)

## 8.89 oscl\_socket\_connect.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclConnectMethod](#)
- class [OsclConnectRequest](#)

## **8.90 oscl\_socket\_imp.h File Reference**

```
#include "oscl_socket_tuneables.h"  
#include "oscl_socket_imp_pv.h"
```

## 8.91 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](#)

## 8.92 oscl\_socket\_imp\_pv.h File Reference

```
#include "oscl_socket_imp_base.h"
```

### Data Structures

- class [OsclSocketI](#)

### Defines

- #define [PVSOCK\\_ERR\\_BAD\\_PARAM](#) (-1)
- #define [PVSOCK\\_ERR SOCK\\_NOT\\_OPEN](#) (-2)
- #define [PVSOCK\\_ERR SOCK\\_NO\\_SERV](#) (-3)
- #define [PVSOCK\\_ERR SERV\\_NOT\\_CONNECTED](#) (-4)
- #define [PVSOCK\\_ERR SOCK\\_NOT\\_CONNECTED](#) (-5)
- #define [PVSOCK\\_ERR NOT\\_IMPLEMENTED](#) (-6)
- #define [PVSOCK\\_ERR NOT\\_SUPPORTED](#) (-7)

### 8.92.1 Define Documentation

#### 8.92.1.1 #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.

#### 8.92.1.2 #define PVSOCK\_ERR\_NOT\_IMPLEMENTED (-6)

#### 8.92.1.3 #define PVSOCK\_ERR\_NOT\_SUPPORTED (-7)

#### 8.92.1.4 #define PVSOCK\_ERR\_SERV\_NOT\_CONNECTED (-4)

#### 8.92.1.5 #define PVSOCK\_ERR SOCK\_NO\_SERV (-3)

#### 8.92.1.6 #define PVSOCK\_ERR SOCK\_NOT\_CONNECTED (-5)

#### 8.92.1.7 #define PVSOCK\_ERR SOCK\_NOT\_OPEN (-2)

## 8.93 oscl\_socket\_listen.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclListenMethod](#)
- class [OsclListenRequest](#)

### Defines

- #define [OSCL\\_SOCKET\\_LISTEN\\_H\\_INCLUDEDd](#)

#### 8.93.1 Define Documentation

##### 8.93.1.1 #define OSCL\_SOCKET\_LISTEN\_H\_INCLUDEDd

## 8.94 oscl\_socket\_method.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_socket_request.h"
#include "pvlogger.h"
#include "oscl_socket_tuneables.h"
#include "oscl_ip_socket.h"
#include "oscl_socket_imp.h"
```

### Data Structures

- class [OsclSocketMethod](#)
- class [OsclSocketRequestAO](#)

### Defines

- #define [MSEC\\_TO\\_MICROSEC](#) 1000

#### 8.94.1 Define Documentation

##### 8.94.1.1 #define MSEC\_TO\_MICROSEC 1000

## 8.95 oscl\_socket\_recv.h File Reference

```
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclRecvMethod](#)
- class [OsclRecvRequest](#)

## 8.96 oscl\_socket\_recv\_from.h File Reference

```
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclRecvFromMethod](#)
- class [OsclRecvFromRequest](#)

## 8.97 oscl\_socket\_request.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_tuneables.h"
```

### Data Structures

- class [AcceptParam](#)
- class [BindParam](#)
- class [ConnectParam](#)
- class [ListenParam](#)
- class [OsclSocketRequest](#)
- class [PVSockBufRecv](#)
- class [PVSockBufSend](#)
- class [RecvFromParam](#)
- class [RecvParam](#)
- class [SendParam](#)
- class [SendToParam](#)
- class [ShutdownParam](#)
- class [SocketRequestParam](#)

## 8.98 oscl\_socket\_send.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclSendMethod](#)
- class [OsclSendRequest](#)

## **8.99 oscl\_socket\_send\_to.h File Reference**

```
#include "oscl_socket_types.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### **Data Structures**

- class [OsclSendToMethod](#)
- class [OsclSendToRequest](#)

## 8.100 oscl\_socket\_serv\_imp.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_tuneables.h"
#include "oscl_socket_serv_imp_pv.h"
```

## **8.101 oscl\_socket\_serv\_imp\_base.h File Reference**

```
#include "oscl_base.h"
#include "oscl_socket_stats.h"
```

### **Data Structures**

- class [OsclSocketServIBase](#)

## 8.102 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

### 8.102.1 Define Documentation

#### 8.102.1.1 #define OSCL\_EXCEPTSET\_FLAG 0x01

#### 8.102.1.2 #define OSCL\_READSET\_FLAG 0x04

A bitmask for socket select operations

#### 8.102.1.3 #define OSCL\_WRITESET\_FLAG 0x02

## 8.103 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](#)

## 8.104 oscl\_socket\_shutdown.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclShutdownMethod](#)
- class [OsclShutdownRequest](#)

## 8.105 oscl\_socket\_stats.h File Reference

```
#include "oscl_base.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
#include "oscl_socket_tuneables.h"
```

### Enumerations

- enum TOsclSocketStatEvent { EOscSocket\_RequestAO\_Success, EOscSocket\_RequestAO\_Canceled, EOscSocket\_RequestAO\_Error, EOscSocket\_RequestAO\_Timeout, EOscSocket\_ServRequestIssued, EOscSocket\_ServPoll, EOscSocket\_OS, EOscSocket\_Readable, EOscSocket\_Writable, EOscSocket\_Except, EOscSocket\_DataRecv, EOscSocket\_DataSent, EOscSocket\_ServRequestComplete, EOscSocket\_ServRequestCancelIssued, EOscSocketServ\_LoopsockOk, EOscSocketServ\_LoopsockError }
- enum TOsclSocketServStatEvent { EOscSocketServ\_SelectNoActivity = 0, EOscSocketServ\_SelectActivity, EOscSocketServ\_SelectRescheduleAsap, EOscSocketServ\_SelectReschedulePoll, EOscSocketServ\_LastEvent }

#### 8.105.1 Enumeration Type Documentation

##### 8.105.1.1 enum TOsclSocketServStatEvent

###### Enumeration values:

**EOscSocketServ\_SelectNoActivity**  
**EOscSocketServ\_SelectActivity**  
**EOscSocketServ\_SelectRescheduleAsap**  
**EOscSocketServ\_SelectReschedulePoll**  
**EOscSocketServ\_LastEvent**

##### 8.105.1.2 enum TOsclSocketStatEvent

Socket diagnostics.

###### Enumeration values:

**EOscSocket\_RequestAO\_Success**  
**EOscSocket\_RequestAO\_Canceled**  
**EOscSocket\_RequestAO\_Error**  
**EOscSocket\_RequestAO\_Timeout**  
**EOscSocket\_ServRequestIssued**  
**EOscSocket\_ServPoll**  
**EOscSocket\_OS**  
**EOscSocket\_Readable**  
**EOscSocket\_Writable**

---

**EOselSocket\_Except**  
**EOselSocket\_DataRecv**  
**EOselSocket\_DataSent**  
**EOselSocket\_ServRequestComplete**  
**EOselSocket\_ServRequestCancelIssued**  
**EOselSocketServ\_LoopsockOk**  
**EOselSocketServ\_LoopsockError**

## 8.106 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

### 8.106.1 Define Documentation

#### 8.106.1.1 #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.

#### 8.106.1.2 #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.

#### 8.106.1.3 #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.

#### 8.106.1.4 #define PV\_SOCKET\_REQUEST\_AO\_PRIORITY OsclActiveObject::EPriority-Nominal

PV\_SOCKET\_REQUEST\_AO\_PRIORITY sets the priority of the socket request completion AOs.

**8.106.1.5 #define PV\_SOCKET\_SERVER 1**

Enable/disable the PV socket server here.

**8.106.1.6 #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.

**8.106.1.7 #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.

**8.106.1.8 #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

**8.106.1.9 #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.

**8.106.1.10 #define PV\_SOCKET\_SERVER\_SELECT\_LOOPBACK\_SOCKET 0**

PV\_SOCKET\_SERVER\_SELECT\_LOOPBACK\_SOCKET enables the feature to wakeup the select call by writing to a loopback socket each time a new request comes in. This option is required to support the blocking select option of threaded server mode. This option is forced to "0" in AO mode.

**8.106.1.11 #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.

**8.106.1.12 #define PV\_SOCKET\_SERVER\_THREAD\_PRIORITY ThreadPriorityAboveNormal**

PV\_SOCKET\_SERVER\_THREAD\_PRIORITY sets the priority of the PV socket server thread.

**8.106.1.13 #define PV\_SOCKET\_SERVI\_STATS 0**

For detailed performance breakdown of time spend in [OsclSocketServI](#) AO. Output is logged under "OsclSchedulerPerfStats" node. Should be off in production code. This option is forced to "0" in threaded mode.

## 8.107 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 [OsclIpMReq](#)
- class [OsclNetworkAddress](#)
- class [OsclSocketObserver](#)
- class [OsclSocketTOS](#)

### 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](#) { EPVIMulticastTTL, EPVIPAddMembership, EPVIPTOS, EPVSockReuseAddr }
- enum [TPVSocketOptionLevel](#) { EPVIPProtoIP, EPVIPProtoTCP, EPVSocket }

#### 8.107.1 Define Documentation

##### 8.107.1.1 #define PVNETWORKADDRESS\_LEN 50

#### 8.107.2 Enumeration Type Documentation

##### 8.107.2.1 enum TPVSocketEvent

Return codes for asynchronous APIs

**Enumeration values:**

**EPVSocketSuccess**  
**EPVSocketPending**  
**EPVSocketTimeout**  
**EPVSocketFailure**

**EPVSocketCancel**  
**EPVSocketNotImplemented**

#### **8.107.2.2 enum TPVSocketFxn**

**Enumeration values:**

**EPVSocketSend**  
**EPVSocketSendTo**  
**EPVSocketRecv**  
**EPVSocketRecvFrom**  
**EPVSocketConnect**  
**EPVSocketAccept**  
**EPVSocketShutdown**  
**EPVSocketBind**  
**EPVSocketListen**  
**EPVSocket\_Last**

#### **8.107.2.3 enum TPVSocketOptionLevel**

**Enumeration values:**

**EPVIPProtoIP**  
**EPVIPProtoTCP**  
**EPVSocket**

#### **8.107.2.4 enum TPVSocketOptionName**

**Enumeration values:**

**EPVIMulticastTTL**  
**EPVIAAddMembership**  
**EPVIPTOS**  
**EPVSockReuseAddr**

#### **8.107.2.5 enum TPVSocketShutdown**

**Enumeration values:**

**EPVSocketSendShutdown**  
**EPVSocketRecvShutdown**  
**EPVSocketBothShutdown**

## 8.108 oscl\_stdstring.h File Reference

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

```
#include "oscl_base.h"
```

### Functions

- OSCL\_IMPORT\_REF uint32 `oscl_strlen` (const char \*str)
- OSCL\_IMPORT\_REF uint32 `oscl_strlen` (const `oscl_wchar` \*str)
- OSCL\_IMPORT\_REF char \* `oscl_strncpy` (char \*dest, const char \*src, uint32 count)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strncpy` (`oscl_wchar` \*dest, const `oscl_wchar` \*src, uint32 count)
- OSCL\_IMPORT\_REF int32 `oscl_strcmp` (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF int32 `oscl_stremp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF int32 `oscl_strncmp` (const char \*str1, const char \*str2, uint32 count)
- OSCL\_IMPORT\_REF int32 `oscl_strnncmp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2, uint32 count)
- OSCL\_IMPORT\_REF char \* `oscl_strncat` (char \*dest, const char \*src, uint32 count)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strncat` (`oscl_wchar` \*dest, const `oscl_wchar` \*src, uint32 count)
- OSCL\_IMPORT\_REF const char \* `oscl_strchr` (const char \*str, int32 c)
- OSCL\_IMPORT\_REF char \* `oscl_strchr` (char \*str, int32 c)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `oscl_strchr` (const `oscl_wchar` \*str, int32 c)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strchr` (`oscl_wchar` \*str, int32 c)
- OSCL\_IMPORT\_REF const char \* `oscl strrchr` (const char \*str, int32 c)
- OSCL\_IMPORT\_REF char \* `oscl strrchr` (char \*str, int32 c)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `oscl strrchr` (const `oscl_wchar` \*str, int32 c)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl strrchr` (`oscl_wchar` \*str, int32 c)
- OSCL\_IMPORT\_REF char \* `oscl_strset` (char \*dest, char val, uint32 count)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strset` (`oscl_wchar` \*dest, `oscl_wchar` val, uint32 count)
- OSCL\_IMPORT\_REF int32 `oscl_CIstrcmp` (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF int32 `oscl_CIstrcmp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF int32 `oscl_CIstrncmp` (const char \*str1, const char \*str2, uint32 count)
- OSCL\_IMPORT\_REF int32 `oscl_CIstrncmp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2, uint32 count)
- OSCL\_IMPORT\_REF char `oscl_tolower` (const char car)
- OSCL\_IMPORT\_REF `oscl_wchar` `oscl_tolower` (const `oscl_wchar` car)
- OSCL\_IMPORT\_REF bool `oscl_isLetter` (const char car)
- OSCL\_IMPORT\_REF const char \* `oscl_strstr` (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF char \* `oscl_strstr` (char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `oscl_strstr` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strstr` (`oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF char \* `oscl_streat` (char \*dest, const char \*src)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_streat` (`oscl_wchar` \*dest, const `oscl_wchar` \*src)

### **8.108.1 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.

## 8.109 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

#### 8.109.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

## 8.110 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](#) }

#### 8.110.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

## 8.111 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](#)

#### 8.111.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

## **8.112 oscl\_string\_rep.h File Reference**

Contains some internal implementation for string containers.

```
#include "oscl_defalloc.h"
```

### **Data Structures**

- class [CFastRep](#)
- class [CHheapRep](#)
- class [CStackRep](#)

#### **8.112.1 Detailed Description**

Contains some internal implementation for string containers.

## 8.113 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*

### 8.113.1 Detailed Description

Utilities to unescape URIs.

## 8.114 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.*

### 8.114.1 Detailed Description

Utilities to validate and truncate UTF-8 encoded strings.

## 8.115 oscl\_string\_utils.h File Reference

Utilities to parse and convert strings.

```
#include "oscl_base.h"
```

### Defines

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

### Functions

- OSCL\_IMPORT\_REF const char \* `skip_whitespace` (const char \*ptr)
- OSCL\_IMPORT\_REF char \* `skip_whitespace` (char \*ptr)
- OSCL\_IMPORT\_REF const char \* `skip_whitespace` (const char \*start, const char \*end)
- OSCL\_IMPORT\_REF const char \* `skip_to_whitespace` (const char \*start, const char \*end)
- OSCL\_IMPORT\_REF const char \* `skip_to_line_term` (const char \*start\_ptr, const char \*end\_ptr)
- OSCL\_IMPORT\_REF const char \* `skip_whitespace_and_line_term` (const char \*start, const char \*end)
- OSCL\_IMPORT\_REF int `extract_string` (const char \*in\_ptr, char \*outstring, int maxsize)
- OSCL\_IMPORT\_REF int `extract_string` (const char \*start, const char \*end, char \*outstring, int maxsize)
- OSCL\_IMPORT\_REF bool `PV_atoi` (const char \*buf, const char new\_format, uint32 &value)
- OSCL\_IMPORT\_REF bool `PV_atoi` (const char \*buf, const char new\_format, int length, uint32 &value)
- OSCL\_IMPORT\_REF bool `PV_atoi` (const char \*buf, const char new\_format, int length, `uint64` &value)
- OSCL\_IMPORT\_REF bool `PV_atof` (const char \*buf, `OsclFloat` &value)
- OSCL\_IMPORT\_REF bool `PV_atof` (const char \*buf, int length, `OsclFloat` &value)
- OSCL\_IMPORT\_REF int `oscl_abs` (int aVal)

### 8.115.1 Detailed Description

Utilities to parse and convert strings.

## 8.116 oscl\_string\_xml.h File Reference

Utilities to escape special characters in XML strings.

```
#include "oscl_base.h"
```

### Functions

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

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

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

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

### 8.116.1 Detailed Description

Utilities to escape special characters in XML strings.

## 8.117 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](#)

#### 8.117.1 Detailed Description

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

#### 8.117.2 Define Documentation

##### 8.117.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

## **8.118 oscl\_tcp\_socket.h File Reference**

```
#include "oscl_ip_socket.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_listen.h"
#include "oscl_socket_recv.h"
#include "oscl_socket_send.h"
#include "oscl_socket_accept.h"
#include "oscl_socket_shutdown.h"
#include "oscl_socket_connect.h"
#include "oscl_socket_bind.h"
```

### **Data Structures**

- class [OsclTCPSocketI](#)

## 8.119 oscl\_thread.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_procstatus.h"
#include "oscl_base.h"
```

### Data Structures

- class [OsclThread](#)

### Typedefs

- typedef [TOsclThreadFuncRet\(OSCL\\_THREAD DECL \\*](#) [TOsclThreadFuncPtr](#))[\(TOsclThreadFuncArg\)](#)

### Enumerations

- enum [OsclThread\\_State](#) { [Start\\_on\\_creation](#), [Suspend\\_on\\_creation](#) }
- enum [OsclThreadPriority](#) { [ThreadPriorityLowest](#), [ThreadPriorityLow](#), [ThreadPriorityBelowNormal](#), [ThreadPriorityNormal](#), [ThreadPriorityAboveNormal](#), [ThreadPriorityHighest](#), [ThreadPriorityTimeCritical](#) }
- enum [TOsclThreadTerminate](#) { [EOsclThreadTerminate\\_Join](#), [EOsclThreadTerminate\\_Kill](#), [EOsclThreadTerminate\\_NOP](#) }

### 8.119.1 Detailed Description

. This file provides THREAD implementation that can be ported  
to three OS LINUX, SYMBIAN, WIN32

### 8.119.2 Typedef Documentation

#### 8.119.2.1 [typedef TOsclThreadFuncRet\(OSCL\\_THREAD DECL \\*](#) [TOsclThreadFuncPtr](#))[\(TOsclThreadFuncArg\)](#)

### 8.119.3 Enumeration Type Documentation

#### 8.119.3.1 enum [OsclThread\\_State](#)

Enumeration values:

[Start\\_on\\_creation](#)  
[Suspend\\_on\\_creation](#)

#### 8.119.3.2 enum [OsclThreadPriority](#)

Enumeration values:

[ThreadPriorityLowest](#)

**ThreadPriorityLow**  
**ThreadPriorityBelowNormal**  
**ThreadPriorityNormal**  
**ThreadPriorityAboveNormal**  
**ThreadPriorityHighest**  
**ThreadPriorityTimeCritical**

#### 8.119.3.3 enum TOsclThreadTerminate

Enumeration values:

**EOsclThreadTerminate\_Join**  
**EOsclThreadTerminate\_Kill**  
**EOsclThreadTerminate\_NOP**

## 8.120 oscl\_tickcount.h File Reference

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

```
#include "oscl_base.h"
#include "oscl_tickcount.inl"
```

### Data Structures

- class [OsclTickCount](#)

### Defines

- #define [OSCLTICKCOUNT\\_MAX\\_TICKS](#) 0xffffffff

### 8.120.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

## 8.121 oscl\_time.h File Reference

The file `oscl_time.h` defines two classes `NTPTime` and `TimeValue` for getting, manipulating, and formatting time values. The `TimeValue` class is based on the native system time format while `NTPTime` is used for the standard Network Time Protocol format.

```
#include "oscl_base.h"
#include "osclconfig_time.h"
#include "oscl_int64_utils.h"
#include "oscl_time.inl"
```

### Data Structures

- class `NTPTime`

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

- class `TimeValue`

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

### Typedefs

- typedef char `CtimeStrBuf` [`CTIME_BUFFER_SIZE`]
- typedef char `PV8601timeStrBuf` [`PV8601TIME_BUFFER_SIZE`]
- 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

### 8.121.1 Detailed Description

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

## 8.122 oscl\_timer.h File Reference

```
#include "oscl_base.h"
#include "osclconfig_util.h"
#include "oscl_vector.h"
#include "oscl_tickcount.h"
#include "oscl_rand.h"
#include "oscl_scheduler_ao.h"
```

### Data Structures

- struct [\\_TimerEntry](#)
- class [CallbackTimer](#)
- class [CallbackTimerObserver](#)
- class [OsclTimer](#)
- class [OsclTimerObserver](#)

## 8.123 oscl\_tls.h File Reference

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

### Data Structures

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

### Defines

- #define [OSCL\\_TLS\\_BASE\\_SLOTS](#) OSCL\_TLS\_ID\_BASE\_LAST +1
- #define [OSCL\\_TLS\\_EXTERNAL\\_SLOTS](#) 0
- #define [OSCL\\_TLS\\_MAX\\_SLOTS](#) ( OSCL\_TLS\_BASE\_SLOTS + OSCL\_TLS\_EXTERNAL\_SLOTS)

### Typedefs

- typedef [OsclAny](#) TOsclTlsKey

### Variables

- const uint32 [OSCL\\_TLS\\_ID\\_MAGICNUM](#) = 0
- const uint32 [OSCL\\_TLS\\_ID\\_ERRORHOOK](#) = 1
- const uint32 [OSCL\\_TLS\\_ID\\_PVLOGGER](#) = 2
- const uint32 [OSCL\\_TLS\\_ID\\_TEST](#) = 3
- const uint32 [OSCL\\_TLS\\_ID\\_PVSCHEDULER](#) = 4
- const uint32 [OSCL\\_TLS\\_ID\\_PVERRORTRAP](#) = 5
- const uint32 [OSCL\\_TLS\\_ID\\_SDPMEDIAPARSER](#) = 6
- const uint32 [OSCL\\_TLS\\_ID\\_PAYLOADPARSER](#) = 7
- const uint32 [OSCL\\_TLS\\_ID\\_PVMFRECOGNIZER](#) = 8
- const uint32 [OSCL\\_TLS\\_ID\\_WMDRM](#) = 9
- const uint32 [OSCL\\_TLS\\_ID\\_OSCLREGISTRY](#) = 10
- const uint32 [OSCL\\_TLS\\_ID\\_SQLITE3](#) = 11
- const uint32 [OSCL\\_TLS\\_ID\\_BASE\\_LAST](#) = 11

## 8.124 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](#)

#### 8.124.1 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.

#### 8.124.2 Define Documentation

##### 8.124.2.1 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

## 8.125 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*

### 8.125.1 Detailed Description

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

## 8.126 oscl\_udp\_socket.h File Reference

```
#include "oscl_ip_socket.h"
#include "oscl_defalloc.h"
#include "oscl_socket_recv_from.h"
#include "oscl_socket_send_to.h"
#include "oscl_socket_bind.h"
```

### Data Structures

- class [OsclUDPSocketI](#)

## 8.127 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.*

### 8.127.1 Detailed Description

Utilities to convert unicode to utf8 and vice versa.

## 8.128 oscl\_uuid.h File Reference

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32.

```
#include "oscl_base_macros.h"  
#include "oscl_mem_basic_functions.h"
```

### Data Structures

- struct [OsclUuid](#)

### Defines

- #define [EMPTY\\_UUID](#) PVUuid(0,0,0,0,0,0,0,0)
- #define [BYTES\\_IN\\_UUID\\_ARRAY](#) 8

### Typedefs

- typedef uint32 [OsclUid32](#)

#### 8.128.1 Detailed Description

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32.

#### 8.128.2 Define Documentation

##### 8.128.2.1 #define [BYTES\\_IN\\_UUID\\_ARRAY](#) 8

##### 8.128.2.2 #define [EMPTY\\_UUID](#) PVUuid(0,0,0,0,0,0,0,0)

#### 8.128.3 Typedef Documentation

##### 8.128.3.1 typedef uint32 [OsclUid32](#)

## 8.129 oscl\_uuid\_utils.h File Reference

```
#include "oscl_string_utils.h"
#include "oscl_stdstring.h"
```

### Variables

- const char **PV\_CHAR\_CLOSE\_BRACKET** = ')
- const char **PV\_CHAR\_COMMA** = ','

#### 8.129.1 Detailed Description

#### 8.129.2 Variable Documentation

8.129.2.1 const char **PV\_CHAR\_CLOSE\_BRACKET** = ')

8.129.2.2 const char **PV\_CHAR\_COMMA** = ','

## 8.130 oscl\_vector.h File Reference

The file `oscl_vector.h` defines the template class `Oscl_Vector` which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_mem_basic_functions.h"
#include "oscl_assert.h"
#include "oscl_opaque_type.h"
#include "oscl_defalloc.h"
#include "oscl_base.h"
```

### Data Structures

- class `Oscl_Vector`
- class `Oscl_Vector_Base`

#### 8.130.1 Detailed Description

The file `oscl_vector.h` defines the template class `Oscl_Vector` which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

## 8.131 osclconfig.h File Reference

This file contains configuration information for the linux platform.

```
#include <dirent.h>
#include <dlsfcn.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

### 8.131.1 Detailed Description

This file contains configuration information for the linux platform.

## 8.131.2 Define Documentation

8.131.2.1 `#define __TFS__ <>`

8.131.2.2 `#define OSCL_EXPORT_REF __attribute__ ((visibility("default")))`

8.131.2.3 `#define OSCL_HAS_ANDROID_FILE_IO_SUPPORT 1`

8.131.2.4 `#define OSCL_HAS_ANDROID_SUPPORT 1`

8.131.2.5 `#define OSCL_HAS_PACKED_STRUCT 1`

8.131.2.6 `#define OSCL_IMPORT_REF __attribute__ ((visibility("default")))`

8.131.2.7 `#define OSCL_NATIVE_UINT64_TYPE u_int64_t`

8.131.2.8 `#define OSCL_PACKED_STRUCT_BEGIN`

8.131.2.9 `#define OSCL_PACKED_STRUCT_END __attribute__((packed))`

8.131.2.10 `#define OSCL_PACKED_VAR(x) x __attribute__((packed))`

8.131.2.11 `#define OSCL_RELEASE_BUILD 0`

8.131.2.12 `#define OSCL_TEMPLATED_DESTRUCTOR_CALL(type, simple_type) ~type()`

8.131.2.13 `#define OSCL_UNSIGNED_CONST(x) x##u`

8.131.2.14 `#define PVLOGGER_INST_LEVEL 5`

## 8.132 osclconfig\_ansi\_memory.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include <memory.h>
```

### Defines

- #define OSCL\_HAS\_ANSI\_MEMORY\_FUNCS 1

### Typedefs

- typedef size\_t oscl\_memsize\_t

#### 8.132.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

#### 8.132.2 Define Documentation

8.132.2.1 #define OSCL\_HAS\_ANSI\_MEMORY\_FUNCS 1

#### 8.132.3 Typedef Documentation

8.132.3.1 typedef size\_t oscl\_memsize\_t

## 8.133 osclconfig\_check.h File Reference

### Typedefs

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

## 8.134 osclconfig\_compiler\_warnings.h File Reference

This file contains the ability to turn off/on compiler warnings.

### Defines

- #define OSCL\_FUNCTION\_PTR(x) (&x)

#### 8.134.1 Detailed Description

This file contains the ability to turn off/on compiler warnings.

#### 8.134.2 Define Documentation

##### 8.134.2.1 #define OSCL\_FUNCTION\_PTR(x) (&x)

## 8.135 osclconfig\_error.h File Reference

This file contains the common typedefs and header files needed to compile osclerror.

```
#include "osclconfig.h"  
#include <setjmp.h>  
#include <errno.h>  
#include "osclconfig_error_check.h"
```

### Defines

- #define OSCL\_HAS\_EXCEPTIONS 1
- #define OSCL\_HAS\_ERRNO\_H 1
- #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0
- #define OSCL\_HAS\_SETJMP\_H 1

### 8.135.1 Detailed Description

This file contains the common typedefs and header files needed to compile osclerror.

### 8.135.2 Define Documentation

- 8.135.2.1 #define OSCL\_HAS\_ERRNO\_H 1
- 8.135.2.2 #define OSCL\_HAS\_EXCEPTIONS 1
- 8.135.2.3 #define OSCL\_HAS\_SETJMP\_H 1
- 8.135.2.4 #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0

## **8.136 osclconfig\_error\_check.h File Reference**

## **8.137 osclconfig\_global\_new\_delete.h File Reference**

### **Functions**

- void \* **operator new** (size\_t)
- void **operator delete** (void \*)

## 8.138 osclconfig\_global\_placement\_new.h File Reference

### Functions

- void \* [operator new](#) (size\_t, void \*ptr)

#### 8.138.1 Function Documentation

**8.138.1.1 void\* operator new (size\_t, void \*ptr) [inline]**

## 8.139 osclconfig\_io.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include "osclconfig.h"  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <stdarg.h>  
#include <sys/socket.h>  
#include <netinet/in.h>  
#include <arpa/inet.h>  
#include <fcntl.h>  
#include <signal.h>  
#include <netdb.h>  
#include <sys/mman.h>  
#include <sys/types.h>  
#include <errno.h>  
#include <sys/vfs.h>  
#include <dirent.h>  
#include <sys/stat.h>  
#include "osclconfig_io_check.h"
```

### Defines

- #define OSCL\_HAS\_GLOB 0
- #define OSCL\_HAS\_ANSI\_FILE\_IO\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_64BIT\_FILE\_IO\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_FILE\_IO\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION 0
- #define OSCL\_HAS\_NATIVE\_FILE\_CACHE\_ENABLE 1
- #define OSCL\_FILE\_BUFFER\_MAX\_SIZE 32768
- #define OSCL\_HAS\_PV\_FILE\_CACHE 0
- #define OSCL\_HAS\_LARGE\_FILE\_SUPPORT 1
- #define OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER 0
- #define OSCL\_HAS\_SYMBIAN\_DNS\_SERVER 0
- #define OSCL\_HAS\_BERKELEY\_SOCKETS 1
- #define OSCL\_HAS\_SOCKET\_SUPPORT 1
- #define OsclValidInetAddr(addr) (inet\_addr(addr)!=INADDR\_NONE)
- #define OsclMakeSockAddr(sockaddr, port, addrstr, ok)
- #define OsclUnMakeSockAddr(sockaddr, addrstr) addrstr=inet\_ntoa(sockaddr.sin\_addr);
- #define OsclMakeInAddr(in\_addr, addrstr, ok)
- #define OsclUnMakeInAddr(in\_addr, addrstr) addrstr=inet\_ntoa(in\_addr);
- #define OsclSetRecvBufferSize(s, val, ok, err)
- #define OsclBind(s, addr, ok, err)

- #define **OsclSetSockOpt**(s, optLevel, optName, optVal, optLen, ok, err)
- #define **OsclJoin**(s, addr, ok, err)
- #define **OsclListen**(s, size, ok, err)
- #define **OsclAccept**(s, accept\_s, ok, err, wouldblock)
- #define **OsclSetNonBlocking**(s, ok, err)
- #define **OsclShutdown**(s, how, ok, err)
- #define **OsclSocket**(s, fam, type, prot, ok, err)
- #define **OsclSendTo**(s, buf, len, addr, ok, err, nbytes, wouldblock)
- #define **OsclSend**(s, buf, len, ok, err, nbytes, wouldblock)
- #define **OsclCloseSocket**(s, ok, err)
- #define **OsclConnect**(s, addr, ok, err, wouldblock)
- #define **OsclGetPeerName**(s, name, namelen, ok, err)
- #define **OsclGetAsyncSockErr**(s, ok, err)
- #define **OsclPipe**(x) pipe(x)
- #define **OsclReadFD**(fd, buf, cnt) read(fd,buf,cnt)
- #define **OsclWriteFD**(fd, buf, cnt) write(fd,buf,cnt)
- #define **OsclConnectComplete**(s, wset, eset, success, fail, ok, err)
- #define **OsclRecv**(s, buf, len, ok, err, nbytes, wouldblock)
- #define **OsclRecvFrom**(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)
- #define **OsclSocketSelect**(nfds, rd, wr, ex, timeout, ok, err, nhandles)
- #define **OsclSocketStartup**(ok)
- #define **OsclSocketCleanup**(ok)
- #define **OsclGethostbyname**(name, hostent, ok, err)
- #define **OsclGetDottedAddr**(hostent, dottedaddr, ok)
- #define **OsclGetDottedAddrVector**(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

## Typedefs

- typedef int **TOsclSocket**
- typedef sockaddr\_in **TOsclSockAddr**
- typedef socklen\_t **TOsclSockAddrLen**
- typedef ip\_mreq **TIpMReq**
- typedef hostent **TOsclHostent**
- typedef off64\_t **TOsclFileOffset**

### **8.139.1 Detailed Description**

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.



## 8.139.2 Define Documentation

- 8.139.2.1 `#define OSCL_AF_INET AF_INET`
- 8.139.2.2 `#define OSCL_FILE_BUFFER_MAX_SIZE 32768`
- 8.139.2.3 `#define OSCL_HAS_ANSI_64BIT_FILE_IO_SUPPORT 0`
- 8.139.2.4 `#define OSCL_HAS_ANSI_FILE_IO_SUPPORT 1`
- 8.139.2.5 `#define OSCL_HAS_BERKELEY_SOCKETS 1`
- 8.139.2.6 `#define OSCL_HAS_GLOB 0`
- 8.139.2.7 `#define OSCL_HAS_LARGE_FILE_SUPPORT 1`
- 8.139.2.8 `#define OSCL_HAS_MSWIN_FILE_IO_SUPPORT 0`
- 8.139.2.9 `#define OSCL_HAS_NATIVE_FILE_CACHE_ENABLE 1`
- 8.139.2.10 `#define OSCL_HAS_PV_FILE_CACHE 0`
- 8.139.2.11 `#define OSCL_HAS_SOCKET_SUPPORT 1`
- 8.139.2.12 `#define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0`
- 8.139.2.13 `#define OSCL_HAS_SYMBIAN_DNS_SERVER 0`
- 8.139.2.14 `#define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0`
- 8.139.2.15 `#define OSCL IPPROTO_IP IPPROTO_IP`
- 8.139.2.16 `#define OSCL IPPROTO_TCP IPPROTO_TCP`
- 8.139.2.17 `#define OSCL IPPROTO_UDP IPPROTO_UDP`
- 8.139.2.18 `#define OSCL_SD_BOTH SHUT_RDWR`
- 8.139.2.19 `#define OSCL_SD_RECEIVE SHUT_RD`
- 8.139.2.20 `#define OSCL_SD_SEND SHUT_WR`
- 8.139.2.21 `#define OSCL SOCK_DGRAM SOCK_DGRAM`
- 8.139.2.22 `#define OSCL SOCK_STREAM SOCK_STREAM`
- 8.139.2.23 `#define OSCL_SOCKOPT_IP_ADDMEMBERSHIP IP_ADD_MEMBERSHIP`
- 8.139.2.24 `#define OSCL_SOCKOPT_IP_MULTICAST_TTL IP_MULTICAST_TTL`
- 8.139.2.25 `#define OSCL_SOCKOPT_IP_TOS IP_TOS`
- 8.139.2.26 `#define OSCL_SOCKOPT_SOL_REUSEADDR SO_REUSEADDR`
- 8.139.2.27 `#define OSCL_SOL_IP IPPROTO_IP`

```
accept_s=accept(s,NULL,NULL);\
ok=(accept_s!=(-1));\
if (!ok){err=errno;wouldblock=(err==EAGAIN)|err==EWOULDBLOCK);}
```

### 8.139.2.32 #define OsclBind(s, addr, ok, err)

**Value:**

```
TOsclSockAddr* tmpadr = &addr;\
sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\
ok=(bind(s,sadr,sizeof(addr))!=(-1));\
if (!ok)err=errno
```

### 8.139.2.33 #define OsclCloseSocket(s, ok, err)

**Value:**

```
ok=(close(s)!=(-1));\
if (!ok)err=errno
```

### 8.139.2.34 #define OsclConnect(s, addr, ok, err, wouldblock)

**Value:**

```
TOsclSockAddr* tmpadr = &addr;\
sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\
ok=(connect(s,sadr,sizeof(addr))!=(-1));\
if (!ok){err=errno;wouldblock=(err==EINPROGRESS);}
```

### 8.139.2.35 #define OsclConnectComplete(s, wset, eset, success, fail, ok, err)

**Value:**

```
success=fail=false;\
if (FD_ISSET(s,&eset))\
{fail=true;OsclGetAsyncSockErr(s,ok,err);}\\
else if (FD_ISSET(s,&wset))\
{OsclGetAsyncSockErr(s,ok,err);if (ok && err==0)success=true;else fail=true;}
```

### 8.139.2.36 #define OsclGetAsyncSockErr(s, ok, err)

**Value:**

```
int opterr;socklen_t optlen(sizeof(opterr));\
ok=(getsockopt(s,SOL_SOCKET,SO_ERROR,(void *)&opterr,&optlen)!=(-1));\
if(ok)err=opterr;else err=errno;
```

**8.139.2.37 #define OsclGetDottedAddr(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);
```

**8.139.2.38 #define OsclGetDottedAddrVector(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];\
    OsclNetworkAddress addr(inet_ntoa(_inaddr), 0);\
    dottedaddrvect->push_back(addr);\
  }\\
  if (!dottedaddrvect->empty())\
    {dottedaddr->port = dottedaddrvect->front().port; dottedaddr->ipAddr.Set(dottedaddrvect->front().ipAd\
ok=(!dottedaddrvect->empty() && (((*dottedaddrvect)[0]).ipAddr.Str() != NULL));\
}\\
else\
{\
  char *add;\
  OsclGetDottedAddr(hostent,add,ok);\
  if(ok) dottedaddr->ipAddr.Set(add);\
}
```

**8.139.2.39 #define OsclGethostbyname(name, hostent, ok, err)**

**Value:**

```
hostent=gethostbyname((const char*)name);\
ok=(hostent!=NULL);\
if (!ok)err=errno;
```

**8.139.2.40 #define OsclGetPeerName(s, name, namelen, ok, err)**

**Value:**

```
ok=(getpeername(s,(sockaddr*)&name,(socklen_t*)&namelen) != (-1) );\
if (!ok)err=errno
```

**8.139.2.41 #define OsclJoin(s, addr, ok, err)**

**Value:**

```
{
    struct ip_mreq mreq; \
    void* p = &addr; \
    ok=(bind(s,(sockaddr*)p,sizeof(addr))!=(-1)); \
    mreq.imr_multiaddr.s_addr = addr.sin_addr.s_addr ; \
    mreq.imr_interface.s_addr = htonl(INADDR_ANY); \
    ok=(setsockopt(s, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq, sizeof(struct ip_mreq))!=(-1)); \
    if (!ok)err=errno; \
}
```

#### 8.139.2.42 #define OsclListen(s, size, ok, err)

**Value:**

```
ok=(listen(iSocket,qSize)!=(-1)); \
if (!ok)err=errno
```

#### 8.139.2.43 #define OsclMakeInAddr(in\_addr, addrstr, ok)

**Value:**

```
int32 result = inet_aton((const char*)addrstr, &in_addr); \
ok=(result!=0);
```

#### 8.139.2.44 #define OsclMakeSockAddr(sockaddr, port, addrstr, ok)

**Value:**

```
sockaddr.sin_family=OSCL_AF_INET; \
sockaddr.sin_port=htons(port); \
int32 result=inet_aton((const char*)addrstr,&sockaddr.sin_addr); \
ok=(result!=0);
```

#### 8.139.2.45 #define OsclPipe(x) pipe(x)

#### 8.139.2.46 #define OsclReadFD(fd, buf, cnt) read(fd,buf,cnt)

#### 8.139.2.47 #define OsclRecv(s, buf, len, ok, err, nbytes, wouldblock)

**Value:**

```
nbytes=recv(s,(void *)buf),(size_t)(len),0); \
ok=(nbytes!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN);}
```

**8.139.2.48 #define OsclRecvFrom(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)**
**Value:**

```
\nvoid* p=paddr;\n\nbytes=recvfrom(s,(void*)(buf),(size_t)(len),0,(struct sockaddr*)p,paddrlen);\n  ok=(nbytes!=(-1));\n  if (!ok){err=errno;wouldblock=(err==EAGAIN);}\n}
```

**8.139.2.49 #define OsclSend(s, buf, len, ok, err, nbytes, wouldblock)**
**Value:**

```
nbytes=send(s,(const void*)(buf),(size_t)(len),0);\n  ok=(nbytes!=(-1));\n  if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

**8.139.2.50 #define OsclSendTo(s, buf, len, addr, ok, err, nbytes, wouldblock)**
**Value:**

```
TOsclSockAddr* tmpadr = &addr;\n  sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\n  nbytes=sendto(s,(const void*)(buf),(size_t)(len),0,sadr,(socklen_t)sizeof(addr));\n  ok=(nbytes!=(-1));\n  if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

**8.139.2.51 #define OsclSetNonBlocking(s, ok, err)**
**Value:**

```
ok=(fcntl(s,F_SETFL,O_NONBLOCK)!=(-1));\n  if (!ok)err=errno
```

**8.139.2.52 #define OsclSetRecvBufferSize(s, val, ok, err)**
**Value:**

```
ok=(setsockopt(s,SOL_SOCKET,SO_RCVBUF,(char*)&val, sizeof(int)) != -1);\n  if (!ok)err=errno
```

**8.139.2.53 #define OsclSetSockOpt(s, optLevel, optName, optVal, optLen, ok, err)**
**Value:**

```
ok=(setsockopt(s,optLevel,optName,OSCL_STATIC_CAST(const char*,optVal),optLen) != (-1));\n  if (!ok)err=errno
```

**8.139.2.54 #define OsclShutdown(s, how, ok, err)****Value:**

```
ok=(shutdown(iSocket,how)!=(-1));\
if (!ok)err=errno
```

**8.139.2.55 #define OsclSocket(s, fam, type, prot, ok, err)****Value:**

```
s=socket(fam,type,prot);\
ok=(s!=(-1));\
if (!ok)err=errno
```

**8.139.2.56 #define OsclSocketCleanup(ok)****Value:**

```
signal(SIGPIPE,SIG_DFL);\
ok=true
```

**8.139.2.57 #define OsclSocketSelect(nfds, rd, wr, ex, timeout, ok, err, nhandles)****Value:**

```
nhandles=select(nfds,&rd,&wr,&ex,&timeout);\
ok=(nhandles!=(-1));\
if (!ok)err=errno
```

**8.139.2.58 #define OsclSocketStartup(ok)****Value:**

```
signal(SIGPIPE,SIG_IGN);\
ok=true
```

8.139.2.59 #define OsclUnMakeInAddr(in\_addr, addrstr) addrstr=inet\_ntoa(in\_addr);

8.139.2.60 #define OsclUnMakeSockAddr(sockaddr, addrstr) addrstr=inet\_ntoa(sockaddr.sin\_addr);

8.139.2.61 #define OsclValidInetAddr(addr) (inet\_addr(addr)!=INADDR\_NONE)

8.139.2.62 #define OsclWriteFD(fd, buf, cnt) write(fd,buf,cnt)

### 8.139.3 Typedef Documentation

8.139.3.1 typedef struct ip\_mreq TIpMReq

8.139.3.2 typedef off64\_t TOsclFileOffset

8.139.3.3 typedef struct hostent TOsclHostent

8.139.3.4 typedef struct sockaddr\_in TOsclSockAddr

8.139.3.5 typedef socklen\_t TOsclSockAddrLen

8.139.3.6 typedef int TOsclSocket

## 8.140 osclconfig\_io\_check.h File Reference

### Typedefs

- `typedef TOsclFileOffset __verify__TOsclFileOffset__defined__`

#### 8.140.1 Typedef Documentation

##### 8.140.1.1 `typedef TOsclFileOffset __verify__TOsclFileOffset__defined__`

type `TOsclFileOffset` should be defined as the type used for file size and offsets on the target platform.  
Example: `typedef size_t TOsclFileOffset;`

## **8.141 osclconfig\_ix86.h File Reference**

This file contains configuration information for the ix86 processor family.

### **Defines**

- #define OSCL\_INTEGERS\_WORD\_ALIGNED 1
- #define OSCL\_BYTE\_ORDER\_BIG\_ENDIAN 0
- #define OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN 1

### **8.141.1 Detailed Description**

This file contains configuration information for the ix86 processor family.

## **8.142 osclconfig\_lib.h File Reference**

This file contains configuration information for the ANSI build.

```
#include "osclconfig_lib_check.h"
```

### **Defines**

- #define OSCL\_HAS\_RUNTIME\_LIB\_LOADING\_SUPPORT 1
- #define PV\_RUNTIME\_LIB\_FILENAME\_EXTENSION "so"
- #define OSCL\_LIB\_READ\_DEBUG\_LIBS 1
- #define PV\_DYNAMIC\_LOADING\_CONFIG\_FILE\_PATH "./"

### **8.142.1 Detailed Description**

This file contains configuration information for the ANSI build.

### **8.142.2 Define Documentation**

**8.142.2.1 #define OSCL\_HAS\_RUNTIME\_LIB\_LOADING\_SUPPORT 1**

**8.142.2.2 #define OSCL\_LIB\_READ\_DEBUG\_LIBS 1**

**8.142.2.3 #define PV\_DYNAMIC\_LOADING\_CONFIG\_FILE\_PATH "./"**

**8.142.2.4 #define PV\_RUNTIME\_LIB\_FILENAME\_EXTENSION "so"**

## **8.143 osclconfig\_lib\_check.h File Reference**

## **8.144 osclconfig\_limits\_typedefs.h File Reference**

This file contains common typedefs based on the ANSI C limits.h header.

```
#include <limits.h>
```

### **Defines**

- #define OSCL\_CHAR\_IS\_UNSIGNED 1
- #define OSCL\_CHAR\_IS\_SIGNED 0

### **8.144.1 Detailed Description**

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

### **8.144.2 Define Documentation**

**8.144.2.1 #define OSCL\_CHAR\_IS\_SIGNED 0**

**8.144.2.2 #define OSCL\_CHAR\_IS\_UNSIGNED 1**

## 8.145 osclconfig\_memory.h File Reference

```
#include "osclconfig.h"
#include "osclconfig_ansi_memory.h"
#include "osclconfig_memory_check.h"
```

### Defines

- #define OSCL\_BYPASS\_MEMMGT 1
- #define OSCL\_HAS\_GLOBAL\_NEW\_DELETE 1
- #define PVMEM\_INST\_LEVEL 1
- #define OSCL\_HAS\_HEAP\_BASE\_SUPPORT 1
- #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0

### 8.145.1 Define Documentation

8.145.1.1 #define OSCL\_BYPASS\_MEMMGT 1

8.145.1.2 #define OSCL\_HAS\_GLOBAL\_NEW\_DELETE 1

8.145.1.3 #define OSCL\_HAS\_HEAP\_BASE\_SUPPORT 1

8.145.1.4 #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0

8.145.1.5 #define PVMEM\_INST\_LEVEL 1

## **8.146 osclconfig\_memory\_check.h File Reference**

## 8.147 osclconfig\_no\_os.h File Reference

### Defines

- #define OSCL\_HAS\_UNIX\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_HAS\_SAVAJE\_SUPPORT 0
- #define OSCL\_HAS\_PV\_C\_OS\_SUPPORT 0
- #define OSCL\_HAS\_ANDROID\_SUPPORT 0
- #define OSCL\_HAS\_IPHONE\_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

## **8.148 osclconfig\_proc.h File Reference**

This file contains configuration information for the linux platform.

```
#include "osclconfig.h"  
#include "osclconfig_proc_unix_android.h"  
#include "osclconfig_proc_check.h"
```

### **8.148.1 Detailed Description**

This file contains configuration information for the linux platform.

## 8.149 osclconfig\_proc\_check.h File Reference

### Typedefs

- `typedef TOsclThreadId __verify__TOsclThreadId_defined__`
- `typedef TOsclThreadFuncRet __verify__TOsclThreadFuncRet_defined__`
- `typedef TOsclThreadFuncArg __verify__TOsclThreadFuncArg_defined__`
- `typedef TOsclThreadObject __verify__TOsclThreadObject_defined__`
- `typedef TOsclMutexObject __verify__TOsclMutexObject_defined__`
- `typedef TOsclSemaphoreObject __verify__TOsclSemaphoreObject_defined__`
- `typedef TOsclConditionObject __verify__TOsclConditionObject_defined__`

### 8.149.1 Typedef Documentation

#### 8.149.1.1 `typedef TOsclConditionObject __verify__TOsclConditionObject_defined__`

type `TOsclConditionObject` should be defined as the type used as a condition variable on the target platform.  
Example: `typedef pthread_cond_t TOsclConditionObject;`

Note: Condition variables are only used with certain semaphore implementations. If the semaphore implementation does not require a condition variable, then this type can be defined as 'int' as follows: `typedef int TOsclConditionObject; //not used`

#### 8.149.1.2 `typedef TOsclMutexObject __verify__TOsclMutexObject_defined__`

type `TOsclMutexObject` should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef pthread_mutex_t TOsclMutexObject;`

#### 8.149.1.3 `typedef TOsclSemaphoreObject __verify__TOsclSemaphoreObject_defined__`

type `TOsclSemaphoreObject` should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef sem_t TOsclSemaphoreObject;`

#### 8.149.1.4 `typedef TOsclThreadFuncArg __verify__TOsclThreadFuncArg_defined__`

type `TOsclThreadFuncArg` should be defined as the type used as a thread function argument on the target platform. Example: `typedef LPVOID TOsclThreadFuncArg;`

#### 8.149.1.5 `typedef TOsclThreadFuncRet __verify__TOsclThreadFuncRet_defined__`

type `TOsclThreadFuncRet` should be defined as the type used as a thread function return value on the target platform. Example: `typedef DWORD TOsclThreadFuncRet;`

#### 8.149.1.6 `typedef TOsclThreadId __verify__TOsclThreadId_defined__`

type `TOsclThreadId` should be defined as the type used as a thread ID on the target platform. Example: `typedef DWORD TOsclThreadId;`

**8.149.1.7 `typedef TOsclThreadObject __verify__TOsclThreadObject_defined__`**

type `TOsclThreadObject` should be defined as the type used as a thread object or handle on the target platform. Example: `typedef pthread_t TOsclThreadObject;`

## 8.150 osclconfig\_proc\_unix\_android.h File Reference

```
#include <pthread.h>
#include <errno.h>
#include <signal.h>
```

### Defines

- #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- #define OSCL\_HAS\_THREAD\_SUPPORT 1
- #define OSCL\_HAS\_NON\_PREEMPTIVE\_THREAD\_SUPPORT 0
- #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 0
- #define OSCL\_HAS\_PTHREAD\_SUPPORT 1
- #define OSCL\_THREAD\_DECL

### TypeDefs

- typedef pthread\_t TOsclThreadId
- typedef void \* TOsclThreadFuncArg
- typedef void \* TOsclThreadFuncRet
- typedef pthread\_t TOsclThreadObject
- typedef pthread\_mutex\_t TOsclMutexObject
- typedef int TOsclSemaphoreObject
- typedef pthread\_cond\_t TOsclConditionObject

### 8.150.1 Define Documentation

- 8.150.1.1 `#define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0`
- 8.150.1.2 `#define OSCL_HAS_PTHREAD_SUPPORT 1`
- 8.150.1.3 `#define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0`
- 8.150.1.4 `#define OSCL_HAS_SYMBIAN_SCHEDULER 0`
- 8.150.1.5 `#define OSCL_HAS_THREAD_SUPPORT 1`
- 8.150.1.6 `#define OSCL_THREAD_DECL`

### 8.150.2 Typedef Documentation

- 8.150.2.1 `typedef pthread_cond_t TOsclConditionObject`
- 8.150.2.2 `typedef pthread_mutex_t TOsclMutexObject`
- 8.150.2.3 `typedef int TOsclSemaphoreObject`
- 8.150.2.4 `typedef void* TOsclThreadFuncArg`
- 8.150.2.5 `typedef void* TOsclThreadFuncRet`
- 8.150.2.6 `typedef pthread_t TOsclThreadId`
- 8.150.2.7 `typedef pthread_t TOsclThreadObject`

## 8.151 osclconfig\_proc\_unix\_common.h File Reference

```
#include <time.h>
#include <semaphore.h>
#include <pthread.h>
#include <errno.h>
```

### Defines

- #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- #define OSCL\_HAS\_THREAD\_SUPPORT 1
- #define OSCL\_HAS\_NON\_PREEMPTIVE\_THREAD\_SUPPORT 0
- #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 1
- #define OSCL\_HAS\_PTHREAD\_SUPPORT 1
- #define OSCL\_THREAD\_DECL

### Typedefs

- typedef pthread\_t TOsclThreadId
- typedef void \* TOsclThreadFuncArg
- typedef void \* TOsclThreadFuncRet
- typedef pthread\_t TOsclThreadObject
- typedef pthread\_mutex\_t TOsclMutexObject
- typedef sem\_t TOsclSemaphoreObject
- typedef pthread\_cond\_t TOsclConditionObject

### 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 1
- 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 sem\_t 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\_time.h File Reference

```
#include "osclconfig.h"
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include "osclconfig_time_check.h"
```

### Defines

- #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 1

### TypeDefs

- typedef timeval OsclBasicTimeStruct
- typedef tm OsclBasicDateTimeStruct

#### 8.152.1 Define Documentation

8.152.1.1 #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 1

#### 8.152.2 Typedef Documentation

8.152.2.1 typedef tm OsclBasicDateTimeStruct

8.152.2.2 typedef struct timeval OsclBasicTimeStruct

## 8.153 osclconfig\_time\_check.h File Reference

### Typedefs

- `typedef OsclBasicTimeStruct __Validate__BasicTimeStruct__`
- `typedef OsclBasicDateTimeStruct __Validate__BasicTimeDateStruct__`

#### 8.153.1 Typedef Documentation

##### 8.153.1.1 `typedef OsclBasicDateTimeStruct __Validate__BasicTimeDateStruct__`

`OsclBasicDateTimeStruct` type should be defined to the platform-specific date + time type.

##### 8.153.1.2 `typedef OsclBasicTimeStruct __Validate__BasicTimeStruct__`

`OsclBasicTimeStruct` type should be defined to the platform-specific time of day type.

## 8.154 osclconfig\_unix\_android.h File Reference

```
#include <stdlib.h>
#include <stdarg.h>
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ctype.h>
#include <math.h>
```

### Defines

- #define OSCL\_DISABLE\_INLINES 0
- #define OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_MATH\_SUPPORT 1
- #define OSCL\_HAS\_GLOBAL\_VARIABLE\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_STRING\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_WIDE\_STRING\_SUPPORT 0
- #define OSCL\_HAS\_ANSI\_STDIO\_SUPPORT 1
- #define OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN 1
- #define OSCL\_HAS\_UNIX\_SUPPORT 1
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_HAS\_IPHONE\_SUPPORT 0
- #define OSCL\_NATIVE\_INT64\_TYPE int64\_t
- #define OSCL\_NATIVE\_UINT64\_TYPE uint64\_t
- #define INT64(x) x##LL
- #define UINT64(x) x##ULL
- #define INT64\_HILO(high, low) (((high##LL))<<32)|low)
- #define UINT64\_HILO(high, low) (((high##ULL))<<32)|low)
- #define OSCL\_HAS\_UNICODE\_SUPPORT 1
- #define OSCL\_NATIVE\_WCHAR\_TYPE wchar\_t
- #define \_STRLIT(x) L ## x
- #define \_STRLIT\_CHAR(x) x
- #define \_STRLIT\_WCHAR(x) L ## x
- #define OSCL\_HAS\_TLS\_SUPPORT 1
- #define OSCL\_TLS\_IS\_KEYED 1
- #define OSCL\_TLS\_KEY\_CREATE\_FUNC(key) (pthread\_key\_create(&key,NULL)==0)
- #define OSCL\_TLS\_KEY\_DELETE\_FUNC(key) pthread\_key\_delete(key)
- #define OSCL\_TLS\_STORE\_FUNC(key, ptr) (pthread\_setspecific(key,(const void\*)ptr)==0)
- #define OSCL\_TLS\_GET\_FUNC(key) pthread\_getspecific(key)
- #define OSCL\_HAS\_BASIC\_LOCK 1

## Typedefs

- `typedef pthread_key_t TOsclTlsKey`
- `typedef pthread_mutex_t TOsclBasicLockObject`



### 8.154.1 Define Documentation

8.154.1.1 `#define _STRLIT(x) L ## x`

8.154.1.2 `#define _STRLIT_CHAR(x) x`

8.154.1.3 `#define _STRLIT_WCHAR(x) L ## x`

8.154.1.4 `#define INT64(x) x##LL`

8.154.1.5 `#define INT64_HILO(high, low) (((high##LL))<<32)|low)`

8.154.1.6 `#define OSCL_DISABLE_INLINES 0`

8.154.1.7 `#define OSCL_HAS_ANSI_MATH_SUPPORT 1`

8.154.1.8 `#define OSCL_HAS_ANSI_STDIO_SUPPORT 1`

8.154.1.9 `#define OSCL_HAS_ANSI_STDLIB_SUPPORT 1`

8.154.1.10 `#define OSCL_HAS_ANSI_STRING_SUPPORT 1`

8.154.1.11 `#define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 0`

8.154.1.12 `#define OSCL_HAS_BASIC_LOCK 1`

8.154.1.13 `#define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1`

8.154.1.14 `#define OSCL_HAS_IPHONE_SUPPORT 0`

8.154.1.15 `#define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0`

8.154.1.16 `#define OSCL_HAS_MSWIN_SUPPORT 0`

8.154.1.17 `#define OSCL_HAS_SYMBIAN_SUPPORT 0`

8.154.1.18 `#define OSCL_HAS_TLS_SUPPORT 1`

8.154.1.19 `#define OSCL_HAS_UNICODE_SUPPORT 1`

8.154.1.20 `#define OSCL_HAS_UNIX_SUPPORT 1`

8.154.1.21 `#define OSCL_MEMFRAG_PTR_BEFORE_LEN 1`

8.154.1.22 `#define OSCL_NATIVE_INT64_TYPE int64_t`

8.154.1.23 `#define OSCL_NATIVE_UINT64_TYPE uint64_t`

8.154.1.24 `#define OSCL_NATIVE_WCHAR_TYPE wchar_t`

8.154.1.25 `#define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)`

8.154.1.26 `#define OSCL_TLS_IS_KEYED 1`

8.154.1.27 `#define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)`

## 8.155 osclconfig\_unix\_common.h File Reference

```
#include <stdlib.h>
#include <stdarg.h>
#include <sys/types.h>
#include <stdio.h>
#include <wchar.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ctype.h>
#include <math.h>
```

### Defines

- #define OSCL\_DISABLE\_INLINES 0
- #define OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_MATH\_SUPPORT 1
- #define OSCL\_HAS\_GLOBAL\_VARIABLE\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_STRING\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_WIDE\_STRING\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_STDIO\_SUPPORT 1
- #define OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN 1
- #define OSCL\_HAS\_UNIX\_SUPPORT 1
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_NATIVE\_INT64\_TYPE int64\_t
- #define OSCL\_NATIVE\_UINT64\_TYPE uint64\_t
- #define INT64(x) x##LL
- #define UINT64(x) x##ULL
- #define INT64\_HILO(high, low) (((high##LL))<<32)|low)
- #define UINT64\_HILO(high, low) (((high##ULL))<<32)|low)
- #define OSCL\_HAS\_UNICODE\_SUPPORT 1
- #define OSCL\_NATIVE\_WCHAR\_TYPE wchar\_t
- #define \_STRLIT(x) L ## x
- #define \_STRLIT\_CHAR(x) x
- #define \_STRLIT\_WCHAR(x) L ## x
- #define OSCL\_HAS\_TLS\_SUPPORT 1
- #define OSCL\_TLS\_IS\_KEYED 1
- #define OSCL\_TLS\_KEY\_CREATE\_FUNC(key) (pthread\_key\_create(&key,NULL)==0)
- #define OSCL\_TLS\_KEY\_DELETE\_FUNC(key) pthread\_key\_delete(key)
- #define OSCL\_TLS\_STORE\_FUNC(key, ptr) (pthread\_setspecific(key,(const void\*)ptr)==0)
- #define OSCL\_TLS\_GET\_FUNC(key) pthread\_getspecific(key)
- #define OSCL\_HAS\_BASIC\_LOCK 1

## Typedefs

- `typedef pthread_key_t TOsclTlsKey`
- `typedef pthread_mutex_t TOsclBasicLockObject`



### 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 1`

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_SYMBIAN_SUPPORT 0`

8.155.1.17 `#define OSCL_HAS_TLS_SUPPORT 1`

8.155.1.18 `#define OSCL_HAS_UNICODE_SUPPORT 1`

8.155.1.19 `#define OSCL_HAS_UNIX_SUPPORT 1`

8.155.1.20 `#define OSCL_MEMFRAG_PTR_BEFORE_LEN 1`

8.155.1.21 `#define OSCL_NATIVE_INT64_TYPE int64_t`

8.155.1.22 `#define OSCL_NATIVE_UINT64_TYPE uint64_t`

8.155.1.23 `#define OSCL_NATIVE_WCHAR_TYPE wchar_t`

8.155.1.24 `#define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)`

8.155.1.25 `#define OSCL_TLS_IS_KEYED 1`

8.155.1.26 `#define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)`

---

## 8.156 osclconfig\_util.h File Reference

```
#include "osclconfig.h"
#include <stdio.h>
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include "osclconfig_util_check.h"
```

### Defines

- #define OSCL\_CLOCK\_HAS\_DRIFT\_CORRECTION 0
- #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- #define OSCL\_HAS\_SYMBIAN\_MATH 0
- #define OSCL RAND\_MAX RAND\_MAX
- #define SLEEP\_ONE\_SEC sleep(1)

### 8.156.1 Define Documentation

- 8.156.1.1 #define OSCL\_CLOCK\_HAS\_DRIFT\_CORRECTION 0
- 8.156.1.2 #define OSCL\_HAS\_SYMBIAN\_MATH 0
- 8.156.1.3 #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- 8.156.1.4 #define OSCL RAND\_MAX RAND\_MAX
- 8.156.1.5 #define SLEEP\_ONE\_SEC sleep(1)

---

## **8.157 osclconfig\_util\_check.h File Reference**

## 8.158 pvlogger.h File Reference

This file contains basic logger interfaces for common use across platforms.

```
#include "oscl_base.h"
#include "oscl_vector.h"
#include "oscl_shared_ptr.h"
#include "oscl_base_alloc.h"
```

### Data Structures

- class [PVLogger](#)

### Defines

- #define [PVLOGMSG\\_INST\\_REL](#) 0
- #define [PVLOGMSG\\_INST\\_PROF](#) 1
- #define [PVLOGMSG\\_INST\\_HLDBG](#) 2
- #define [PVLOGMSG\\_INST\\_MLDBG](#) 3
- #define [PVLOGMSG\\_INST\\_LLDBG](#) 4
- #define [PVLOGGER\\_INST\\_LEVEL](#) 5
- #define [\\_PVLOGGER\\_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [\\_PVLOGGER\\_LOGMSG\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [\\_PVLOGGER\\_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [\\_PVLOGGER\\_LOGBIN\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_INST\\_LEVEL\\_SUPPORT](#) 1
- #define [PVLOGGER\\_LOGMSG\\_PVLOGMSG\\_INST\\_REL\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGMSG\\_V\\_PVLOGMSG\\_INST\\_REL\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_PVLOGMSG\\_INST\\_REL\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_V\\_PVLOGMSG\\_INST\\_REL\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGMSG\\_PVLOGMSG\\_INST\\_PROF\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGMSG\\_V\\_PVLOGMSG\\_INST\\_PROF\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_PVLOGMSG\\_INST\\_PROF\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_V\\_PVLOGMSG\\_INST\\_PROF\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGMSG\\_PVLOGMSG\\_INST\\_HLDBG\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGMSG\\_V\\_PVLOGMSG\\_INST\\_HLDBG\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGMSG\\_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_PVLOGMSG\\_INST\\_HLDBG\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER\\_LOGBIN\\_V\\_PVLOGMSG\\_INST\\_HLDBG\(LOGGER, LEVEL, MESSAGE\) \\_- PVLOGGER\\_LOGBIN\\_V\(LOGGER, LEVEL, MESSAGE\)](#)

- #define **PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_V\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGMSG\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_V**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGMSG\_V\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGBIN\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_V**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGBIN\_V\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOG\_USE\_ONLY**(x) x
- #define **PVLOGGER\_ENABLE** 1

## Variables

- const int32 **PVLOGGER\_LEVEL\_UNINITIALIZED** = -1
- const **PVLogger::log\_level\_type** **PVLOGMSG\_EMERG** = 0
- const **PVLogger::log\_level\_type** **PVLOGMSG\_ALERT** = 1
- const **PVLogger::log\_level\_type** **PVLOGMSG\_CRIT** = 2
- const **PVLogger::log\_level\_type** **PVLOGMSG\_ERR** = 3
- const **PVLogger::log\_level\_type** **PVLOGMSG\_WARNING** = 4
- const **PVLogger::log\_level\_type** **PVLOGMSG\_NOTICE** = 5
- const **PVLogger::log\_level\_type** **PVLOGMSG\_INFO** = 6
- const **PVLogger::log\_level\_type** **PVLOGMSG\_STACK\_TRACE** = 7
- const **PVLogger::log\_level\_type** **PVLOGMSG\_DEBUG** = 8
- const **PVLogger::log\_level\_type** **PVLOGMSG\_FATAL\_ERROR** = **PVLOGMSG\_EMERG**
- const **PVLogger::log\_level\_type** **PVLOGMSG\_NONFATAL\_ERROR** = **PVLOGMSG\_ERR**
- const **PVLogger::log\_level\_type** **PVLOGMSG\_STATISTIC** = **PVLOGMSG\_INFO**
- const **PVLogger::log\_level\_type** **PVLOGMSG\_VERBOSE** = **PVLOGMSG\_DEBUG**

### 8.158.1 Detailed Description

This file contains basic logger interfaces for common use across platforms.

This is the main entry point header file for the logger library. It should be the only one users directly include.

## 8.158.2 Define Documentation

### 8.158.2.1 #define \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgBuffers MESSAGE; \
    }\
  }\
}
```

### 8.158.2.2 #define \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgBuffersV MESSAGE; \
    }\
  }\
}
```

### 8.158.2.3 #define \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgString MESSAGE; \
    }\
  }\
}
```

### 8.158.2.4 #define \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgStringV MESSAGE; \
    }\
  }\
}
```

**8.158.2.5 #define PVLOGGER\_ENABLE 1**

In case logging is compiled out, there is no need to compile the logger runtime code either.

**8.158.2.6 #define PVLOGGER\_INST\_LEVEL 5****8.158.2.7 #define PVLOGGER\_INST\_LEVEL\_SUPPORT 1****8.158.2.8 #define PVLOGGER\_LOG\_USE\_ONLY(x) x**

Used to compile in/out lines of code that are used only for **PVLogger** macros.

This code will be removed at compile time when **PVLogger** is disabled, i.e. Release mode. So do not put in any code that is necessary for correct functionality of the module

**8.158.2.9 #define PVLOGGER\_LOGBIN(IL, LOGGER, LEVEL, MESSAGE)  
PVLOGGER\_LOGBIN\_ ## IL (LOGGER, LEVEL, MESSAGE)**

This is a binary API to log messages

**Parameters:**

**IL** Instrumentation level.

**LOGGER** Pointer to the logger object, that acts as the logging control/interface point

**LEVEL** Log level of the message

**MESSAGE** Log Message which includes the message id, and message buffers that need to be logged.

Example Usage: PVLOGGER\_LOGBIN (PVLOGMSG\_INST\_LLDBG, logger\_1, PVLOGMSG\_WARNING, (10, 3, msgBuf1Size, msgBuf1, msgBuf2Size, msgBuf2, msgBuf3Size, msgBuf3));

-This message contains THREE (ptr\_len, ptr) pairs. Log level of this msg is PVLOGMSG\_WARNING, message id is 10.

- 8.158.2.10 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_HLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.11 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_LLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.12 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_MLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.13 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.14 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.15 #define PVLOGGER\_LOGBIN\_V(IL, LOGGER, LEVEL, MESSAGE)  
PVLOGGER\_LOGBIN\_V\_## IL (LOGGER, LEVEL, MESSAGE)
- 8.158.2.16 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_HLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.17 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_LLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.18 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.19 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.20 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_V\_INST\_MLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.21 #define PVLOGGER\_LOGMSG(IL, LOGGER, LEVEL, MESSAGE)  
PVLOGGER\_LOGMSG\_## IL (LOGGER, LEVEL, MESSAGE)

This is the text based API to log messages

**Parameters:**

**IL** Instrumentation level.

**LOGGER** Pointer to the logger object, that acts as the logging control/interface point

**LEVEL** Log level of the message

**MESSAGE** Log Message which includes the message id, and any kind of formatting information

Example Usage: PVLOGGER\_LOGMSG(PVLOGMSG\_INST\_LLDBG, logger\_1, PVLOGMSG\_WARNING, (13, "Test Messsage to Node 1

")); -This message of log level PVLOGMSG\_WARNING, and has a message id of 13

- 8.158.2.22 #define PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_HLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- 8.158.2.23 #define PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_LLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- 8.158.2.24 #define PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_MLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- 8.158.2.25 #define PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- 8.158.2.26 #define PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- 8.158.2.27 #define PVLOGGER\_LOGMSG\_V(IL, LOGGER, LEVEL, MESSAGE)  
PVLOGGER\_LOGMSG\_V\_## IL (LOGGER, LEVEL, MESSAGE)
- 8.158.2.28 #define PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_HLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.29 #define PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_LLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.30 #define PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_MLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.31 #define PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.32 #define PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.33 #define PVLOGMSG\_INST\_HLDBG 2

#### High Level Debug Layer

This layer should contain messages that have very minimal impact on performance, but are at lower level (i.e., provide more information) than would be appropriate in a shipping product. The messages are probably used to gather information and validate proper functionality at a high level as might be appropriate for IOT, stress testing, or QA testing.

#### 8.158.2.34 #define PVLOGMSG\_INST\_LLDBG 4

#### Low Level Debug Layer

This layer should contain messages for early functional testing. The messages are typically at a very low-level and allow testing the functionality of individual modules and components. Messages at this layer will typically have a performance impact (sometimes significant) due to the fact that they are at such a low level.

### 8.158.2.35 #define PVLOGMSG\_INST\_MLDBG 3

Mid Level Debug Layer

This layer should contain messages that are useful in the middle stages of the development cycle where major components are being integrated. The components themselves should already be well-tested so the emphasis is on interfaces between these components and integration testing. Messages at this layer may have some performance impact.

### 8.158.2.36 #define PVLOGMSG\_INST\_PROF 1

Profile Layer

The profile layer is used for messages and information related to measuring and reporting performance-related information.

### 8.158.2.37 #define PVLOGMSG\_INST\_REL 0

Release Layer

The release layer should only be used for messages that should remain in the final release. In certain cases all messaging may be disabled depending on customer requirements. However, when allowed the release layer should contain information that will be useful diagnosing problems in a released product (perhaps after entering a diagnostic mode), but with absolutely minimal performance impact when disabled at runtime.

## 8.158.3 Variable Documentation

### 8.158.3.1 const int32 PVLOGGER\_LEVEL\_UNINITIALIZED = -1

### 8.158.3.2 const PVLogger::log\_level\_type PVLOGMSG\_ALERT = 1

action must be taken immediately

### 8.158.3.3 const PVLogger::log\_level\_type PVLOGMSG\_CRIT = 2

critical conditions

### 8.158.3.4 const PVLogger::log\_level\_type PVLOGMSG\_DEBUG = 8

debug-level messages

### 8.158.3.5 const PVLogger::log\_level\_type PVLOGMSG\_EMERG = 0

system is unusable

### 8.158.3.6 const PVLogger::log\_level\_type PVLOGMSG\_ERR = 3

error conditions

**8.158.3.7 const PVLogger::log\_level\_type PVLOGMSG\_FATAL\_ERROR = PVLOGMSG\_EMERG**

**8.158.3.8 const PVLogger::log\_level\_type PVLOGMSG\_INFO = 6**

informational

**8.158.3.9 const PVLogger::log\_level\_type PVLOGMSG\_NONFATAL\_ERROR = PVLOGMSG\_ERR**

**8.158.3.10 const PVLogger::log\_level\_type PVLOGMSG\_NOTICE = 5**

normal but significant condition

**8.158.3.11 const PVLogger::log\_level\_type PVLOGMSG\_STACK\_TRACE = 7**

function enter and exit

**8.158.3.12 const PVLogger::log\_level\_type PVLOGMSG\_STATISTIC = PVLOGMSG\_INFO**

**8.158.3.13 const PVLogger::log\_level\_type PVLOGMSG\_VERBOSE = PVLOGMSG\_DEBUG**

**8.158.3.14 const PVLogger::log\_level\_type PVLOGMSG\_WARNING = 4**

warning conditions

## 8.159 pvlogger\_accessories.h File Reference

```
#include "oscl_base.h"
#include "pvlogger.h"
```

### Data Structures

- class [AllPassFilter](#)
- class [PVLoggerAppender](#)
- class [PVLoggerFilter](#)
- class [PVLoggerLayout](#)

### Variables

- const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_ACCEPT = 1](#)
- const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_REJECT = 2](#)
- const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_NEUTRAL = 3](#)

#### 8.159.1 Variable Documentation

8.159.1.1 const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_ACCEPT = 1](#)

8.159.1.2 const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_NEUTRAL = 3](#)

8.159.1.3 const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_REJECT = 2](#)

## 8.160 pvlogger\_c.h File Reference

This file contains basic logger interfaces for common use across platforms. C-callable version.

```
#include "osclconfig.h"
```

### Defines

- #define PVLOGGER\_C\_INST\_LEVEL 5
- #define PVLOGMSG\_C\_INST\_REL 0
- #define PVLOGMSG\_C\_INST\_PROF 1
- #define PVLOGMSG\_C\_INST\_HLDBG 2
- #define PVLOGMSG\_C\_INST\_MLDBG 3
- #define PVLOGMSG\_C\_INST\_LLDBG 4
- #define PVLOGMSG\_C\_EMERG 0
- #define PVLOGMSG\_C\_ALERT 1
- #define PVLOGMSG\_C\_CRIT 2
- #define PVLOGMSG\_C\_ERR 3
- #define PVLOGMSG\_C\_WARNING 4
- #define PVLOGMSG\_C\_NOTICE 5
- #define PVLOGMSG\_C\_INFO 6
- #define PVLOGMSG\_C\_STACK\_TRACE 7
- #define PVLOGMSG\_C\_STACK\_DEBUG 8

### Functions

- OSCL\_IMPORT\_REF void \* [pvLogger\\_GetLoggerObject](#) (const char \*tag)
- OSCL\_IMPORT\_REF int [pvLogger\\_IsActive](#) (void \*logger, int log\_level)
- OSCL\_IMPORT\_REF void [pvLogger\\_LogMsgString](#) (void \*logger, int msgID, const char \*fmt,...)

### 8.160.1 Detailed Description

This file contains basic logger interfaces for common use across platforms. C-callable version.

This is the main entry point header file for the logger library. It should be the only one users directly include.

## 8.160.2 Define Documentation

- 8.160.2.1 `#define PVLOGGER_C_INST_LEVEL 5`
- 8.160.2.2 `#define PVLOGMSG_C_ALERT 1`
- 8.160.2.3 `#define PVLOGMSG_C_CRIT 2`
- 8.160.2.4 `#define PVLOGMSG_C_EMERG 0`
- 8.160.2.5 `#define PVLOGMSG_C_ERR 3`
- 8.160.2.6 `#define PVLOGMSG_C_INFO 6`
- 8.160.2.7 `#define PVLOGMSG_C_INST_HLDBG 2`
- 8.160.2.8 `#define PVLOGMSG_C_INST_LLDBG 4`
- 8.160.2.9 `#define PVLOGMSG_C_INST_MLDBG 3`
- 8.160.2.10 `#define PVLOGMSG_C_INST_PROF 1`
- 8.160.2.11 `#define PVLOGMSG_C_INST_REL 0`
- 8.160.2.12 `#define PVLOGMSG_C_NOTICE 5`
- 8.160.2.13 `#define PVLOGMSG_C_STACK_DEBUG 8`
- 8.160.2.14 `#define PVLOGMSG_C_STACK_TRACE 7`
- 8.160.2.15 `#define PVLOGMSG_C_WARNING 4`

## 8.160.3 Function Documentation

- 8.160.3.1 `OSCL_IMPORT_REF void* pvLogger_GetLoggerObject (const char * tag)`
- 8.160.3.2 `OSCL_IMPORT_REF int pvLogger_IsActive (void * logger, int log_level)`
- 8.160.3.3 `OSCL_IMPORT_REF void pvLogger_LogMsgString (void * logger, int msgID, const char * fmt, ...)`

## **8.161 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, 115  
~BufFragGroup  
    BufFragGroup, 121  
~BufferMgr  
    BufferMgr, 118  
~CallbackTimer  
    CallbackTimer, 124  
~CallbackTimerObserver  
    CallbackTimerObserver, 126  
~DNSRequestParam  
    DNSRequestParam, 133  
~GetHostByNameParam  
    GetHostByNameParam, 136  
~HeapBase  
    HeapBase, 138  
~MM\_AllocInfo  
    MM\_AllocInfo, 150  
~MM\_AllocNode  
    MM\_AllocNode, 151  
~MM\_Audit\_Imp  
    MM\_Audit\_Imp, 154  
~MediaData  
    MediaData, 143  
~MemAllocator  
    MemAllocator, 146  
~OSCLMemAutoPtr  
    OSCLMemAutoPtr, 436  
~OSCL\_FastString  
    OSCL\_FastString, 176  
~OSCL\_HeapString  
    osclutil, 84  
~OSCL\_HeapStringA  
    OSCL\_HeapStringA, 200  
~OSCL\_StackString  
    osclutil, 84  
~OSCL\_String  
    OSCL\_String, 261  
~OSCL\_wFastString  
    OSCL\_wFastString, 295  
~OSCL\_wHeapString  
    osclutil, 84  
~OSCL\_wHeapStringA  
    OSCL\_wHeapStringA, 300  
~OSCL\_wStackString  
    osclutil, 84  
~OSCL\_wString  
    OSCL\_wString, 305  
~OsclAcceptMethod  
    OsclAcceptMethod, 308  
~OsclActiveObject  
    OsclActiveObject, 311  
~OsclAllocDestructDealloc  
    OsclAllocDestructDealloc, 314  
~OsclAsyncFile  
    OsclAsyncFile, 317  
~OsclAsyncFileBuffer  
    OsclAsyncFileBuffer, 320  
~OsclBinIStream  
    OsclBinIStream, 324  
~OsclBinOStream  
    OsclBinOStream, 331  
~OsclBindMethod  
    OsclBindMethod, 322  
~OsclCacheObserver  
    Oscl\_File::OsclCacheObserver, 187  
~OsclComponentRegistry  
    OsclComponentRegistry, 344  
~OsclComponentRegistryElement  
    OsclComponentRegistryElement, 346  
~OsclConnectMethod  
    OsclConnectMethod, 348  
~OsclDNS  
    OsclDNS, 351  
~OsclDNSI  
    OsclDNSI, 353  
~OsclDNSIBase  
    OsclDNSIBase, 356  
~OsclDNSObserver  
    OsclDNSObserver, 361  
~OsclDNSRequest  
    OsclDNSRequest, 362  
~OsclDestructDealloc  
    OsclDestructDealloc, 350  
~OsclExclusiveArrayPtr  
    OsclExclusiveArrayPtr, 381  
~OsclExclusivePtr  
    OsclExclusivePtr, 384  
~OsclExclusivePtrA  
    OsclExclusivePtrA, 387

---

~OsclExecSchedulerCommonBase  
     OsclExecSchedulerCommonBase, 395  
 ~OsclFileCache  
     OsclFileCache, 402  
 ~OsclGetHostByNameMethod  
     OsclGetHostByNameMethod, 413  
 ~OsclIPSocketI  
     OsclIPSocketI, 419  
 ~OsclJump  
     OsclJump, 421  
 ~OsclListenMethod  
     OsclListenMethod, 422  
 ~OsclLockBase  
     OsclLockBase, 424  
 ~OsclMemAudit  
     OsclMemAudit, 429  
 ~OsclMemPoolFixedChunkAllocator  
     OsclMemPoolFixedChunkAllocator, 444  
 ~OsclMemPoolFixedChunkAllocatorObserver  
     OsclMemPoolFixedChunkAllocator-  
         Observer, 447  
 ~OsclMemPoolResizableAllocator  
     OsclMemPoolResizableAllocator, 449  
 ~OsclMemPoolResizableAllocatorMemoryObserver  
     OsclMemPoolResizableAllocatorMemory-  
         Observer, 456  
 ~OsclMemPoolResizableAllocatorObserver  
     OsclMemPoolResizableAllocatorObserver,  
         457  
 ~OsclMemStatsNode  
     OsclMemStatsNode, 458  
 ~OsclMutex  
     OsclMutex, 459  
 ~OsclNativeFile  
     OsclNativeFile, 463  
 ~OsclNullLock  
     OsclNullLock, 467  
 ~OsclPriorityQueue  
     OsclPriorityQueue, 471  
 ~OsclPriorityQueueBase  
     OsclPriorityQueueBase, 474  
 ~OsclRecvFromMethod  
     OsclRecvFromMethod, 486  
 ~OsclRecvMethod  
     OsclRecvMethod, 490  
 ~OsclRefCounter  
     OsclRefCounter, 492  
 ~OsclRefCounterDA  
     OsclRefCounterDA, 494  
 ~OsclRefCounterMTDA  
     OsclRefCounterMTDA, 498  
 ~OsclRefCounterMTSA  
     OsclRefCounterMTSA, 500  
 ~OsclRefCounterMemFrag

OsclRefCounterMemFrag, 496  
 ~OsclRefCounterSA  
     OsclRefCounterSA, 502  
 ~OsclRegistryAccessClient  
     OsclRegistryAccessClient, 504  
 ~OsclRegistryClient  
     OsclRegistryClient, 509  
 ~OsclRegistryServTlsImpl  
     OsclRegistryServTlsImpl, 515  
 ~OsclSchedulerObserver  
     OsclSchedulerObserver, 517  
 ~OsclScopedLock  
     OsclScopedLock, 518  
 ~OsclSemaphore  
     OsclSemaphore, 521  
 ~OsclSendMethod  
     OsclSendMethod, 523  
 ~OsclSendToMethod  
     OsclSendToMethod, 525  
 ~OsclSharedPtr  
     OsclSharedPtr, 528  
 ~OsclShutdownMethod  
     OsclShutdownMethod, 530  
 ~OsclSingleton  
     OsclSingleton, 532  
 ~OsclSocketI  
     OsclSocketI, 536  
 ~OsclSocketIBase  
     OsclSocketIBase, 541  
 ~OsclSocketMethod  
     OsclSocketMethod, 546  
 ~OsclSocketObserver  
     OsclSocketObserver, 548  
 ~OsclSocketRequestAO  
     OsclSocketRequestAO, 551  
 ~OsclSocketServ  
     OsclSocketServ, 554  
 ~OsclSocketServIBase  
     OsclSocketServIBase, 559  
 ~OsclTCPSocket  
     OsclTCPSocket, 566  
 ~OsclTCPSocketI  
     OsclTCPSocketI, 573  
 ~OsclTLS  
     OsclTLS, 592  
 ~OsclTLSEx  
     OsclTLSEx, 594  
 ~OsclThread  
     OsclThread, 575  
 ~OsclThreadLock  
     OsclThreadLock, 579  
 ~OsclTimer  
     OsclTimer, 583  
 ~OsclTimerObject

OsclTimerObject, 587  
 ~OsclTimerObserver  
   OsclTimerObserver, 590  
 ~OsclUDPSocket  
   OsclUDPSocket, 602  
 ~OsclUDPSocketI  
   OsclUDPSocketI, 608  
 ~Oscl\_Alloc  
   Oscl\_Alloc, 170  
 ~Oscl\_Dealloc  
   Oscl\_Dealloc, 171  
 ~Oscl\_File  
   Oscl\_File, 181  
 ~Oscl\_FileFind  
   Oscl\_FileFind, 190  
 ~Oscl\_FileServer  
   Oscl\_FileServer, 193  
 ~Oscl\_Linked\_List  
   Oscl\_Linked\_List, 206  
 ~Oscl\_Linked\_List\_Base  
   Oscl\_Linked\_List\_Base, 212  
 ~Oscl\_MTLinked\_List  
   Oscl\_MTLinked\_List, 225  
 ~Oscl\_Opaque\_Type\_Alloc  
   Oscl\_Opaque\_Type\_Alloc, 229  
 ~Oscl\_Opaque\_Type\_Alloc\_LL  
   Oscl\_Opaque\_Type\_Alloc\_LL, 231  
 ~Oscl\_Opaque\_Type\_Compare  
   Oscl\_Opaque\_Type\_Compare, 233  
 ~Oscl\_Queue  
   Oscl\_Queue, 237  
 ~Oscl\_Queue\_Base  
   Oscl\_Queue\_Base, 239  
 ~Oscl\_Rb\_Tree  
   Oscl\_Rb\_Tree, 244  
 ~Oscl\_TAlloc  
   Oscl\_TAlloc, 282  
 ~Oscl\_Tag  
   Oscl\_Tag, 265  
 ~Oscl\_TagTree  
   Oscl\_TagTree, 270  
 ~Oscl\_Vector  
   Oscl\_Vector, 286  
 ~Oscl\_Vector\_Base  
   Oscl\_Vector\_Base, 291  
 ~PVActiveBase  
   PVActiveBase, 613  
 ~PVLogger  
   PVLogger, 618  
 ~PVLoggerAppender  
   PVLoggerAppender, 623  
 ~PVLoggerFilter  
   PVLoggerFilter, 625  
 ~PVLoggerLayout  
   PVLoggerLayout, 626  
 ~PVLoggerRegistry  
   PVLoggerRegistry, 628  
 ~PVSchedulerStopper  
   PVSchedulerStopper, 631  
 ~PVThreadContext  
   PVThreadContext, 634  
 ~SendToParam  
   SendToParam, 640  
 ~\_OsclBasicAllocator  
   \_OsclBasicAllocator, 109  
 ~\_OsclHeapBase  
   \_OsclHeapBase, 111  
 \_OSCL\_Abort  
   osclbase, 36  
 \_OSCL\_CLEANUP\_BASE\_CLASS  
   osclmemory, 50  
 \_OSCL\_TRAP\_NEW  
   osclmemory, 50  
 \_OsclBasicAllocator, 108  
 \_OsclBasicAllocator  
   ~\_OsclBasicAllocator, 109  
   allocate, 109  
   deallocate, 109  
 \_OsclHeapBase, 110  
   \_OsclHeapBase, 111  
 \_OsclHeapBase  
   ~\_OsclHeapBase, 111  
   \_OsclHeapBase, 111  
   PVCleanupStack, 111  
 \_OsclInteger64Transport  
   oscl\_int64\_utils.h, 705  
 \_Ownership  
   OSCLMemAutoPtr, 438  
 \_PVLOGGER\_LOGBIN  
   pvlogger.h, 851  
 \_PVLOGGER\_LOGBIN\_V  
   pvlogger.h, 851  
 \_PVLOGGER\_LOGMSG  
   pvlogger.h, 851  
 \_PVLOGGER\_LOGMSG\_V  
   pvlogger.h, 851  
 \_PV\_TRAP  
   oscl\_error\_imp\_fatalerror.h, 685  
   oscl\_error\_imp\_jumps.h, 686  
   osclerror, 88  
 \_PV\_TRAP\_NO\_TLS  
   oscl\_error\_imp\_fatalerror.h, 685  
   oscl\_error\_imp\_jumps.h, 686  
   osclerror, 88  
 \_Ptr  
   OsclExclusiveArrayPtr, 382  
   OsclExclusivePtr, 385  
   OsclExclusivePtrA, 388

OsclSingleton, 533  
 OsclTLS, 593  
 OsclTLSEx, 595  
**\_STRLIT**  
 osclconfig\_unix\_android.h, 842  
 osclconfig\_unix\_common.h, 846  
**\_STRLIT\_CHAR**  
 osclconfig\_unix\_android.h, 842  
 osclconfig\_unix\_common.h, 846  
**\_STRLIT\_WCHAR**  
 osclconfig\_unix\_android.h, 842  
 osclconfig\_unix\_common.h, 846  
**\_TFS\_**  
 osclconfig.h, 803  
**\_Validate\_BasicTimeDateStruct\_**  
 osclconfig\_time\_check.h, 838  
**\_Validate\_BasicTimeStruct\_**  
 osclconfig\_time\_check.h, 838  
**\_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\_TOsclConditionObject\_defined\_**  
 osclconfig\_proc\_check.h, 831  
**\_verify\_TOsclFileOffset\_defined\_**  
 osclconfig\_io\_check.h, 822  
**\_verify\_TOsclMutexObject\_defined\_**  
 osclconfig\_proc\_check.h, 831  
**\_verify\_TOsclSemaphoreObject\_defined\_**  
 osclconfig\_proc\_check.h, 831  
**\_verify\_TOsclThreadFuncArg\_defined\_**  
 osclconfig\_proc\_check.h, 831  
**\_verify\_TOsclThreadFuncRet\_defined\_**  
 osclconfig\_proc\_check.h, 831  
**\_verify\_TOsclThreadId\_defined\_**  
 osclconfig\_proc\_check.h, 831  
**\_verify\_TOsclThreadObject\_defined\_**  
 osclconfig\_proc\_check.h, 831  
**\_fixedCaches**  
 OsclFileCache, 402  
**\_movableCache**  
 OsclFileCache, 402  
**\_oscl\_audit\_malloc**  
 osclmemory, 59  
**\_oscl\_audit\_free**  
 osclmemory, 59  
**\_oscl\_audit\_new**  
 osclmemory, 59  
**\_oscl\_audit\_realloc**  
 osclmemory, 60  
**\_oscl\_calloc**  
 osclmemory, 60  
**\_oscl\_default\_audit\_malloc**  
 osclmemory, 60  
**\_oscl\_default\_audit\_new**  
 osclmemory, 60  
**\_oscl\_default\_audit\_realloc**  
 osclmemory, 60  
**\_oscl\_free**  
 osclmemory, 60  
**\_oscl\_malloc**  
 osclmemory, 60  
**\_oscl\_realloc**  
 osclmemory, 60  
  
**a**  
 internalLeave, 139  
**Abort**  
 OsclDNSMethod, 359  
 OsclDNSRequestAO, 364  
 OsclSocketMethod, 546  
 OsclSocketRequestAO, 551  
**AbortAll**  
 OsclDNSMethod, 359  
 OsclSocketMethod, 546  
**Accept**  
 OsclAcceptMethod, 308  
 OsclAcceptRequest, 309  
 OsclSocketI, 536  
 OsclSocketIBase, 541  
 OsclTCPSocket, 566  
 OsclTCPSocketI, 573  
**AcceptParam**, 112  
 AcceptParam, 112  
**AcceptParam**  
 AcceptParam, 112  
 iBlankSocket, 112  
**AcceptRequest**  
 OsclAcceptMethod, 308  
**Activate**  
 OsclDNSRequest, 362  
 OsclSocketRequest, 549  
 PVActiveBase, 613  
**Add**  
 OsclSocketServRequestList, 560  
 OsclTimerQ, 591

add\_element  
     Oscl\_Linked\_List, 207  
     Oscl\_Linked\_List\_Base, 212  
     Oscl\_MTLinked\_List, 226  
 add\_ref  
     CHHeapRep, 130  
 add\_to\_front  
     Oscl\_Linked\_List, 207  
     Oscl\_Linked\_List\_Base, 212  
     Oscl\_MTLinked\_List, 226  
 addAllocNode  
     MM\_Audit\_Imp, 154  
 AddAppender  
     PVLogger, 618  
 AddFilter  
     PVLogger, 618  
 AddFixedCache  
     Oscl\_File, 181  
     OsclFileCache, 402  
 AddFragment  
     BufFragGroup, 121  
 AddLocalFragment  
     MediaData, 143  
 addnewmempoolbuffer  
     OsclMemPoolResizableAllocator, 449  
 addRef  
     Oscl\_DefAllocWithRefCounter, 173  
     OsclMemPoolFixedChunkAllocator, 444  
     OsclMemPoolResizableAllocator, 449  
     OsclRefCount, 492  
     OsclRefCountDA, 495  
     OsclRefCountMTDA, 499  
     OsclRefCountMTSA, 501  
     OsclRefCountSA, 503  
 address  
     Oscl\_TAlloc, 282  
 addressListCapacity  
     GetHostNameParam, 135  
 AddToExecTimerQ  
     OsclExecSchedulerCommonBase, 395  
 AddToScheduler  
     OsclActiveObject, 311  
     OsclTimerObject, 587  
     PVActiveBase, 613  
 After  
     OsclTimerObject, 587  
 Alloc  
     OsclIPSocketI, 419  
     OsclSocketMethod, 546  
     OsclSocketRequestAO, 551  
 ALLOC\_AND\_CONSTRUCT  
     osclbase, 32  
 alloc\_and\_construct  
     Oscl\_TAlloc, 282  
 alloc\_and\_construct\_fl  
     Oscl\_TAlloc, 282  
 ALLOC\_NODE\_FLAG  
     osclmemory, 62  
 alloc\_type  
     PVLogger, 618  
     PVLoggerRegistry, 628  
 ALLOCATE  
     osclbase, 32  
 allocate  
     \_OsclBasicAllocator, 109  
     MemAllocator, 146  
     Oscl\_Alloc, 170  
     Oscl\_DefAlloc, 172  
     Oscl\_Opaque\_Type\_Alloc, 229  
     Oscl\_Opaque\_Type\_Alloc\_LL, 231  
     Oscl\_TAlloc, 282  
     OsclErrorAllocator, 374  
     OsclMemAllocator, 426  
     OsclMemAllocDestructDealloc, 427  
     OSCLMemAutoPtr, 437  
     OsclMemBasicAllocator, 439  
     OsclMemBasicAllocDestructDealloc, 440  
     OsclMemPoolFixedChunkAllocator, 444  
     OsclMemPoolResizableAllocator, 450  
     OsclReadyAlloc, 482  
 allocate\_fl  
     Oscl\_Alloc, 170  
     Oscl\_DefAlloc, 172  
     Oscl\_TAlloc, 282  
     OsclMemAllocator, 426  
     OsclMemAllocDestructDealloc, 427  
     OsclReadyAlloc, 482  
 allocateblock  
     OsclMemPoolResizableAllocator, 450  
 allocator, 113  
 allocNum  
     MM\_AllocInfo, 150  
     MM\_AllocQueryInfo, 152  
 AllPassFilter, 114  
     AllPassFilter, 115  
 AllPassFilter  
     ~AllPassFilter, 115  
     AllPassFilter, 115  
     filter\_status\_type, 114  
     FilterOpaqueMessge, 115  
     FilterString, 115  
     log\_level\_type, 114  
     message\_id\_type, 114  
 ALREADY\_SUSPENDED\_ERROR  
     OsclProcStatus, 475  
 Append  
     OsclPtr, 477  
 append

CFastRep, 128  
 CHheapRep, 130  
 CStackRep, 132  
**APPEND\_MEDIA\_AT\_END**  
 osclutil, 84  
**append\_rep**  
 CHheapRep, 130  
 OSCL\_String, 261  
 OSCL\_wString, 305  
**AppendBuffers**  
 PVLoggerAppender, 623  
**AppendNext**  
 BufFragGroup, 121  
**AppendString**  
 PVLoggerAppender, 623  
**assign**  
 CHheapRep, 130  
**assign\_vector**  
 Oscl\_Vector\_Base, 291  
**asyncfilereadcancel\_test**  
 Oscl\_File, 186  
**asyncfilereadwrite\_test**  
 Oscl\_File, 186  
**Attach**  
 OsclBinStream, 337  
**audit\_type**  
 OsclMemGlobalAuditObject, 441  
**available\_localbuf**  
 MediaData, 144  
  
**back**  
 Oscl\_Queue, 237  
 Oscl\_Vector, 287  
**BAD\_THREADID\_ADDR\_ERROR**  
 OsclProcStatus, 475  
**base\_link\_type**  
 Oscl\_Rb\_Tree\_Base, 246  
 Oscl\_Rb\_Tree\_Const\_Iterator, 248  
 Oscl\_Rb\_Tree\_Iterator, 251  
 Oscl\_Rb\_Tree\_Node\_Base, 254  
**begin**  
 Oscl\_Map, 219  
 Oscl\_Rb\_Tree, 244  
 Oscl\_TagTree, 270  
 Oscl\_Vector, 287  
**BeginScheduling**  
 OsclExecSchedulerCommonBase, 395  
**BeginStats**  
 OsclExecSchedulerCommonBase, 395  
**BFG\_SUCCESS**  
 BufFragStatusClass, 123  
**big\_endian\_to\_host**  
 osclbase, 36  
**Bind**  
 osclbase, 36  
 OsclBindMethod, 322  
 OsclBindRequest, 323  
 OsclIPSocketI, 419  
 OsclSocketI, 536  
 OsclSocketIBase, 541  
 OsclTCPSocket, 566  
 OsclUDPSocket, 602  
**bind**  
 BufferState, 119  
**BindAsync**  
 OsclSocketIBase, 541  
 OsclTCPSocket, 566  
 OsclTCPSocketI, 573  
 OsclUDPSocket, 602  
 OsclUDPSocketI, 608  
**BindParam**, 116  
 BindParam, 116  
**BindParam**  
 BindParam, 116  
 iAddr, 116  
**BindRequest**  
 OsclBindMethod, 322  
**black**  
 Oscl\_Rb\_Tree\_Node\_Base, 254  
**BlockingLoopL**  
 OsclExecSchedulerCommonBase, 395  
**bSetFailure**  
 MM\_AllocInfo, 150  
**Buffer**  
 OsclAsyncFileBuffer, 320  
**buffer**  
 CFastRep, 128  
 CHheapRep, 130  
 CStackRep, 132  
**buffer\_states**  
 BufFragGroup, 122  
 BufferFragment, 117  
 BufferFreeFuncPtr  
 osclutil, 69  
 BufferMgr, 118  
**BufferMgr**  
 ~BufferMgr, 118  
 BufferReleased, 118  
**BufferReleased**  
 BufferMgr, 118  
**BufferState**, 119  
 BufferState, 119  
**BufferState**  
 bind, 119  
 BufferState, 119  
 decrement\_refcnt, 119  
 get\_buf\_mgr, 119  
 get\_free\_function, 119

get\_ptr, 119  
 get\_refcount, 119  
 increment\_refcnt, 119  
 reset, 119  
**BuFragGroup**, 120  
     BuFragGroup, 121  
**BuFragGroup**  
     ~BuFragGroup, 121  
     AddFragment, 121  
     AppendNext, 121  
     buffer\_states, 122  
     BuFragGroup, 121  
     Clear, 121  
     fragments, 122  
     GetLength, 121  
     GetMaxFrags, 122  
     GetNext, 122  
     GetNumFrags, 122  
     length, 122  
     next, 122  
     num\_fragments, 122  
**BuFragStatusClass**, 123  
     BFG\_SUCCESS, 123  
     EMPTY\_FRAGMENT, 123  
     FIXED\_FRAG\_LOC\_FULL, 123  
     INTERNAL\_ERROR, 123  
     INVALID\_ID, 123  
     NOT\_ENOUGH\_SPACE, 123  
     NULL\_INPUT, 123  
     TOO\_MANY\_FRAGS, 123  
**BuFragStatusClass**  
     status\_t, 123  
**bufsize**  
     Oscl\_Queue\_Base, 241  
     Oscl\_Vector\_Base, 293  
**BYTES\_IN\_UUID\_ARRAY**  
     oscl\_uuid.h, 799

**c**  
     OsclPriorityQueue, 473  
**c\_bool**  
     osclbase, 34  
**c\_str**  
     StrPtrLen, 648  
     WStrPtrLen, 659  
**Callback**  
     OsclReadyQ, 485  
**callback\_timer\_type**  
     OsclTimer, 583  
**CallbackTimer**, 124  
     CallbackTimer, 124  
**CallbackTimer**  
     ~CallbackTimer, 124  
     CallbackTimer, 124  
**Run**, 124  
**CallbackTimer< Alloc >**  
     OsclTimer, 584  
**CallbackTimerObserver**, 126  
**CallbackTimerObserver**  
     ~CallbackTimerObserver, 126  
     TimerBaseElapsed, 126  
**CallRunExec**  
     OsclExecSchedulerCommonBase, 395  
**Cancel**  
     OsclActiveObject, 311  
     OsclTimer, 583  
     OsclTimerObject, 587  
     PVActiveBase, 613  
**CancelAccept**  
     OsclSocketIBase, 542  
     OsclTCPSocket, 567  
     OsclTCPSocketI, 573  
**CancelBind**  
     OsclSocketIBase, 542  
     OsclTCPSocket, 567  
     OsclTCPSocketI, 573  
     OsclUDPSocket, 602  
     OsclUDPSocketI, 608  
**CancelConnect**  
     OsclSocketIBase, 542  
     OsclTCPSocket, 567  
     OsclTCPSocketI, 573  
**CancelFreeChunkAvailableCallback**  
     OsclMemPoolFixedChunkAllocator, 444  
     OsclMemPoolResizableAllocator, 450  
**CancelFreeMemoryAvailableCallback**  
     OsclMemPoolResizableAllocator, 450  
**CancelFxn**  
     OsclDNSIBase, 356  
     OsclSocketIBase, 542  
**CancelGetHostName**  
     OsclDNS, 351  
     OsclDNSIBase, 356  
**Cancelled**  
     OsclDNSRequestAO, 364  
**CancelListen**  
     OsclSocketIBase, 542  
     OsclTCPSocket, 567  
     OsclTCPSocketI, 573  
**CancelMethod**  
     OsclDNSMethod, 359  
     OsclSocketMethod, 546  
**CancelRecv**  
     OsclSocketIBase, 542  
     OsclTCPSocket, 567  
     OsclTCPSocketI, 573  
**CancelRecvFrom**  
     OsclSocketIBase, 542

OsclUDPSocket, 602  
 OsclUDPSocketI, 608  
**CancelRequest**  
 OsclIDNSRequest, 362  
 OsclSocketRequest, 549  
**CancelSend**  
 OsclSocketIBase, 542  
 OsclTCPSocket, 567  
 OsclTCPSocketI, 573  
**CancelSendTo**  
 OsclSocketIBase, 542  
 OsclUDPSocket, 602  
 OsclUDPSocketI, 608  
**CancelShutdown**  
 OsclSocketIBase, 542  
 OsclTCPSocket, 567  
 OsclTCPSocketI, 573  
**canPersistMoreHostAddresses**  
 GetHostNameParam, 136  
**CanTerminate**  
 OsclThread, 575  
**capacity**  
 Oscl\_Queue\_Base, 240  
 Oscl\_Vector\_Base, 291  
 OsclFileCacheBuffer, 404  
**CFastRep**, 127  
 CFastRep, 128  
**CFastRep**  
 append, 128  
 buffer, 128  
 CFastRep, 128  
 maxsize, 128  
 overwrite, 128  
 set\_r, 128  
 set\_w, 128  
 size, 128  
 writable, 128  
**chartype**  
 OSCL\_FastString, 176  
 OSCL\_HeapString, 197  
 OSCL\_HeapStringA, 199  
 OSCL\_StackString, 258  
 OSCL\_String, 261  
 OSCL\_wFastString, 294  
 OSCL\_wHeapString, 298  
 OSCL\_wHeapStringA, 300  
 OSCL\_wStackString, 303  
 OSCL\_wString, 305  
**CHeapRep**, 129  
 CHeapRep, 130  
**CHeapRep**  
 add\_ref, 130  
 append, 130  
 append\_rep, 130  
 assign, 130  
 buffer, 130  
 CHheapRep, 130  
 maxsize, 130  
 refcount, 130  
 remove\_ref, 130  
 set, 130  
 set\_rep, 130  
 size, 130  
**check\_fence**  
 MM\_AllocBlockFence, 147  
**check\_list**  
 Oscl\_Linked\_List, 207  
 Oscl\_Linked\_List\_Base, 212  
**checkSum**  
 StrCSumPtrLen, 645  
**CheckSumType**  
 StrCSumPtrLen, 645  
**children**  
 Oscl\_TagTree::Node, 280  
**children\_type**  
 Oscl\_TagTree, 270  
 Oscl\_TagTree::Node, 280  
**ChooseCurCache**  
 Oscl\_File::OsclCacheObserver, 187  
**CleanInUse**  
 OsclAsyncFileBuffer, 320  
**Cleanup**  
 OsclErrorTrap, 376  
 OsclInit, 415  
 OsclMem, 425  
 OsclScheduler, 516  
 PVLogger, 619  
**CleanupExecQ**  
 OsclExecSchedulerCommonBase, 395  
**CleanupParam**  
 OsclSocketRequestAO, 551  
**CleanupStatQ**  
 OsclExecSchedulerCommonBase, 395  
**Clear**  
 BufFragGroup, 121  
 MediaData, 143  
 OsclTimer, 583  
**clear**  
 Oscl\_Linked\_List, 207  
 Oscl\_Map, 219  
 Oscl\_Queue, 237  
 Oscl\_Queue\_Base, 240  
 Oscl\_Rb\_Tree, 244  
 Oscl\_TagTree, 271  
 Oscl\_Vector, 287  
**ClearTOS**  
 OsclSocketTOS, 564  
**Close**

Oscl\_File, 181  
 Oscl\_FileFind, 190  
 Oscl\_FileServer, 193  
 OsclAsyncFile, 317  
 OsclDNSI, 353  
 OsclDNSIBase, 356  
 OsclFileCache, 402  
 OsclIPSocketI, 419  
 OsclMutex, 459  
 OsclNativeFile, 463  
 OsclRegistryAccessClient, 504  
 OsclRegistryClient, 509  
 OsclRegistryClientImpl, 512  
 OsclRegistryServTlsImpl, 515  
 OsclSemaphore, 521  
 OsclSocketI, 536  
 OsclSocketIBase, 542  
 OsclSocketServ, 554  
 OsclSocketServI, 556  
 OsclSocketServIBase, 559  
 OsclSocketServRequestList, 560  
 OsclTCPSocket, 568  
 OsclTCPSocketI, 573  
 OsclUDPSocket, 603  
 OsclUDPSocketI, 608  
**CloseSession**  
 OsclComponentRegistry, 344  
**color**  
 Oscl\_Rb\_Tree\_Node\_Base, 255  
**color\_type**  
 Oscl\_Rb\_Tree\_Node\_Base, 254  
**comp**  
 Oscl\_Map::value\_compare, 223  
 OsclPriorityQueue, 473  
**compare**  
 OsclCompareLess, 342  
 OsclReadyCompare, 483  
 OsclTimerCompare, 585  
**compare\_data**  
 Oscl\_Opaque\_Type\_Alloc\_LL, 231  
**compare\_EQ**  
 Oscl\_Opaque\_Type\_Compare, 233  
 OsclPriorityQueue, 471  
**compare\_LT**  
 Oscl\_Opaque\_Type\_Compare, 233  
 OsclPriorityQueue, 471  
**CompareId**  
 OsclThread, 576  
**Complete**  
 OsclDNSRequest, 362  
 OsclSocketRequest, 549  
**COMPUTE\_MEM\_ALIGN\_SIZE**  
 osclmemory, 51  
**Connect**  
 Oscl\_FileServer, 193  
 OsclConnectMethod, 348  
 OsclConnectRequest, 349  
 OsclRegistryAccessClient, 504  
 OsclRegistryClient, 509  
 OsclRegistryClientImpl, 512  
 OsclRegistryServTlsImpl, 515  
 OsclSocketI, 536  
 OsclSocketIBase, 542  
 OsclSocketServ, 554  
 OsclSocketServI, 556  
 OsclSocketServIBase, 559  
 OsclTCPSocket, 568  
 OsclTCPSocketI, 573  
 ConnectParam, 131  
 ConnectParam, 131  
**ConnectParam**  
 ConnectParam, 131  
 iAddr, 131  
**ConnectRequest**  
 OsclConnectMethod, 348  
**const\_iterator**  
 Oscl\_Map, 218  
 Oscl\_Rb\_Tree, 244  
 Oscl\_Rb\_Tree\_Const\_Iterator, 248  
 Oscl\_TagTree::const\_iterator, 274  
 Oscl\_Vector, 286  
**const\_pointer**  
 Oscl\_Rb\_Tree, 244  
 Oscl\_TAlloc, 282  
**const\_reference**  
 Oscl\_Map, 218  
 Oscl\_Queue, 237  
 Oscl\_Rb\_Tree, 244  
 Oscl\_TAlloc, 282  
 Oscl\_Vector, 286  
 OsclPriorityQueue, 471  
**Construct**  
 OsclReadyQ, 485  
 OsclTimerQ, 591  
**construct**  
 Oscl\_Linked\_List\_Base, 212  
 Oscl\_Opaque\_Type\_Alloc, 229  
 Oscl\_Opaque\_Type\_Alloc\_LL, 231  
 Oscl\_Queue\_Base, 240  
 Oscl\_TAlloc, 282  
 Oscl\_Vector\_Base, 291  
 OsclPriorityQueueBase, 474  
**ConstructL**  
 OsclDNSMethod, 359  
 OsclDNSRequestAO, 364  
 OsclExecSchedulerCommonBase, 395  
 OsclIPSocketI, 419  
 OsclSocketMethod, 546

OsclSocketRequestAO, 551  
 ConstructStatQ  
   OsclExecSchedulerCommonBase, 395  
 container\_type  
   OsclPriorityQueue, 471  
 Contains  
   Oscl\_File::OsclFixedCacheParam, 188  
   OsclFileCacheBuffer, 404  
 count  
   Oscl\_Map, 219  
   Oscl\_Rb\_Tree, 244  
   Oscl\_TagTree, 271  
 CPVInterfaceProxy  
   OsclErrorTrapImp, 378  
 Create  
   GetHostNameParam, 136  
   OsclMutex, 459  
   OsclSemaphore, 521  
   OsclThread, 576  
 createmempool  
   OsclMemPoolFixedChunkAllocator, 444  
 CreatePVLogger  
   PVLoggerRegistry, 629  
 createStatsNode  
   MM\_Audit\_Imp, 154  
 CStackRep, 132  
   CStackRep, 132  
 CStackRep  
   append, 132  
   buffer, 132  
   CStackRep, 132  
   maxsize, 132  
   set, 132  
   size, 132  
 CTIME\_BUFFER\_SIZE  
   osclbase, 46  
 CtimeStrBuf  
   osclbase, 34  
 Current  
   OsclExecScheduler, 389  
 currentPos  
   OsclFileCacheBuffer, 404  
  
 data  
   LinkedListElement, 140  
 data1  
   OsclUuid, 611  
 data2  
   OsclUuid, 611  
 data3  
   OsclUuid, 611  
 data4  
   OsclUuid, 611  
 deallocate  
  
   \_OsclBasicAllocator, 109  
   MemAllocator, 146  
   Oscl\_Dealloc, 171  
   Oscl\_DefAlloc, 172  
   Oscl\_Opaque\_Type\_Alloc, 229  
   Oscl\_Opaque\_Type\_Alloc\_LL, 231  
   Oscl\_TAlloc, 282  
   OsclErrorAllocator, 374  
   OsclMemAllocator, 426  
   OsclMemAllocDestructDealloc, 427  
   OSCLMemAutoPtr, 437  
   OsclMemBasicAllocator, 439  
   OsclMemBasicAllocDestructDealloc, 440  
   OsclMemPoolFixedChunkAllocator, 445  
   OsclMemPoolResizableAllocator, 450  
   OsclReadyAlloc, 482  
 deallocateblock  
   OsclMemPoolResizableAllocator, 450  
 decrement\_refcnt  
   BufferState, 119  
 DEFAULT\_MM\_AUDIT\_MODE  
   osclmemory, 52  
 DEFAULT\_POSTFILL\_PATTERN  
   osclmemory, 52  
 DEFAULT\_PREFILL\_PATTERN  
   osclmemory, 52  
 Delete  
   Oscl\_DefAllocWithRefCounter, 173  
   OsclAsyncFile, 317  
   OsclBuf, 341  
 Depth  
   OsclReadyQ, 485  
 depth  
   Oscl\_TagTree::Node, 280  
 dequeue\_element  
   Oscl\_Linked\_List, 207  
   Oscl\_MTLLinked\_List, 226  
 Des  
   OsclBuf, 341  
 DesC  
   OsclBuf, 341  
 Destroy  
   DNSRequestParam, 133  
   GetHostNameParam, 136  
   PVActiveBase, 613  
 destroy  
   Oscl\_Linked\_List\_Base, 212  
   Oscl\_Opaque\_Type\_Alloc, 229  
   Oscl\_Opaque\_Type\_Alloc\_LL, 232  
   Oscl\_Queue\_Base, 240  
   Oscl\_TAlloc, 282  
   Oscl\_Vector, 287  
   Oscl\_Vector\_Base, 291  
 destroyallmempoolbuffers

OsclMemPoolResizableAllocator, 450  
 destroymempool  
   OsclMemPoolFixedChunkAllocator, 445  
 destruct\_and\_dealloc  
   Oscl\_TAlloc, 282  
   OsclDestructDealloc, 350  
   OsclMemAllocDestructDealloc, 427  
   OsclMemBasicAllocDestructDealloc, 440  
 difference\_type  
   Oscl\_Rb\_Tree, 244  
 DIR\_TYPE  
   Oscl\_FileFind, 189  
 DisableAppenderInheritance  
   PVLogger, 619  
 DiscardAcceptedSocket  
   OsclAcceptMethod, 308  
 DNSRequestParam, 133  
   DNSRequestParam, 133  
   OsclIDNSI, 354  
 DNSRequestParam  
   ~DNSRequestParam, 133  
   Destroy, 133  
   DNSRequestParam, 133  
   iDNSRequest, 134  
   iFxn, 134  
   InThread, 133  
   iRefCount, 134  
   RemoveRef, 134  
 DoCancel  
   OsclActiveObject, 312  
   OsclIDNSRequestAO, 364  
   OsclSocketRequestAO, 551  
   OsclTimerObject, 587  
   PVActiveBase, 613  
  
 E\_BUFFER\_TOO\_SMALL  
   Oscl\_FileFind, 190  
 E\_INVALID\_ARG  
   Oscl\_FileFind, 189  
 E\_INVALID\_STATE  
   Oscl\_FileFind, 189  
 E\_MEMORY\_ERROR  
   Oscl\_FileFind, 190  
 E\_NO\_MATCH  
   Oscl\_FileFind, 190  
 E\_NOT\_IMPLEMENTED  
   Oscl\_FileFind, 190  
 E\_OK  
   Oscl\_FileFind, 189  
 E\_OTHER  
   Oscl\_FileFind, 190  
 E\_PATH\_NOT\_FOUND  
   Oscl\_FileFind, 189  
 E\_PATH\_TOO\_LONG

Oscl\_FileFind, 189  
 element\_type  
   Oscl\_FileFind, 189  
 elems  
   Oscl\_Queue\_Base, 241  
   Oscl\_Vector\_Base, 293  
 empty  
   Oscl\_Map, 219  
   Oscl\_Queue\_Base, 240  
   Oscl\_Rb\_Tree, 244  
   Oscl\_TagTree, 271  
   Oscl\_Vector\_Base, 291  
   OsclPriorityQueue, 472  
 EMPTY\_FRAGMENT  
   BuffFragStatusClass, 123  
 EMPTY\_UUID  
   oscl\_uuid.h, 799  
 enablenullpointerreturn  
   OsclMemPoolFixedChunkAllocator, 445  
   OsclMemPoolResizableAllocator, 450  
 End  
   OsclFileStats, 411  
 end  
   Oscl\_Map, 219  
   Oscl\_Rb\_Tree, 244  
   Oscl\_TagTree, 271  
   Oscl\_Vector, 287  
 EndOfFile  
   Oscl\_File, 181  
   OsclAsyncFile, 317  
   OsclFileCache, 402  
   OsclNativeFile, 463  
 endPos  
   OsclFileCacheBuffer, 404  
 EndScheduling  
   OsclExecSchedulerCommonBase, 395  
 EndStats  
   OsclExecSchedulerCommonBase, 395  
 EnterThreadContext  
   PVThreadContext, 634  
 eof  
   OsclBinStream, 337  
 EOF\_STATE  
   OsclBinStream, 337  
 EOSCL\_StringOp\_CompressASCII  
   osclutil, 70  
 EOSCL\_StringOp\_UTF16ToUTF8  
   osclutil, 70  
 EOSCL\_wStringOp\_ExpandASCII  
   osclutil, 70  
 EOSCL\_wStringOp\_UTF8ToUTF16  
   osclutil, 70  
 EOscFileOp\_Close  
   osclio, 98

EOscIFileOp\_EndOfFile  
     osclio, [98](#)  
 EOscIFileOp\_Flush  
     osclio, [98](#)  
 EOscIFileOp\_Last  
     osclio, [99](#)  
 EOscIFileOp\_NativeClose  
     osclio, [98](#)  
 EOscIFileOp\_NativeEndOfFile  
     osclio, [99](#)  
 EOscIFileOp\_NativeFlush  
     osclio, [99](#)  
 EOscIFileOp\_NativeOpen  
     osclio, [98](#)  
 EOscIFileOp\_NativeRead  
     osclio, [98](#)  
 EOscIFileOp\_NativeSeek  
     osclio, [99](#)  
 EOscIFileOp\_NativeSetSize  
     osclio, [99](#)  
 EOscIFileOp\_NativeSize  
     osclio, [99](#)  
 EOscIFileOp\_NativeTell  
     osclio, [99](#)  
 EOscIFileOp\_NativeWrite  
     osclio, [99](#)  
 EOscIFileOp\_Open  
     osclio, [98](#)  
 EOscIFileOp\_Read  
     osclio, [98](#)  
 EOscIFileOp\_Seek  
     osclio, [98](#)  
 EOscIFileOp\_SetSize  
     osclio, [98](#)  
 EOscIFileOp\_Size  
     osclio, [98](#)  
 EOscIFileOp\_Tell  
     osclio, [98](#)  
 EOscIFileOp\_Write  
     osclio, [98](#)  
 eOsclProcError  
     OsclProcStatus, [475](#)  
 EOscISocket\_DataRecv  
     oscl\_socket\_stats.h, [771](#)  
 EOscISocket\_DataSent  
     oscl\_socket\_stats.h, [771](#)  
 EOscISocket\_Except  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocket\_OS  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocket\_Readable  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocket\_RequestAO\_Canceled  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocket\_RequestAO\_Error  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocket\_RequestAO\_Success  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocket\_RequestAO\_Timeout  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocket\_ServPoll  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocket\_ServRequestCancelIssued  
     oscl\_socket\_stats.h, [771](#)  
 EOscISocket\_ServRequestComplete  
     oscl\_socket\_stats.h, [771](#)  
 EOscISocket\_ServRequestIssued  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocket\_Writable  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocketServ\_LastEvent  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocketServ\_LoopsockError  
     oscl\_socket\_stats.h, [771](#)  
 EOscISocketServ\_LoopsockOk  
     oscl\_socket\_stats.h, [771](#)  
 EOscISocketServ\_SelectActivity  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocketServ\_SelectNoActivity  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocketServ\_SelectRescheduleAsap  
     oscl\_socket\_stats.h, [770](#)  
 EOscISocketServ\_SelectReschedulePoll  
     oscl\_socket\_stats.h, [770](#)  
 EOscIThreadTerminate\_Join  
     oscl\_thread.h, [789](#)  
 EOscIThreadTerminate\_Kill  
     oscl\_thread.h, [789](#)  
 EOscIThreadTerminate\_NOP  
     oscl\_thread.h, [789](#)  
 EOtherExecStats\_Last  
     OsclExecSchedulerCommonBase, [394](#)  
 EOtherExecStats\_NativeOS  
     OsclExecSchedulerCommonBase, [394](#)  
 EOtherExecStats\_QueueTime  
     OsclExecSchedulerCommonBase, [394](#)  
 EOtherExecStats\_ReleaseTime  
     OsclExecSchedulerCommonBase, [394](#)  
 EOtherExecStats\_WaitTime  
     OsclExecSchedulerCommonBase, [394](#)  
 EPriorityHigh  
     OsclActiveObject, [311](#)  
 EPriorityHighest  
     OsclActiveObject, [311](#)  
 EPriorityIdle  
     OsclActiveObject, [311](#)  
 EPriorityLow  
     OsclActiveObject, [311](#)

EPriorityNominal  
     OsclActiveObject, 311  
 EPV\_ARM\_GNUC  
     osclbase, 32  
 EPV\_ARM\_MSEVC  
     osclbase, 32  
 EPV\_ARM\_RVCT  
     osclbase, 32  
 EPVCritic\_Ecp  
     OsclSocketTOS, 563  
 EPVDNSCancel  
     osclio, 99  
 EPVDNSFailure  
     osclio, 99  
 EPVDNSGetHostName  
     osclio, 99  
 EPVDNSPending  
     osclio, 99  
 EPVDNSSuccess  
     osclio, 99  
 EPVDNSTimeout  
     osclio, 99  
 EPVFlash  
     OsclSocketTOS, 563  
 EPVHiRel  
     OsclSocketTOS, 563  
 EPVHiThrpt  
     OsclSocketTOS, 563  
 EPVImmediate  
     OsclSocketTOS, 563  
 EPVInetControl  
     OsclSocketTOS, 563  
 EPVIPAddMembership  
     oscl\_socket\_types.h, 775  
 EPVIPMulticastTTL  
     oscl\_socket\_types.h, 775  
 EPVIPProtoIP  
     oscl\_socket\_types.h, 775  
 EPVIPProtoTCP  
     oscl\_socket\_types.h, 775  
 EPVIPTOS  
     oscl\_socket\_types.h, 775  
 EPVLDelay  
     OsclSocketTOS, 563  
 EPVNetControl  
     OsclSocketTOS, 563  
 EPVNoTOS  
     OsclSocketTOS, 563  
 EPVOverrideFlash  
     OsclSocketTOS, 563  
 EPVPriority  
     OsclSocketTOS, 563  
 EPVRoutine  
     OsclSocketTOS, 563

EPVSocket  
     oscl\_socket\_types.h, 775  
 EPVSocket\_Last  
     oscl\_socket\_types.h, 775  
 EPVSocketAccept  
     oscl\_socket\_types.h, 775  
 EPVSocketBind  
     oscl\_socket\_types.h, 775  
 EPVSocketBothShutdown  
     oscl\_socket\_types.h, 775  
 EPVSocketCancel  
     oscl\_socket\_types.h, 774  
 EPVSocketConnect  
     oscl\_socket\_types.h, 775  
 EPVSocketFailure  
     oscl\_socket\_types.h, 774  
 EPVSocketListen  
     oscl\_socket\_types.h, 775  
 EPVSocketNotImplemented  
     oscl\_socket\_types.h, 775  
 EPVSocketPending  
     oscl\_socket\_types.h, 774  
 EPVSocketRecv  
     oscl\_socket\_types.h, 775  
 EPVSocketRecvFrom  
     oscl\_socket\_types.h, 775  
 EPVSocketRecvShutdown  
     oscl\_socket\_types.h, 775  
 EPVSocketSend  
     oscl\_socket\_types.h, 775  
 EPVSocketSendShutdown  
     oscl\_socket\_types.h, 775  
 EPVSocketSendTo  
     oscl\_socket\_types.h, 775  
 EPVSocketShutdown  
     oscl\_socket\_types.h, 775  
 EPVSocketSuccess  
     oscl\_socket\_types.h, 774  
 EPVSocketTimeout  
     oscl\_socket\_types.h, 774  
 EPVSockReuseAddr  
     oscl\_socket\_types.h, 775  
 EPVThreadContext\_InThread  
     osclproc, 106  
 EPVThreadContext\_NonOsclThread  
     osclproc, 106  
 EPVThreadContext\_OsclThread  
     osclproc, 106  
 EPVThreadContext\_Undetermined  
     osclproc, 106  
 equal\_range  
     Oscl\_Map, 219  
     Oscl\_Rb\_Tree, 244  
 erase

Oscl\_Map, 220  
 Oscl\_Rb\_Tree, 244  
 Oscl\_TagTree, 271  
 Oscl\_Vector, 287  
 Oscl\_Vector\_Base, 291, 292  
**Error**  
   OsclExecSchedulerCommonBase, 395  
**error\_type**  
   Oscl\_FileFind, 189  
**ESocketServ\_Connected**  
   OsclSocketServIBase, 558  
**ESocketServ\_Error**  
   OsclSocketServIBase, 559  
**ESocketServ\_Idle**  
   OsclSocketServIBase, 558  
**ESymbianAccessMode\_Rfile**  
   Oscl\_File, 180  
**ESymbianAccessMode\_RfileBuf**  
   Oscl\_File, 180  
**EXCEED\_MAX\_COUNT\_VARIABLE\_ERROR**  
   OsclProcStatus, 476  
**EXCEED\_MAX\_SEM\_COUNT\_ERROR**  
   OsclProcStatus, 476  
**Exit**  
   OsclThread, 576  
**ExitThreadContext**  
   PVThreadContext, 634  
**extract\_string**  
   osclutil, 70  
  
**fail**  
   OsclBinStream, 338  
**FAIL\_STATE**  
   OsclBinStream, 337  
**Failure**  
   OsclIDNSRequestAO, 364  
**FENCE\_PATTERN**  
   osclmemory, 52  
**FILE\_TYPE**  
   Oscl\_FileFind, 189  
**fileName**  
   MM\_AllocQueryInfo, 152  
**filePosition**  
   OsclFileCacheBuffer, 404  
**FileSize**  
   OsclFileCache, 402  
**fill\_fence**  
   MM\_AllocBlockFence, 147  
**FillFromFile**  
   OsclFileCacheBuffer, 404  
**filter\_status\_type**  
   AllPassFilter, 114  
   PVLogger, 618  
  
 PVLoggerFilter, 624  
 FilterOpaqueMessage  
   AllPassFilter, 115  
   PVLoggerFilter, 625  
 FilterString  
   AllPassFilter, 115  
   PVLoggerFilter, 625  
**Find**  
   OsclComponentRegistryData, 345  
**find**  
   Oscl\_Map, 220  
   Oscl\_Rb\_Tree, 244  
   Oscl\_TagTree, 271  
**find\_heap**  
   OsclPriorityQueue, 472  
   OsclPriorityQueueBase, 474  
**FindExact**  
   OsclComponentRegistry, 344  
**FindFirst**  
   Oscl\_FileFind, 190  
**findfreeblock**  
   OsclMemPoolResizableAllocator, 451  
**FindHierarchical**  
   OsclComponentRegistry, 344  
**FindNext**  
   Oscl\_FileFind, 191  
**FindPVA**  
   OsclExecSchedulerCommonBase, 395  
**first**  
   Oscl\_Pair, 235  
**firstFragPtr**  
   OsclBinStream, 339  
**FIXED\_FRAG\_LOC\_FULL**  
   BuffFragStatusClass, 123  
**Flush**  
   Oscl\_File, 182  
   OsclAsyncFile, 317  
   OsclFileCache, 402  
   OsclNativeFile, 463  
**FormatOpaqueMessage**  
   PVLoggerLayout, 626  
**FormatString**  
   PVLoggerLayout, 626  
**fragments**  
   BuffFragGroup, 122  
**fragsLeft**  
   OsclBinStream, 339  
**freeblockavailable**  
   OsclMemPoolResizableAllocatorObserver, 457  
**freebytes**  
   oscl\_fsstat, 195  
**freechunkavailable**

OsclMemPoolFixedChunkAllocator-  
   Observer, 447  
 freememoryavailable  
   OsclMemPoolResizableAllocatorMemory-  
     Observer, 456  
 front  
   Oscl\_Queue, 238  
   Oscl\_Vector, 288  
 Fxn  
   OsclSocketRequest, 549  
  
 get  
   OsclBinIStream, 324  
   OsclExclusiveArrayPtr, 381  
   OsclExclusivePtr, 384  
   OsclExclusivePtrA, 387  
   OSCLMemAutoPtr, 437  
 get\_buf\_mgr  
   BufferState, 119  
 get\_count  
   OsclSharedPtr, 528  
 get\_cstr  
   OSCL\_FastString, 177  
   OSCL\_HeapStringA, 200  
   OSCL\_String, 261  
   OSCL\_wFastString, 295  
   OSCL\_wHeapStringA, 300  
   OSCL\_wString, 305  
   osclutil, 70  
 get\_data  
   Oscl\_Opaque\_Type\_Alloc\_LL, 232  
 get\_element  
   Oscl\_Linked\_List, 207  
   Oscl\_Linked\_List\_Base, 212  
   Oscl\_MTLLinked\_List, 226  
 get\_first  
   Oscl\_Linked\_List, 208  
   Oscl\_Linked\_List\_Base, 213  
 get\_free\_function  
   BufferState, 119  
 get\_index  
   Oscl\_Linked\_List, 208  
   Oscl\_Linked\_List\_Base, 213  
   Oscl\_MTLLinked\_List, 226  
 get\_int64\_lower32  
   Oscl\_Int64\_Utils, 204  
 get\_int64\_middle32  
   Oscl\_Int64\_Utils, 204  
 get\_int64\_upper32  
   Oscl\_Int64\_Utils, 204  
 get\_ISO8601\_str\_time  
   TimeValue, 652  
 get\_local\_time  
   TimeValue, 652  
  
 get\_lower32  
   NTPTime, 168  
 get\_maxsize  
   OSCL\_FastString, 177  
   OSCL\_HeapStringA, 200  
   OSCL\_String, 261  
   OSCL\_wFastString, 295  
   OSCL\_wHeapStringA, 300  
   OSCL\_wString, 305  
   osclutil, 71  
 get\_middle32  
   NTPTime, 168  
 get\_next  
   Oscl\_Linked\_List, 208  
   Oscl\_Linked\_List\_Base, 213  
   Oscl\_Opaque\_Type\_Alloc\_LL, 232  
 get\_num\_elements  
   Oscl\_Linked\_List, 208  
 get\_ptr  
   BufferState, 119  
 get\_pv8601\_str\_time  
   TimeValue, 652  
 get\_refcount  
   BufferState, 119  
 get\_registry  
   TLSStorageOps, 656  
 get\_rfc822\_gmtime\_str  
   TimeValue, 652  
 get\_sec  
   TimeValue, 653  
 get\_size  
   OSCL\_FastString, 177  
   OSCL\_HeapStringA, 201  
   OSCL\_String, 262  
   OSCL\_wFastString, 295  
   OSCL\_wHeapStringA, 300  
   OSCL\_wString, 305  
   osclutil, 71  
 get\_str  
   OSCL\_FastString, 177  
   OSCL\_HeapStringA, 201  
   OSCL\_String, 262  
   OSCL\_wFastString, 295  
   OSCL\_wHeapStringA, 301  
   OSCL\_wString, 305  
   osclutil, 72  
 get\_str\_ctime  
   TimeValue, 653  
 get\_timeval\_ptr  
   TimeValue, 653  
 get\_timevalue\_in\_usec  
   TimeValue, 653  
 get\_uint64\_lower32  
   Oscl\_Int64\_Utils, 204

---

get\_uint64\_middle32  
     Oscl\_Int64\_Utils, 204  
 get\_uint64\_upper32  
     Oscl\_Int64\_Utils, 204  
 get\_upper32  
     NTPTime, 168  
 get\_usec  
     TimeValue, 653  
 get\_value  
     NTPTime, 168  
 GetAcceptedSocket  
     OsclAcceptMethod, 308  
 GetAcceptedSocketL  
     OsclTCPSocket, 568  
     OsclTCPSocketI, 573  
 getAllocatedSize  
     OsclMemPoolResizableAllocator, 451  
 getAuditRoot  
     MM\_Audit\_Imp, 154  
 GetAvailableBufferSize  
     MediaData, 143  
 getAvailableSize  
     OsclMemPoolResizableAllocator, 451  
 getBufferSize  
     OsclMemPoolResizableAllocator, 451  
 GetBufferState  
     osclutil, 72  
 getCapacity  
     OsclRefCounterMemFrag, 497  
 getCheckSum  
     StrCSumPtrLen, 645  
 getCount  
     Oscl\_DefAllocWithRefCounter, 173  
     OsclRefCounter, 492  
     OsclRefCounterDA, 495  
     OsclRefCounterMemFrag, 497  
     OsclRefCounterMTDA, 499  
     OsclRefCounterMTSA, 501  
     OsclRefCounterSA, 503  
 GetElementType  
     Oscl\_FileFind, 191  
 GetError  
     Oscl\_File, 182  
     OsclNativeFile, 463  
 GetErrorTrapImp  
     OsclErrorTrap, 376  
 GetFactories  
     OsclRegistryAccessClient, 504  
     OsclRegistryClientImpl, 512  
     OsclRegistryServTlsImpl, 515  
 GetFactory  
     OsclRegistryAccessClient, 504  
     OsclRegistryClientImpl, 512  
     OsclRegistryServTlsImpl, 515  
 GetFragment  
     osclutil, 72  
 getGlobalMemAuditObject  
     OsclMemGlobalAuditObject, 441  
 getHead  
     OsclDoubleListBase, 369  
 GetHostName  
     OsclDNS, 352  
     OsclDNSI, 353  
     OsclDNSIBase, 356  
     OsclGetHostByNameMethod, 413  
 GetHostNameParam, 135  
     addressListCapacity, 135  
     OsclDNSRequestAO, 365  
 GetHostNameParam  
     ~GetHostNameParam, 136  
     canPersistMoreHostAddresses, 136  
     Create, 136  
     Destroy, 136  
     iAddr, 136  
     iAddressList, 136  
     iName, 136  
     PersistHostAddress, 136  
 GetHostNameResponseContainsAliasInfo  
     OsclDNSI, 354  
     OsclDNSIBase, 356  
 GetHostNameSuccess  
     OsclDNSI, 354  
     OsclDNSIBase, 356  
 GetId  
     OsclExecSchedulerCommonBase, 395  
     OsclThread, 576  
 getInstance  
     OsclSingletonRegistry, 534  
     OsclTLSRegistry, 596  
     OsclTLSRegistryEx, 597  
 getLargestContiguousFreeBlockSize  
     OsclMemPoolResizableAllocator, 451  
 GetLastError  
     Oscl\_FileFind, 191  
 getLeaveCode  
     OsclException, 379  
 GetLength  
     BuffFragGroup, 121  
 GetLocalBufsize  
     MediaData, 144  
 GetLocalFragment  
     MediaData, 144  
 GetLock  
     OsclMemAudit, 430  
 getLoggerObject  
     PVLogger, 619  
 GetLogLevel  
     PVLogger, 619

GetMaxFrags  
     BufFragGroup, 122  
 GetMediaFragment  
     MediaData, 144  
 GetMediaSize  
     MediaData, 144  
 getMemFrag  
     OsclRefCounterMemFrag, 497  
 getMemFragPtr  
     OsclRefCounterMemFrag, 497  
 getMemFragSize  
     OsclRefCounterMemFrag, 497  
 getMemPoolBufferAllocatedSize  
     OsclMemPoolResizableAllocator, 451  
 getMemPoolBufferSize  
     OsclMemPoolResizableAllocator, 451  
 GetName  
     OsclExecSchedulerCommonBase, 395  
 GetNext  
     BufFragGroup, 122  
 GetNextHost  
     OsclIDNSI, 354  
     OsclIDNSIBase, 356  
 GetNextHostSuccess  
     OsclIDNSI, 354  
     OsclIDNSIBase, 356  
 GetNumAppenders  
     PVLogger, 619  
 GetNumFrags  
     BufFragGroup, 122  
 GetNumMediaFrags  
     MediaData, 144  
 getOffset  
     OsclDoubleListBase, 369  
 GetParent  
     PVLogger, 620  
 GetPeerName  
     OsclIPSocketI, 419  
     OsclSocketI, 536  
     OsclTCPSocket, 568  
     OsclUDPSocket, 603  
 GetPriority  
     OsclThread, 577  
 GetPVLoggerObject  
     PVLoggerRegistry, 629  
 GetPVLoggerRegistry  
     PVLoggerRegistry, 629  
 GetReadAsyncNumElements  
     OsclNativeFile, 463  
 GetRecvData  
     OsclIPSocketI, 419  
     OsclRecvFromMethod, 486  
     OsclRecvFromRequest, 488  
     OsclRecvMethod, 490  
     OsclRecvRequest, 491  
     OsclTCPSocket, 569  
     OsclTCPSocketI, 573  
     OsclUDPSocket, 603  
     OsclUDPSocketI, 608  
 GetRefCounter  
     OsclSharedPtr, 528  
 getRefCounter  
     OsclRefCounterMemFrag, 497  
 GetRep  
     OsclSharedPtr, 528  
 GetScheduler  
     OsclExecSchedulerCommonBase, 395  
 GetSendData  
     OsclIPSocketI, 419  
     OsclSendMethod, 523  
     OsclSendRequest, 524  
     OsclSendToMethod, 525  
     OsclSendToRequest, 526  
     OsclTCPSocket, 569  
     OsclTCPSocketI, 573  
     OsclUDPSocket, 603  
     OsclUDPSocketI, 608  
 GetShutdown  
     OsclSocketIBase, 542  
 getSize  
     MM\_Audit\_Imp, 154  
 GetSocketError  
     OsclDNSRequestAO, 364  
     OsclSocketRequestAO, 551  
 getTagActualSize  
     MM\_Audit\_Imp, 154  
 GetTimestamp  
     MediaData, 144  
 GetTOS  
     OsclSocketTOS, 564  
 good  
     OsclBinStream, 338  
 GOOD\_STATE  
     OsclBinStream, 337  
 Handle  
     Oscl\_File, 182  
     OsclFileHandle, 405  
 HandleDNSEvent  
     OsclDNSObserver, 361  
 HandleSocketEvent  
     OsclSocketObserver, 548  
 HasAsyncBind  
     OsclSocketIBase, 542  
 HasAsyncListen  
     OsclSocketIBase, 542  
 HasAsyncRead  
     OsclNativeFile, 463

hash  
   OSCL\_String, 262  
   OSCL\_wString, 305  
 HasThisOffset  
   OsclAsyncFileBuffer, 320  
 HaveRoomInCurrentBlock  
   OsclBinStream, 338  
 Head  
   OsclDoubleList, 367  
   OsclPriorityList, 469  
 head  
   Oscl\_Linked\_List\_Base, 215  
 HeapBase, 137  
   HeapBase, 138  
 HeapBase  
   ~HeapBase, 138  
   HeapBase, 138  
 host\_to\_big\_endian  
   osclbase, 36  
 host\_to\_little\_endian  
   osclbase, 36

iActive  
   OsclDNSRequest, 362  
 iAddedNum  
   PVActiveBase, 615  
 iAddr  
   BindParam, 116  
   ConnectParam, 131  
   GetHostNameParam, 136  
   RecvFromParam, 636  
   SendToParam, 640  
 iAddress  
   OsclIPSocketI, 420  
 iAddressList  
   GetHostNameParam, 136  
 iAlloc  
   OsclDNSIBase, 357  
   OsclDNSMethod, 360  
   OsclExecSchedulerCommonBase, 399  
   OsclIPSocketI, 420  
   OsclSocketIBase, 544  
   OsclSocketServIBase, 559  
 iAllocatedSz  
   OsclMemPoolResizableAllocator::Mem-  
     PoolBufferInfo, 455  
 iAOPriority  
   TReadyQueLink, 657  
 iAsyncReadBufferSize  
   OsclNativeFileParams, 465  
 iBlankSocket  
   AcceptParam, 112  
 iBlockBuffer

OsclMemPoolResizableAllocator::Mem-  
   PoolBlockInfo, 454  
 iBlockInfoAlignedSize  
   OsclMemPoolResizableAllocator, 453  
 iBlockingMode  
   OsclExecSchedulerCommonBase, 399  
 iBlockPostFence  
   OsclMemPoolResizableAllocator::Mem-  
     PoolBlockInfo, 454  
 iBlockPreFence  
   OsclMemPoolResizableAllocator::Mem-  
     PoolBlockInfo, 454  
 iBlockSize  
   OsclMemPoolResizableAllocator::Mem-  
     PoolBlockInfo, 454  
 iBuffer  
   OsclBuf, 341  
 iBufferInfoAlignedSize  
   OsclMemPoolResizableAllocator, 453  
 iBufferPostFence  
   OsclMemPoolResizableAllocator::Mem-  
     PoolBufferInfo, 455  
 iBufferPreFence  
   OsclMemPoolResizableAllocator::Mem-  
     PoolBufferInfo, 455  
 iBufferSize  
   OsclMemPoolResizableAllocator::Mem-  
     PoolBufferInfo, 455  
 iBufRecv  
   RecvFromParam, 636  
   RecvParam, 638  
 iBufSend  
   SendParam, 639  
   SendToParam, 640  
 iBusy  
   PVActiveBase, 615  
 iCancel  
   OsclSocketServRequestQElem, 562  
 iCBase  
   OsclTrapStackItem, 600  
 iCheckFreeMemoryAvailable  
   OsclMemPoolResizableAllocator, 453  
 iCheckNextAvailable  
   OsclMemPoolResizableAllocator, 453  
 iCheckNextAvailableFreeChunk  
   OsclMemPoolFixedChunkAllocator, 446  
 iChunkAlignment  
   OsclMemPoolFixedChunkAllocator, 446  
 iChunkSize  
   OsclMemPoolFixedChunkAllocator, 446  
 iChunkSizeMemAligned  
   OsclMemPoolFixedChunkAllocator, 446  
 iComponentId  
   OsclComponentRegistryElement, 346

iComponentIdCounter  
     OsclComponentRegistry, 344

iContainer  
     OsclFileCacheBuffer, 404  
     OsclSocketMethod, 547  
     OsclSocketRequestAO, 553

Id  
     OsclAsyncFileBuffer, 320  
     OsclSocketRequestAO, 552  
     PVThreadContext, 634

iData  
     OsclComponentRegistry, 344

iDebugLogger  
     OsclExecSchedulerCommonBase, 399

iDefAlloc  
     OsclExecSchedulerCommonBase, 399

iDelta  
     OsclExecSchedulerCommonBase, 399

iDNSFxN  
     OsclIDNSMethod, 360

iDNSI  
     OsclIDNSRequestAO, 365

iDNSMethod  
     OsclIDNSRequestAO, 365

iDNSObserver  
     OsclIDNSMethod, 360

iDNSRequest  
     DNSRequestParam, 134

iDNSRequestAO  
     OsclIDNSMethod, 360  
     OsclIDNSRequest, 362

iDNSRequestParam  
     OsclIDNSRequest, 362

iDoStop  
     OsclExecSchedulerCommonBase, 399

iDoSuspend  
     OsclExecSchedulerCommonBase, 399

iEnableNullPtrReturn  
     OsclMemPoolFixedChunkAllocator, 446  
     OsclMemPoolResizableAllocator, 453

iEndAddr  
     OsclMemPoolResizableAllocator::Mem-  
       PoolBufferInfo, 455

iErrAlloc  
     OsclSelect, 520

iErrorTrapImp  
     OsclExecSchedulerCommonBase, 399

iExecTimerQ  
     OsclExecSchedulerCommonBase, 399

iExpectedNumBlocksPerBuffer  
     OsclMemPoolResizableAllocator, 453

iFactory  
     OsclComponentRegistryElement, 346  
     OsclRegistryAccessElement, 508

iFilePosition  
     Oscl\_File::OsclFixedCacheParam, 188

iFlags  
     RecvFromParam, 636  
     RecvParam, 638  
     SendParam, 639  
     SendToParam, 640

iFreeMemChunkList  
     OsclMemPoolFixedChunkAllocator, 446

iFreeMemContextData  
     OsclMemPoolResizableAllocator, 453

iFreeMemPoolObserver  
     OsclMemPoolResizableAllocator, 453

ifront  
     Oscl\_Queue\_Base, 241

iFxn  
     DNSRequestParam, 134  
     SocketRequestParam, 643

iGrandTotalTicks  
     OsclExecSchedulerCommonBase, 399

iHead  
     OsclDoubleListBase, 369  
     OsclDoubleRunner, 370

iHeapCheck  
     OsclSelect, 520

iHigh  
     OsclInteger64Transport, 416

iHow  
     ShutdownParam, 641

iId  
     OsclComponentRegistryElement, 346  
     OsclIDNSMethod, 360  
     OsclIPSocketI, 420

iIsIn  
     TReadyQueLink, 657

iJumpData  
     OsclErrorTrapImp, 378

iLeave  
     OsclErrorTrapImp, 378

iLen  
     PVSockBufRecv, 632  
     PVSockBufSend, 633

iLength  
     OsclBuf, 341

iLogger  
     OsclIDNSMethod, 360  
     OsclIDNSRequestAO, 365  
     OsclExecSchedulerCommonBase, 399  
     OsclIPSocketI, 420  
     OsclSocketServIBase, 559

iLogPerfIndentStr  
     OsclExecSchedulerCommonBase, 399

iLogPerfIndentStrLen  
     OsclExecSchedulerCommonBase, 399

iLogPerfTotal  
     OsclExecSchedulerCommonBase, 399

iLow  
     OsclInteger64Transport, 416

iMaxLen  
     PVSockBufRecv, 632

iMaxLength  
     OsclBuf, 341

iMaxNewMemPoolBufferSz  
     OsclMemPoolResizableAllocator, 453

iMemPool  
     OsclMemPoolFixedChunkAllocator, 446

iMemPoolAligned  
     OsclMemPoolFixedChunkAllocator, 446

iMemPoolAllocator  
     OsclMemPoolFixedChunkAllocator, 446

iMemPoolBufferAllocator  
     OsclMemPoolResizableAllocator, 453

iMemPoolBufferList  
     OsclMemPoolResizableAllocator, 453

iMemPoolBufferNumLimit  
     OsclMemPoolResizableAllocator, 453

iMemPoolBufferSize  
     OsclMemPoolResizableAllocator, 453

iMimeType  
     OsclRegistryAccessElement, 508

iMultiMaxLen  
     RecvFromParam, 636

iMutex  
     OsclComponentRegistry, 344

iName  
     GetHostNameParam, 136  
     OsclExecSchedulerCommonBase, 399  
     PVActiveBase, 615

iNativeAccessMode  
     OsclNativeFileParams, 465

iNativeBufferSize  
     OsclNativeFileParams, 465

iNativeMode  
     OsclExecSchedulerCommonBase, 399

IncLogPerf  
     OsclExecSchedulerCommonBase, 396

increment\_refcnt  
     BufferState, 119

iNext  
     OsclDoubleLink, 366  
     OsclDoubleRunner, 370  
     OsclTrapStackItem, 600

iNextAvailableContextData  
     OsclMemPoolFixedChunkAllocator, 446  
     OsclMemPoolResizableAllocator, 453

iNextFreeBlock  
     OsclMemPoolResizableAllocator::Mem-  
       PoolBlockInfo, 454

OsclMemPoolResizableAllocator::Mem-  
   PoolBufferInfo, 455

Init  
     OsclErrorTrap, 376  
     OsclInit, 415  
     OsclMem, 425  
     OsclScheduler, 516  
     PVLogger, 620

InitExecQ  
     OsclExecSchedulerCommonBase, 396

Insert  
     OsclDoubleListBase, 369  
     OsclPriorityList, 469

insert  
     Oscl\_Map, 220  
     Oscl\_TagTree, 272  
     Oscl\_Vector, 288  
     Oscl\_Vector\_Base, 292

insert\_element  
     Oscl\_Linked\_List, 208  
     Oscl\_Linked\_List\_Base, 213

insert\_unique  
     Oscl\_Rb\_Tree, 244

InsertAfter  
     OsclDoubleLink, 366

InsertBefore  
     OsclDoubleLink, 366

InsertHead  
     OsclDoubleList, 367  
     OsclDoubleListBase, 369

InsertTail  
     OsclDoubleList, 367  
     OsclDoubleListBase, 369

InstallScheduler  
     OsclExecSchedulerCommonBase, 396

INT64  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846

int64  
     osclbase, 35

INT64\_HILO  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846

interfaceAddr  
     OsclIpMReq, 417

INTERNAL\_ERROR  
     BuffFragStatusClass, 123

internalLeave  
     osclerror, 88

internalLeave  
     a, 139

InThread  
     DNSRequestParam, 133

iNumAOAdded

OscIExecSchedulerCommonBase, 399  
iNumChunk  
    OsclMemPoolFixedChunkAllocator, 446  
iNumOfRun  
    OsclAsyncFile, 318  
iNumOfRunErr  
    OsclAsyncFile, 318  
iNumOutstanding  
    OsclMemPoolResizableAllocator::Mem-  
        PoolBufferInfo, 455  
iNumSessions  
    OsclComponentRegistry, 344  
INVALID\_ACCESS\_ERROR  
    OsclProcStatus, 476  
INVALID\_ARGUMENT\_ERROR  
    OsclProcStatus, 476  
INVALID\_FUNCTION\_ERROR  
    OsclProcStatus, 476  
INVALID\_HANDLE\_ERROR  
    OsclProcStatus, 476  
INVALID\_ID  
    BufFragStatusClass, 123  
INVALID\_OPERATION\_ERROR  
    OsclProcStatus, 476  
INVALID\_PARAM\_ERROR  
    OsclProcStatus, 475  
INVALID\_POINTER\_ERROR  
    OsclProcStatus, 476  
INVALID\_PRIORITY\_ERROR  
    OsclProcStatus, 475  
INVALID\_THREAD\_ERROR  
    OsclProcStatus, 475  
INVALID\_THREAD\_ID\_ERROR  
    OsclProcStatus, 475  
INVALID\_TYPE  
    Oscl\_FileFind, 189  
iObserver  
    OsclIPSocketI, 420  
    OsclMemPoolFixedChunkAllocator, 446  
    OsclMemPoolResizableAllocator, 453  
iOffset  
    OsclDoubleListBase, 369  
    OsclDoubleRunner, 370  
iOpCount  
    OsclFileStatsItem, 412  
iOsclBase  
    OsclSelect, 520  
iOsclErrorTrap  
    OsclSelect, 520  
iOsclLogger  
    OsclSelect, 520  
iOsclMemory  
    OsclSelect, 520  
iOsclScheduler  
    OsclSelect, 520  
iOtherExecStats  
    OsclExecSchedulerCommonBase, 399  
iOutputFile  
    OsclSelect, 520  
iPacketLen  
    RecvFromParam, 636  
iPacketSource  
    RecvFromParam, 636  
ipAddr  
    OsclNetworkAddress, 466  
iParam  
    OsclFileStatsItem, 412  
    OsclSocketRequest, 549  
    OsclSocketRequestAO, 553  
iParam2  
    OsclFileStatsItem, 412  
iParamSize  
    OsclSocketRequestAO, 553  
iParentBuffer  
    OsclMemPoolResizableAllocator::Mem-  
        PoolBlockInfo, 454  
iPrev  
    OsclDoubleLink, 366  
iPrevFreeBlock  
    OsclMemPoolResizableAllocator::Mem-  
        PoolBlockInfo, 454  
iPriority  
    OsclPriorityLink, 468  
iPtr  
    PVSockBufRecv, 632  
    PVSockBufSend, 633  
iPVActiveStats  
    PVActiveBase, 615  
iPVReadyQLink  
    PVActiveBase, 615  
iPVStatQ  
    OsclExecSchedulerCommonBase, 399  
iPVStats  
    OsclExecSchedulerCommonBase, 399  
iQSize  
    ListenParam, 141  
iReadyQ  
    OsclExecSchedulerCommonBase, 399  
irear  
    Oscl\_Queue\_Base, 241  
iRefCount  
    DNSRequestParam, 134  
    OsclMemPoolFixedChunkAllocator, 446  
    OsclMemPoolResizableAllocator, 453  
iRequestedAvailableFreeMemSize  
    OsclMemPoolResizableAllocator, 453  
iRequestedNextAvailableSize  
    OsclMemPoolResizableAllocator, 453

iResumeSem  
     OsclExecSchedulerCommonBase, 399

is\_writable  
     OSCL\_String, 262  
     OSCL\_wString, 306

is\_zero  
     TimeValue, 654

is\_zulu  
     TimeValue, 654

IsActive  
     PVLogger, 620

IsAdded  
     PVActiveBase, 613

isAllocNodePtr  
     MM\_AllocBlockHdr, 148

IsBusy  
     OsclActiveObject, 312  
     OsclTimerObject, 588

iSchedulerAlloc  
     OsclSelect, 520

iSchedulerName  
     OsclSelect, 520

iSchedulerReserve  
     OsclSelect, 520

isCIEquivalentTo  
     StrCSumPtrLen, 645  
     StrPtrLen, 648  
     WStrPtrLen, 659

isCIPrefixOf  
     StrPtrLen, 648

iSelect  
     OsclSocketServRequestQElem, 562

IsEmpty  
     OsclDoubleListBase, 369

iSeqNum  
     TReadyQueLink, 657

iServerError  
     OsclSocketServIBase, 559

iServerState  
     OsclSocketServIBase, 559

isFixed  
     OsclFileCacheBuffer, 404

IsHead  
     OsclDoubleList, 367  
     OsclPriorityList, 469

IsIn  
     OsclReadyQ, 485  
     OsclTimerQ, 591

IsInAnyQ  
     PVActiveBase, 614

IsInstalled  
     OsclExecSchedulerCommonBase, 396

IsInUse  
     OsclAsyncFileBuffer, 320

iSize  
     Oscl\_File::OsclFixedCacheParam, 188

isLetter  
     StrPtrLen, 648

IsLocalData  
     MediaData, 144

ISO8601TIME\_BUFFER\_SIZE  
     osclbase, 46

ISO8601timeStrBuf  
     osclbase, 35

ISO8601ToRFC822  
     osclbase, 36

iSocket  
     OsclIPSocketI, 420

iSocketError  
     OsclDNSRequestAO, 365  
     OsclSocketRequestAO, 553

iSocketFxn  
     OsclSocketMethod, 547

iSocketI  
     OsclSocketRequest, 549

iSocketRequest  
     OsclSocketServRequestQElem, 562

iSocketRequestAO  
     OsclSocketMethod, 547  
     OsclSocketRequest, 549

iSocketServ  
     OsclDNSIBase, 357  
     OsclIPSocketI, 420  
     OsclSocketIBase, 544

IsOpen  
     OsclSocketIBase, 542

IsReady  
     OsclDNSIBase, 356

IsSameThreadContext  
     PVThreadContext, 634

IsServConnected  
     OsclSocketServIBase, 559

IsServerThread  
     OsclSocketServI, 557

isSetFailure  
     MM\_Audit\_Imp, 155

IsStarted  
     OsclExecSchedulerCommonBase, 396

IsTail  
     OsclDoubleList, 367  
     OsclPriorityList, 469

iStartAddr  
     OsclMemPoolResizableAllocator::MemPoolBufferInfo, 455

iStartTick  
     OsclFileStatsItem, 412

iStatus  
     PVActiveBase, 615

iStopper  
   OsclExecSchedulerCommonBase, 399

iStopperCrit  
   OsclExecSchedulerCommonBase, 399

IsUpdated  
   OsclFileCacheBuffer, 404

iSuspended  
   OsclExecSchedulerCommonBase, 399

IsValid  
   OsclAsyncFileBuffer, 320

iTAny  
   OsclTrapStackItem, 600

iterator  
   Oscl\_Linked\_List\_Base, 215  
   Oscl\_Map, 218  
   Oscl\_Rb\_Tree, 244  
   Oscl\_Rb\_Tree\_Iterator, 251  
   Oscl\_TagTree::iterator, 277  
   Oscl\_Vector, 286  
   OsclPriorityQueue, 471

iThreadContext  
   OsclExecSchedulerCommonBase, 399  
   PVActiveBase, 615

iTime  
   OsclExecSchedulerCommonBase, 399

iTimeCompareThreshold  
   OsclExecSchedulerCommonBase, 399

iTimeQueuedTicks  
   TReadyQueLink, 657

iTimeToRunTicks  
   TReadyQueLink, 657

iTotalPercent  
   OsclExecSchedulerCommonBase, 399

iTotalTicks  
   OsclFileStatsItem, 412

iTotalTicksTemp  
   OsclExecSchedulerCommonBase, 399

iTrapOperation  
   OsclTrapStackItem, 600

iTrapStack  
   OsclErrorTrapImp, 378

iVec  
   OsclComponentRegistryData, 345

iXferLen  
   SendParam, 639  
   SendToParam, 640

Join  
   OsclIPSocketI, 419  
   OsclSocketI, 537  
   OsclSocketIBase, 542  
   OsclUDPSocket, 603

JoinMulticastGroup  
   OsclUDPSocket, 604

OsclUDPSocketI, 608

Jump  
   OsclJump, 421

key\_comp  
   Oscl\_Map, 221

key\_compare  
   Oscl\_Map, 218

key\_type  
   Oscl\_Map, 218  
   Oscl\_Rb\_Tree, 244

largeasyncfilereadwrite\_test  
   Oscl\_File, 186

Leave  
   OsclError, 372

LeaveIfError  
   OsclError, 372

LeaveIfNull  
   OsclError, 372

Left  
   OsclPtrC, 480

left  
   Oscl\_Rb\_Tree\_Node\_Base, 255

len  
   OsclMemoryFragment, 442  
   StrPtrLen, 648  
   WStrPtrLen, 659

Length  
   OsclAsyncFileBuffer, 320  
   OsclBuf, 341  
   OsclPtr, 477  
   OsclPtrC, 480

length  
   BuffFragGroup, 122  
   OsclBinStream, 339  
   StrPtrLen, 648  
   WStrPtrLen, 659

lineNo  
   MM\_AllocInfo, 150  
   MM\_AllocQueryInfo, 152

link\_type  
   Oscl\_Rb\_Tree, 244  
   Oscl\_Rb\_Tree\_Const\_Iterator, 248  
   Oscl\_Rb\_Tree\_Iterator, 251  
   Oscl\_Rb\_Tree\_Node, 253

LinkedListElement, 140  
   LinkedListElement, 140

LinkedListElement  
   data, 140  
   LinkedListElement, 140  
   next, 140

Listen  
   OsclListenMethod, 422

OsclListenRequest, 423  
 OsclSocketI, 537  
 OsclSocketIBase, 542  
 OsclTCPSocket, 569  
 OsclTCPSocketI, 573  
**ListenAsync**  
 OsclSocketIBase, 542  
 OsclTCPSocket, 569  
 OsclTCPSocketI, 574  
**ListenParam**, 141  
 ListenParam, 141  
**ListenParam**  
 iQSize, 141  
 ListenParam, 141  
**ListenRequest**  
 OsclListenMethod, 422  
**little\_endian\_to\_host**  
 osclbase, 37  
**localbuf**  
 MediaData, 144  
**Lock**  
 OsclLockBase, 424  
 OsclMutex, 460  
 OsclNullLock, 467  
 OsclThreadLock, 579  
**lockAndGetInstance**  
 OsclSingletonRegistry, 534  
**Log**  
 OsclFileStats, 411  
**log\_level\_type**  
 AllPassFilter, 114  
 PVLogger, 618  
 PVLoggerFilter, 624  
 PVLoggerRegistry, 628  
**LogAll**  
 OsclFileStats, 411  
**Logger**  
 OsclSocketI, 537  
**LogMsgBuffers**  
 PVLogger, 620  
**LogMsgBuffersV**  
 PVLogger, 620  
**LogMsgString**  
 PVLogger, 621  
**LogMsgStringV**  
 PVLogger, 621  
**LoopbackSocket**  
 OsclSocketServI, 557  
**lower\_bound**  
 Oscl\_Map, 221  
 Oscl\_Rb\_Tree, 244  
**MakeAddr**  
 OsclSocketI, 537  
 MakeMulticastGroupInformation  
 OsclSocketI, 537  
**makeValidTag**  
 MM\_Audit\_Imp, 155  
**map\_type**  
 Oscl\_TagTree, 270  
**mapit**  
 Oscl\_TagTree::const\_iterator, 274  
 Oscl\_TagTree::iterator, 277  
**mapiter**  
 Oscl\_TagTree::const\_iterator, 274  
 Oscl\_TagTree::iterator, 277  
**Match**  
 OsclComponentRegistryElement, 346  
**MAX\_NUMBER\_OF\_BYTE\_PER\_UTF8**  
 osclutil, 69  
**max\_size**  
 Oscl\_Map, 221  
 Oscl\_Rb\_Tree, 244  
**MAX\_THRDS\_REACHED\_ERROR**  
 OsclProcStatus, 475  
**maximum**  
 Oscl\_Rb\_Tree\_Node\_Base, 255  
**MaxLen**  
 OsclNameString, 461  
**maxsize**  
 CFastRep, 128  
 CHeapRep, 130  
 CStackRep, 132  
**mbchar**  
 osclbase, 35  
**MediaData**, 142  
 MediaData, 143  
**MediaData**  
 ~MediaData, 143  
 AddLocalFragment, 143  
 available\_localbuf, 144  
 Clear, 143  
 GetAvailableBufferSize, 143  
 GetLocalBufsize, 144  
 GetLocalFragment, 144  
 GetMediaFragment, 144  
 GetMediaSize, 144  
 GetNumMediaFrags, 144  
 GetTimestamp, 144  
 IsLocalData, 144  
 localbuf, 144  
 MediaData, 143  
 num\_reserved\_fragments, 144  
 SetTimestamp, 144  
 timestamp, 144  
 MediaStatusClass, 145  
 MediaTimestamp  
 osclutil, 69

**MEM\_ALIGN\_SIZE**  
 osclmemory, 52

**MemAllocator**, 146

**MemAllocator**  
 ~MemAllocator, 146

allocate, 146

deallocate, 146

pointer, 146

**memoryPoolBufferMgmtOverhead**  
 OsclMemPoolResizableAllocator, 451

**message\_id\_type**  
 AllPassFilter, 114

PVLogger, 618

PVLoggerAppender, 623

PVLoggerFilter, 624

PVLoggerLayout, 626

**MethodDone**  
 OsclDNSMethod, 359

OsclSocketMethod, 546

**MICROSECONDS**  
 osclbase, 35

**MILLISECONDS**  
 osclbase, 35

**MIN\_FENCE\_SIZE**  
 osclmemory, 52

**minimum**  
 Oscl\_Rb\_Tree\_Node\_Base, 255

**MM\_AddTag**  
 MM\_Audit\_Imp, 155

OsclMemAudit, 430

**MM\_ALLOC\_MAX\_QUERY\_FILENAME\_-LEN**  
 osclmemory, 52

**MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN**  
 osclmemory, 52

**MM\_allocate**  
 MM\_Audit\_Imp, 155

OsclMemAudit, 430

**MM\_AllocBlockFence**, 147

MM\_AllocBlockFence, 147

**MM\_AllocBlockFence**  
 check\_fence, 147

fill\_fence, 147

MM\_AllocBlockFence, 147

pad, 147

**MM\_AllocBlockHdr**, 148

MM\_AllocBlockHdr, 148

**MM\_AllocBlockHdr**  
 isAllocNodePtr, 148

MM\_AllocBlockHdr, 148

pad, 148

pNode, 148

pRootNode, 148

setAllocNodeFlag, 148

size, 148

**MM\_AllocInfo**, 149

MM\_AllocInfo, 150

**MM\_AllocInfo**  
 ~MM\_AllocInfo, 150

allocNum, 150

bSetFailure, 150

lineNo, 150

MM\_AllocInfo, 150

operator delete, 150

operator new, 150

pFileName, 150

pMemBlock, 150

pStatsNode, 150

size, 150

**MM\_AllocNode**, 151

MM\_AllocNode, 151

**MM\_AllocNode**  
 ~MM\_AllocNode, 151

MM\_AllocNode, 151

operator delete, 151

operator new, 151

pAllocInfo, 151

pNext, 151

pPrev, 151

**MM\_AllocNodeAutoPtr**  
 osclmemory, 59

**MM\_AllocQueryInfo**, 152

**MM\_AllocQueryInfo**  
 allocNum, 152

fileName, 152

lineNo, 152

pMemBlock, 152

size, 152

tag, 152

**MM\_AUDIT\_ALLOC\_NODE\_ENABLE\_-FLAG**  
 osclmemory, 52

**MM\_AUDIT\_ALLOC\_NODE\_SUPPORT**  
 osclmemory, 52

**MM\_AUDIT\_FAILURE\_SIMULATION\_-SUPPORT**  
 osclmemory, 52

**MM\_AUDIT\_FENCE\_SUPPORT**  
 osclmemory, 52

**MM\_AUDIT\_FILL\_SUPPORT**  
 osclmemory, 52

**MM\_Audit\_Imp**, 153

~MM\_Audit\_Imp, 154

addAllocNode, 154

createStatsNode, 154

getAuditRoot, 154

getSize, 154

getTagActualSize, 154

isSetFailure, 155  
 makeValidTag, 155  
**MM\_AddTag**, 155  
**MM\_allocate**, 155  
**MM\_Audit\_Imp**, 154  
**MM\_CreateAllocNodeInfo**, 155  
**MM\_deallocate**, 155  
**MM\_GetAllocNo**, 155  
**MM\_GetAllocNodeInfo**, 156  
**MM\_GetExistingTag**, 156  
**MM\_GetMode**, 156  
**MM\_GetNumAllocNodes**, 156  
**MM\_GetOverheadStats**, 156  
**MM\_GetPostfillPattern**, 156  
**MM\_GetPrefillPattern**, 156  
**MM\_GetRootNode**, 157  
**MM\_GetStats**, 157  
**MM\_GetStatsInDepth**, 157  
**MM\_GetTagName**, 157  
**MM\_GetTreeNodes**, 157  
**MM\_ReleaseAllocNodeInfo**, 157  
**MM\_SetFailurePoint**, 158  
**MM\_SetMode**, 158  
**MM\_SetPostfillPattern**, 158  
**MM\_SetPrefillPattern**, 158  
**MM\_SetTagLevel**, 158  
**MM\_UnsetFailurePoint**, 158  
**MM\_Validate**, 158  
**pruneSubtree**, 159  
**removeALLAllocNodes**, 159  
**removeAllocNode**, 159  
**retrieveParentTag**, 159  
**retrieveParentTagLength**, 159  
**updateStatsNode**, 159  
**updateStatsNodeInFailure**, 159  
**validate**, 159  
**validate\_all\_heap**, 159  
**MM\_AUDIT\_INCLUDE\_ALL\_HEAP\_-VALIDATION**  
     osclmemory, 52  
**MM\_AUDIT\_POSTFILL\_FLAG**  
     osclmemory, 52  
**MM\_AUDIT\_PREFILL\_FLAG**  
     osclmemory, 52  
**MM\_AUDIT\_SUPPRESS\_FILENAME\_FLAG**  
     osclmemory, 52  
**MM\_AUDIT\_VALIDATE\_ALL\_HEAP\_FLAG**  
     osclmemory, 52  
**MM\_AUDIT\_VALIDATE\_BLOCK**  
     osclmemory, 52  
**MM\_AUDIT\_VALIDATE\_ON\_FREE\_FLAG**  
     osclmemory, 52  
**MM\_AuditOverheadStats**, 161  
**MM\_AuditOverheadStats**  
     per\_allocation\_overhead, 161  
     stats\_overhead, 161  
**MM\_CreateAllocNodeInfo**  
     **MM\_Audit\_Imp**, 155  
     OsclMemAudit, 430  
**MM\_deallocate**  
     **MM\_Audit\_Imp**, 155  
     OsclMemAudit, 430  
**MM\_FailInsertParam**, 162  
     **MM\_FailInsertParam**, 162  
**MM\_FailInsertParam**  
     **MM\_FailInsertParam**, 162  
     nAllocNum, 162  
     operator delete, 162  
     operator new, 162  
     reset, 162  
     xsubi, 162  
**MM\_GetAllocNo**  
     **MM\_Audit\_Imp**, 155  
     OsclMemAudit, 430  
**MM\_GetAllocNodeInfo**  
     **MM\_Audit\_Imp**, 156  
     OsclMemAudit, 430  
**MM\_GetExistingTag**  
     **MM\_Audit\_Imp**, 156  
     OsclMemAudit, 431  
**MM\_GetMode**  
     **MM\_Audit\_Imp**, 156  
     OsclMemAudit, 431  
**MM\_GetNumAllocNodes**  
     **MM\_Audit\_Imp**, 156  
     OsclMemAudit, 431  
**MM\_GetOverheadStats**  
     **MM\_Audit\_Imp**, 156  
     OsclMemAudit, 431  
**MM\_GetPostfillPattern**  
     **MM\_Audit\_Imp**, 156  
     OsclMemAudit, 431  
**MM\_GetPrefillPattern**  
     **MM\_Audit\_Imp**, 156  
     OsclMemAudit, 431  
**MM\_GetRefCount**  
     OsclMemAudit, 431  
**MM\_GetRootNode**  
     **MM\_Audit\_Imp**, 157  
     OsclMemAudit, 432  
**MM\_GetStats**  
     **MM\_Audit\_Imp**, 157  
     OsclMemAudit, 432  
**MM\_GetStatsInDepth**  
     **MM\_Audit\_Imp**, 157  
     OsclMemAudit, 432  
**MM\_GetTagName**  
     **MM\_Audit\_Imp**, 157

OsclMemAudit, 432  
**MM\_GetTreeNodes**  
 MM\_Audit\_Imp, 157  
 OsclMemAudit, 432  
**MM\_ReleaseAllocNodeInfo**  
 MM\_Audit\_Imp, 157  
 OsclMemAudit, 432  
**MM\_SetFailurePoint**  
 MM\_Audit\_Imp, 158  
 OsclMemAudit, 432  
**MM\_SetMode**  
 MM\_Audit\_Imp, 158  
 OsclMemAudit, 433  
**MM\_SetPostfillPattern**  
 MM\_Audit\_Imp, 158  
 OsclMemAudit, 433  
**MM\_SetPrefillPattern**  
 MM\_Audit\_Imp, 158  
 OsclMemAudit, 433  
**MM\_SetTagLevel**  
 MM\_Audit\_Imp, 158  
 OsclMemAudit, 433  
**MM\_Stats\_CB**, 163  
 MM\_Stats\_CB, 163  
 num\_child\_nodes, 163  
 operator delete, 163  
 operator new, 163  
 pStats, 163  
 tag, 163  
**MM\_Stats\_t**, 164  
 MM\_Stats\_t, 165  
 numAllocFails, 165  
 numAllocs, 165  
 numBytes, 165  
 operator delete, 165  
 operator new, 165  
 peakNumAllocs, 165  
 peakNumBytes, 165  
 reset, 165  
 totalNumAllocs, 165  
 totalNumBytes, 165  
 update, 165  
**MM\_StatsNodeTagTreeType**  
 osclmemory, 59  
**MM\_UnsetFailurePoint**  
 MM\_Audit\_Imp, 158  
 OsclMemAudit, 433  
**MM\_Validate**  
 MM\_Audit\_Imp, 158  
 OsclMemAudit, 433  
**MMAuditCharAutoPtr**  
 osclmemory, 59  
**MMAuditUint8AutoPtr**  
 osclmemory, 59

Mode  
 OsclNativeFile, 463  
 mode  
 oscl\_stat\_buf, 259  
**MODE\_APPEND**  
 Oscl\_File, 180  
**MODE\_BINARY**  
 Oscl\_File, 180  
**MODE\_READ**  
 Oscl\_File, 180  
**MODE\_READ\_PLUS**  
 Oscl\_File, 180  
**MODE\_READWRITE**  
 Oscl\_File, 180  
**MODE\_TEXT**  
 Oscl\_File, 180  
**mode\_type**  
 Oscl\_File, 180  
**move\_to\_end**  
 Oscl\_Linked\_List, 209  
 Oscl\_Linked\_List\_Base, 213  
 Oscl\_MTLinkedList, 226  
**move\_to\_front**  
 Oscl\_Linked\_List, 209  
 Oscl\_Linked\_List\_Base, 214  
 Oscl\_MTLinkedList, 227  
**MSEC\_PER\_SEC**  
 oscibase, 46  
**MSEC\_TO\_MICROSEC**  
 oscl\_socket\_method.h, 759  
**MsecToTicks**  
 OsclTickCount, 580  
**multicastAddr**  
 OsclIpMReq, 417  
**MUTEX\_LOCKED\_ERROR**  
 OsclProcStatus, 476

**nAllocNum**  
 MM\_FailInsertParam, 162

**New**  
 Oscl\_DefAllocWithRefCounter, 174

**NewL**  
 OsclAcceptMethod, 308  
 OsclAsyncFile, 317  
 OsclAsyncFileBuffer, 320  
 OsclBindMethod, 322  
 OsclBuf, 341  
 OsclConnectMethod, 348  
 OsclDNS, 352  
 OsclDNSI, 354  
 OsclGetHostNameMethod, 413  
 OsclListenMethod, 422  
 OsclRecvFromMethod, 486  
 OsclRecvMethod, 490

OsclSendMethod, 523  
 OsclSendToMethod, 525  
 OsclShutdownMethod, 530  
 OsclSocketI, 537  
 OsclSocketServ, 555  
 OsclSocketServI, 557  
 OsclTCPSocket, 569  
 OsclTCPSocketI, 574  
 OsclUDPSocket, 604  
 OsclUDPSocketI, 608  
**NewRequest**  
   OsclIDNSRequestAO, 364  
   OsclSocketRequestAO, 552  
**next**  
   BufFragGroup, 122  
   LinkedListElement, 140  
**nextFragPtr**  
   OsclBinStream, 339  
**NO\_PERMISSION\_ERROR**  
   OsclProcStatus, 475  
**Node**  
   Oscl\_TagTree::Node, 280  
**node**  
   Oscl\_Rb\_Tree\_Const\_Iterator, 248  
   Oscl\_Rb\_Tree\_Iterator, 251  
**node\_ptr**  
   Oscl\_TagTree, 270  
**node\_type**  
   Oscl\_TagTree, 270  
**NOT\_ENOUGH\_MEMORY\_ERROR**  
   OsclProcStatus, 475  
**NOT\_ENOUGH\_RESOURCES\_ERROR**  
   OsclProcStatus, 475  
**NOT\_ENOUGH\_SPACE**  
   BufFragStatusClass, 123  
**NOT\_IMPLEMENTED**  
   OsclProcStatus, 476  
**NOT\_SUSPENDED\_ERROR**  
   OsclProcStatus, 475  
**notifyfreeblockavailable**  
   OsclMemPoolResizableAllocator, 451  
**notifyfreechunkavailable**  
   OsclMemPoolFixedChunkAllocator, 445  
**notifyfreememoryavailable**  
   OsclMemPoolResizableAllocator, 451  
**NTPTime**, 166  
   get\_lower32, 168  
   get\_middle32, 168  
   get\_upper32, 168  
   get\_value, 168  
   NTPTime, 167, 168  
   operator+=, 168  
   operator-, 168  
   operator=, 168, 169  
**operator \***  
   Oscl\_Rb\_Tree\_Const\_Iterator, 248  
   Oscl\_Rb\_Tree\_Iterator, 251  
   Oscl\_TagTree::const\_iterator, 274  
   Oscl\_TagTree::iterator, 277  
   OsclExclusiveArrayPtr, 381  
   OsclExclusivePtr, 384  
   OsclExclusivePtrA, 387  
**set\_from\_system\_time**, 169  
**set\_to\_current\_time**, 169  
**TimeValue**, 655  
**to\_system\_time**, 169  
**NULL**  
   osclbase, 32  
**NULL\_INPUT**  
   BuffFragStatusClass, 123  
**NULL\_TERM\_CHAR**  
   osclbase, 32  
**num\_child\_nodes**  
   MM\_Stats\_CB, 163  
**num\_elements**  
   Oscl\_Linked\_List\_Base, 215  
**num\_fragments**  
   BufFragGroup, 122  
**num\_reserved\_fragments**  
   MediaData, 144  
**numAllocFails**  
   MM\_Stats\_t, 165  
**numAllocs**  
   MM\_Stats\_t, 165  
**numBytes**  
   MM\_Stats\_t, 165  
**numelems**  
   Oscl\_Queue\_Base, 241  
   Oscl\_Vector\_Base, 293  
**numFrags**  
   OsclBinStream, 339  
**octet**  
   osclbase, 35  
**Offset**  
   OsclAsyncFileBuffer, 320  
**Open**  
   Oscl\_File, 182  
   OsclAsyncFile, 317, 318  
   OsclDNSI, 354  
   OsclDNSIBase, 356  
   OsclFileCache, 402  
   OsclNativeFile, 463  
   OsclSocketI, 537  
   OsclSocketIBase, 543  
   OsclSocketServRequestList, 560  
**OpenSession**  
   OsclComponentRegistry, 344  
**operator \***  
   Oscl\_Rb\_Tree\_Const\_Iterator, 248  
   Oscl\_Rb\_Tree\_Iterator, 251  
   Oscl\_TagTree::const\_iterator, 274  
   Oscl\_TagTree::iterator, 277  
   OsclExclusiveArrayPtr, 381  
   OsclExclusivePtr, 384  
   OsclExclusivePtrA, 387

OSCLMemAutoPtr, 437  
 OsclSharedPtr, 528  
 OsclSingleton, 532  
 OsclTLS, 592  
 OsclTLSEEx, 594  
 operator \*=  
     TimeValue, 654  
 operator delete  
     MM\_AllocInfo, 150  
     MM\_AllocNode, 151  
     MM\_FailInsertParam, 162  
     MM\_Stats\_CB, 163  
     MM\_Stats\_t, 165  
     oscl\_mem.h, 715  
     OsclErrorAllocator, 375  
     osclmemory, 60  
     OsclMemStatsNode, 458  
 operator delete[]  
     osclmemory, 60  
 operator new  
     MM\_AllocInfo, 150  
     MM\_AllocNode, 151  
     MM\_FailInsertParam, 162  
     MM\_Stats\_CB, 163  
     MM\_Stats\_t, 165  
     oscl\_mem.h, 715  
     osclconfig\_global\_placement\_new.h, 810  
     OsclErrorAllocator, 375  
     osclmemory, 60  
     OsclMemStatsNode, 458  
 operator new[]  
     osclmemory, 60  
 operator T \*  
     OsclDoubleRunner, 370  
 operator TheClass \*  
     OsclSharedPtr, 529  
 operator!=  
     Oscl\_Rb\_Tree\_Const\_Iterator, 248  
     Oscl\_Rb\_Tree\_Iterator, 251  
     OSCL\_String, 262  
     Oscl\_TagTree::const\_iterator, 274  
     Oscl\_TagTree::iterator, 277  
     OSCL\_wString, 306  
     OsclAOStatus, 315  
     OsclUuid, 611  
     StrCSumPtrLen, 645  
     StrPtrLen, 648  
     TimeValue, 655  
     WStrPtrLen, 659  
 operator()  
     Oscl\_Less, 205  
     Oscl\_Map::value\_compare, 223  
     Oscl\_Select1st, 256  
     Oscl\_Tag\_Base, 268  
 operator+  
     osclbase, 37  
 operator++  
     Oscl\_Rb\_Tree\_Const\_Iterator, 248  
     Oscl\_Rb\_Tree\_Iterator, 251  
     Oscl\_TagTree::const\_iterator, 274  
     Oscl\_TagTree::iterator, 277  
     OsclDoubleRunner, 370  
 operator+=  
     NTPTime, 168  
     OSCL\_String, 262  
     OSCL\_wString, 306  
     TimeValue, 654  
 operator-  
     NTPTime, 168  
     osclbase, 37  
 operator-  
     Oscl\_Rb\_Tree\_Const\_Iterator, 248  
     Oscl\_Rb\_Tree\_Iterator, 251  
     Oscl\_TagTree::const\_iterator, 274  
     Oscl\_TagTree::iterator, 277  
     OsclDoubleRunner, 370  
 operator-=  
     TimeValue, 654  
 operator->  
     Oscl\_Rb\_Tree\_Const\_Iterator, 248  
     Oscl\_Rb\_Tree\_Iterator, 251  
     Oscl\_TagTree::const\_iterator, 274  
     Oscl\_TagTree::iterator, 277  
     OsclExclusiveArrayPtr, 381  
     OsclExclusivePtr, 384  
     OsclExclusivePtrA, 387  
     OSCLMemAutoPtr, 437  
     OsclSharedPtr, 529  
     OsclSingleton, 532  
     OsclTLS, 592  
     OsclTLSEEx, 594  
 operator<  
     OSCL\_String, 262  
     Oscl\_Tag, 265  
     OSCL\_wString, 306  
     OsclAOStatus, 315  
     TimeValue, 655  
 operator<<  
     OsclBinOStreamBigEndian, 333  
     OsclBinOStreamLittleEndian, 335  
 operator<=

    OSCL\_String, 263  
     OSCL\_wString, 306  
     OsclAOStatus, 315  
     TimeValue, 655  
 operator=

    NTPTime, 168, 169  
     OSCL\_FastString, 177

OSCL\_HeapStringA, 201  
 Oscl\_Map, 221  
 Oscl\_Rb\_Tree, 244  
 OSCL\_String, 263  
 Oscl\_TagTree, 272  
 Oscl\_Vector, 288  
 OSCL\_wFastString, 295  
 OSCL\_wHeapStringA, 301  
 OSCL\_wString, 306  
 OsclAOStatus, 315  
 OsclComponentRegistryElement, 346  
 OsclExclusiveArrayPtr, 381  
 OsclExclusivePtr, 384  
 OsclExclusivePtrA, 387  
 OSCLMemAutoPtr, 437  
 OsclRefCounterMemFrag, 497  
 OsclSharedPtr, 529  
 osclutil, 72–74  
 OsclUuid, 611  
 StrCSumPtrLen, 645  
 StrPtrLen, 648  
 TimeValue, 654  
 WStrPtrLen, 659  
**operator==**  
   Oscl\_Rb\_Tree\_Const\_Iterator, 248  
   Oscl\_Rb\_Tree\_Iterator, 251  
   OSCL\_String, 263  
   Oscl\_TagTree::const\_iterator, 274  
   Oscl\_TagTree::iterator, 277  
   OSCL\_wString, 306  
   OsclAOStatus, 315  
   osclbase, 37  
   OsclNetworkAddress, 466  
   OsclUuid, 611  
   StrCSumPtrLen, 645  
   StrPtrLen, 648  
   TimeValue, 655  
   WStrPtrLen, 659  
**operator>**  
   OSCL\_String, 263  
   OSCL\_wString, 306  
   OsclAOStatus, 315  
   TimeValue, 655  
**operator>=**  
   OSCL\_String, 263  
   OSCL\_wString, 306  
   OsclAOStatus, 315  
   TimeValue, 655  
**operator>>**  
   OsclBinIStreamBigEndian, 327  
   OsclBinIStreamLittleEndian, 330  
**operator[]**  
   Oscl\_Map, 221  
   OSCL\_String, 263  
   Oscl\_TagTree, 272  
   Oscl\_Vector, 288  
   OSCL\_wString, 306  
**optype**  
   OSCL\_FastString, 176  
   OSCL\_HeapString, 197  
   OSCL\_HeapStringA, 199  
   OSCL\_StackString, 258  
   OSCL\_wFastString, 294  
   OSCL\_wHeapString, 298  
   OSCL\_wHeapStringA, 300  
   OSCL\_wStackString, 303  
**OSCL Base**, 25  
**OSCL config**, 21  
**OSCL Error**, 85  
**OSCL Init**, 107  
**OSCL IO**, 95  
**OSCL Memory**, 47  
**OSCL Proc**, 103  
**OSCL Util**, 63  
**OSCL\_ABS**  
   osclbase, 32  
**oscl\_abs**  
   osclutil, 74  
**OSCL\_AF\_INET**  
   osclconfig\_io.h, 815  
**Oscl\_Alloc**, 170  
   ~Oscl\_Alloc, 170  
   allocate, 170  
   allocate\_fl, 170  
**OSCL\_ALLOC\_DELETE**  
   osclmemory, 52  
**OSCL\_ALLOC\_NEW**  
   osclmemory, 53  
**oscl\_aostatus.h**, 660  
**OSCL\_ARRAY\_DELETE**  
   osclmemory, 53  
**OSCL\_ARRAY\_NEW**  
   osclmemory, 53  
**OSCL\_ASCII\_CASE\_MAGIC\_BIT**  
   osclutil, 84  
**oscl\_asin**  
   osclutil, 74  
**OSCL\_ASSERT**  
   osclbase, 32  
**OSCL\_Assert**  
   osclbase, 37  
**oscl\_assert.h**, 661  
**OSCL\_ASSERT\_ALWAYS**  
   osclconfig, 22  
**oscl\_atan**  
   osclutil, 74  
**OSCL\_AUDIT\_ARRAY\_NEW**  
   osclmemory, 53

**OSCL\_AUDIT\_CALLOC**  
 osclmemory, 54  
**OSCL\_AUDIT\_MALLOC**  
 osclmemory, 54  
**OSCL\_AUDIT\_NEW**  
 osclmemory, 54  
**OSCL\_AUDIT\_REALLOC**  
 osclmemory, 55  
**OSCL\_BAD\_ALLOC\_EXCEPTION\_CODE**  
 oscrror, 88  
**oscl\_base.h**, 662  
**oscl\_base\_alloc.h**, 663  
**oscl\_base\_macros.h**, 664  
**oscl\_bin\_stream.h**, 665  
**OSCL\_BYPASS\_MEMMGT**  
 osclconfig\_memory.h, 827  
**oscl\_byte\_order.h**, 666  
**OSCL\_BYTE\_ORDER\_BIG\_ENDIAN**  
 osclconfig, 22  
**OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN**  
 osclconfig, 22  
**OSCL\_CALLOC**  
 osclmemory, 55  
**oscl\_calloc**  
 osclmemory, 55  
**OSCL\_CATCH**  
 oscrror, 88  
**OSCL\_CATCH\_ANY**  
 oscrror, 88  
**OSCL\_CHAR\_IS\_SIGNED**  
 osclconfig\_limits\_typedefs.h, 826  
**OSCL\_CHAR\_IS\_UNSIGNED**  
 osclconfig\_limits\_typedefs.h, 826  
**oscl\_chdir**  
 osclio, 99  
**oscl\_CIstrcmp**  
 osclbase, 37, 38  
**oscl\_CIstrcmp**  
 osclbase, 38  
**OSCL\_CLEANUP\_BASE\_CLASS**  
 osclmemory, 55  
**OSCL\_CLOCK\_HAS\_DRIFT\_CORRECTION**  
 osclconfig\_util.h, 847  
**OSCL\_COND\_EXPORT\_REF**  
 osclbase, 32  
**OSCL\_COND\_IMPORT\_REF**  
 osclbase, 32  
**OSCL\_CONST\_CAST**  
 osclbase, 32  
**oscl\_cos**  
 osclutil, 74  
**Oscl\_Dealloc**, 171  
 ~Oscl\_Dealloc, 171  
 deallocate, 171  
**Oscl\_DefAlloc**, 172  
**Oscl\_DefAlloc**  
 allocate, 172  
 allocate\_fl, 172  
 deallocate, 172  
**oscl\_defalloc.h**, 667  
**Oscl\_DefAllocWithRefCounter**, 173  
**Oscl\_DefAllocWithRefCounter**  
 addRef, 173  
 Delete, 173  
 getCount, 173  
 New, 174  
 removeRef, 174  
**OSCL\_DEFAULT\_FREE**  
 osclmemory, 56  
**OSCL\_DEFAULT\_MALLOC**  
 osclmemory, 56  
**OSCL\_DELETE**  
 osclmemory, 56  
**Oscl\_DeleteFile**  
 Oscl\_FileServer, 193, 194  
**OSCL\_DISABLE\_INLINES**  
 osclconfig\_unix\_android.h, 842  
 osclconfig\_unix\_common.h, 846  
**OSCL\_DISABLE\_WARNING\_RETURN\_-TYPE\_NOT\_UDT**  
 osclbase, 32  
 osclmemory, 56  
**OSCL\_DISABLE\_WARNING\_TRUNCATE\_-DEBUG\_MESSAGE**  
 oscl\_map.h, 709  
 oscl\_mem.h, 715  
 oscl\_mem\_audit.h, 717  
 oscl\_mem\_audit\_internals.h, 718  
 oscl\_mem\_auto\_ptr.h, 719  
 oscl\_tagtree.h, 786  
 oscl\_tree.h, 795  
 osclbase, 33  
 osclmemory, 56  
**oscl\_dll.h**, 668  
**OSCL\_DLL\_ENTRY\_POINT**  
 osclbase, 33  
**OSCL\_DLL\_ENTRY\_POINT\_DEFAULT**  
 osclbase, 33  
**oscl\_dns.h**, 669  
**oscl\_dns\_gethostbyname.h**, 670  
**oscl\_dns\_imp.h**, 671  
**oscl\_dns\_imp\_base.h**, 672  
**oscl\_dns\_imp\_pv.h**, 673  
**oscl\_dns\_method.h**, 674  
**oscl\_dns\_param.h**, 675  
 TDNSRequestParamAllocator, 675  
**oscl\_dns\_request.h**, 676  
**oscl\_dns\_tuneables.h**, 677

---

PV\_DNS\_IS\_THREAD, 677  
 PV\_DNS\_SERVER, 677  
 oscl\_double\_list.h, 678  
 OSCL\_DYNAMIC\_CAST  
     osclbase, 33  
 OSCL\_ERR\_NONE  
     osclerror, 89  
 oscl\_errno.h, 679  
 oscl\_error.h, 680  
 oscl\_error\_allocator.h, 681  
 oscl\_error\_codes.h, 682  
 oscl\_error\_imp.h, 683  
 oscl\_error\_imp\_cppexceptions.h, 684  
 oscl\_error\_imp\_fatalerror.h, 685  
     \_PV\_TRAP, 685  
     \_PV\_TRAP\_NO\_TLS, 685  
     PVError\_DoLeave, 685  
 oscl\_error\_imp\_jumps.h, 686  
     \_PV\_TRAP, 686  
     \_PV\_TRAP\_NO\_TLS, 686  
     PVError\_DoLeave, 687  
 oscl\_error\_trapcleanup.h, 688  
 oscl\_exception.h, 689  
 OSCL\_EXCEPTSET\_FLAG  
     oscl\_socket\_serv\_imp\_pv.h, 767  
 oscl\_exclusive\_ptr.h, 690  
 oscl\_exp  
     osclutil, 74  
 OSCL\_EXPORT\_REF  
     osclconfig.h, 803  
 OSCL\_FastString, 175  
     OSCL\_FastString, 176  
 OSCL\_FastString  
     ~OSCL\_FastString, 176  
     chartype, 176  
     get\_cstr, 177  
     get\_maxsize, 177  
     get\_size, 177  
     get\_str, 177  
     operator=, 177  
     otype, 176  
     OSCL\_FastString, 176  
 OSCL\_String, 178  
 other\_chartype, 176  
 set, 177, 178  
 set\_length, 178  
 Oscl\_File  
     ESymbianAccessMode\_Rfile, 180  
     ESymbianAccessMode\_RfileBuf, 180  
     MODE\_APPEND, 180  
     MODE\_BINARY, 180  
     MODE\_READ, 180  
     MODE\_READ\_PLUS, 180  
     MODE\_READWRITE, 180  
         MODE\_TEXT, 180  
         SEEKCUR, 180  
         SEEKEND, 180  
         SEEKSET, 180  
     Oscl\_File, 179  
         ~Oscl\_File, 181  
         AddFixedCache, 181  
         asyncfilereadcancel\_test, 186  
         asyncfilereadwrite\_test, 186  
         Close, 181  
         EndOfFile, 181  
         Flush, 182  
         GetError, 182  
         Handle, 182  
         largeasyncfilereadwrite\_test, 186  
         mode\_type, 180  
         Open, 182  
         Oscl\_File, 181  
         Oscl\_FileServer, 194  
         OsclFileCache, 186  
         OsclFileCacheBuffer, 186  
         OsclFileHandle, 405  
         Read, 183  
         RemoveFixedCache, 183  
         Seek, 183  
         seek\_type, 180  
         SetAsyncReadBufferSize, 183  
         SetCacheObserver, 184  
         SetFileHandle, 184  
         SetLoggingEnable, 184  
         SetNativeAccessMode, 184  
         SetNativeBufferSize, 185  
         SetPVCacheSize, 185  
         SetSize, 185  
         SetSummaryStatsLoggingEnable, 185  
         Size, 185  
         Tell, 185  
         TSymbianAccessMode, 180  
         Write, 186  
     Oscl\_File::OsclCacheObserver, 187  
     Oscl\_File::OsclCacheObserver  
         ~OsclCacheObserver, 187  
         ChooseCurCache, 187  
     Oscl\_File::OsclFixedCacheParam, 188  
     Oscl\_File::OsclFixedCacheParam  
         Contains, 188  
         iFilePath, 188  
         iSize, 188  
     oscl\_file\_async\_read.h, 691  
 OSCL\_FILE\_ATTRIBUTE\_ARCHIVE  
     OsclFileManager, 406  
 OSCL\_FILE\_ATTRIBUTE\_DIRECTORY  
     OsclFileManager, 406  
 OSCL\_FILE\_ATTRIBUTE\_HIDDEN

OsclFileManager, 406  
**OSCL\_FILE\_ATTRIBUTE\_NORMAL**  
 OsclFileManager, 406  
**OSCL\_FILE\_ATTRIBUTE\_READONLY**  
 OsclFileManager, 406  
**OSCL\_FILE\_ATTRIBUTE\_SYSTEM**  
 OsclFileManager, 406  
**OSCL\_FILE\_ATTRIBUTE\_TYPE**  
 OsclFileManager, 406  
**OSCL\_FILE\_BUFFER\_MAX\_SIZE**  
 osclconfig\_io.h, 815  
**oscl\_file\_cache.h**, 692  
**OSCL\_FILE\_CHAR\_PATH\_DELIMITER**  
 oscilio, 97  
**oscl\_file\_dir\_utils.h**, 693  
**oscl\_file\_find.h**, 695  
**oscl\_file\_handle.h**, 696  
**oscl\_file\_io.h**, 697  
**oscl\_file\_manager.h**, 698  
**oscl\_file\_native.h**, 699  
**oscl\_file\_server.h**, 700  
**oscl\_file\_stats.h**, 701  
**OSCL\_FILE\_STATS\_LOGGER\_NODE**  
 oscilio, 97  
**oscl\_file\_types.h**, 702  
**OSCL\_FILE\_WCHAR\_PATH\_DELIMITER**  
 oscilio, 97  
**Oscl\_FileFind**  
 DIR\_TYPE, 189  
 E\_BUFFER\_TOO\_SMALL, 190  
 E\_INVALID\_ARG, 189  
 E\_INVALID\_STATE, 189  
 E\_MEMORY\_ERROR, 190  
 E\_NO\_MATCH, 190  
 E\_NOT\_IMPLEMENTED, 190  
 E\_OK, 189  
 E\_OTHER, 190  
 E\_PATH\_NOT\_FOUND, 189  
 E\_PATH\_TOO\_LONG, 189  
 FILE\_TYPE, 189  
 INVALID\_TYPE, 189  
**Oscl\_FileFind**, 189  
 Oscl\_FileFind, 190  
**Oscl\_FileFind**  
 ~Oscl\_FileFind, 190  
 Close, 190  
 element\_type, 189  
 error\_type, 189  
 FindFirst, 190  
 FindNext, 191  
 GetElementType, 191  
 GetLastError, 191  
 Oscl\_FileFind, 190  
**OSCL\_FILEMGMT\_E\_ALREADY\_EXISTS**  
 oscilio, 98  
**OSCL\_FILEMGMT\_E\_NO\_MATCH**  
 oscilio, 98  
**OSCL\_FILEMGMT\_E\_NOT\_EMPTY**  
 oscilio, 98  
**OSCL\_FILEMGMT\_E\_NOT\_IMPLEMENTED**  
 oscilio, 98  
**OSCL\_FILEMGMT\_E\_OK**  
 oscilio, 98  
**OSCL\_FILEMGMT\_E\_PATH\_NOT\_FOUND**  
 oscilio, 98  
**OSCL\_FILEMGMT\_E\_PATH\_TOO\_LONG**  
 oscilio, 98  
**OSCL\_FILEMGMT\_E\_PERMISSION\_DENIED**  
 oscilio, 98  
**OSCL\_FILEMGMT\_E\_SYS\_SPECIFIC**  
 oscilio, 98  
**OSCL\_FILEMGMT\_E\_UNKNOWN**  
 oscilio, 98  
**OSCL\_FILEMGMT\_ERR\_TYPE**  
 oscilio, 98  
**OSCL\_FILEMGMT\_MODE\_DIR**  
 oscilio, 98  
**OSCL\_FILEMGMT\_MODES**  
 oscilio, 98  
**OSCL\_FILEMGMT\_PERMS**  
 oscilio, 98  
**OSCL\_FILEMGMT\_PERMS\_EXECUTE**  
 oscilio, 98  
**OSCL\_FILEMGMT\_PERMS\_READ**  
 oscilio, 98  
**OSCL\_FILEMGMT\_PERMS\_WRITE**  
 oscilio, 98  
**Oscl\_FileServer**, 193  
 Oscl\_FileServer, 193  
**Oscl\_FileServer**  
 ~Oscl\_FileServer, 193  
 Close, 193  
 Connect, 193  
 Oscl\_DeleteFile, 193, 194  
 Oscl\_File, 194  
 Oscl\_FileServer, 193  
 OsclNativeFile, 194  
**OSCL\_FIRST\_CATCH**  
 osclerror, 89  
**OSCL\_FIRST\_CATCH\_ANY**  
 osclerror, 89  
**oscl\_floor**  
 osclutil, 74  
**OSCL\_FREE**  
 osclmemory, 56  
**oscl\_free**

osclmemory, 56  
**OSCL\_FSSTAT**  
     osclio, 97  
**oscl\_fsstat**, 195  
     freebytes, 195  
     totalbytes, 195  
**OSCL\_FUNCTION\_PTR**  
     osclconfig\_compiler\_warnings.h, 806  
**oscl\_getcwd**  
     osclio, 99, 100  
**OSCL\_GetLastError**  
     osclerror, 93  
**OSCL\_HAS\_ANDROID\_FILE\_IO\_SUPPORT**  
     osclconfig.h, 803  
**OSCL\_HAS\_ANDROID\_SUPPORT**  
     osclconfig, 22  
     osclconfig.h, 803  
**OSCL\_HAS\_ANSI\_64BIT\_FILE\_IO\_-  
    SUPPORT**  
     osclconfig\_io.h, 815  
**OSCL\_HAS\_ANSI\_FILE\_IO\_SUPPORT**  
     osclconfig\_io.h, 815  
**OSCL\_HAS\_ANSI\_MATH\_SUPPORT**  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_ANSI\_MEMORY\_FUNCS**  
     osclconfig\_ansi\_memory.h, 804  
**OSCL\_HAS\_ANSI\_STDIO\_SUPPORT**  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT**  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_ANSI\_STRING\_SUPPORT**  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_ANSI\_WIDE\_STRING\_-  
    SUPPORT**  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_BASIC\_LOCK**  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_BERKELEY\_SOCKETS**  
     osclconfig, 22  
     osclconfig\_io.h, 815  
**OSCL\_HAS\_ERRNO\_H**  
     osclconfig\_error.h, 807  
**OSCL\_HAS\_EXCEPTIONS**  
     osclconfig\_error.h, 807  
**OSCL\_HAS\_GLOB**  
     osclconfig\_io.h, 815  
**OSCL\_HAS\_GLOBAL\_NEW\_DELETE**  
     osclconfig\_memory.h, 827  
  
     osclmemory, 56  
**OSCL\_HAS\_GLOBAL\_VARIABLE\_-  
    SUPPORT**  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_HEAP\_BASE\_SUPPORT**  
     osclconfig\_memory.h, 827  
**OSCL\_HAS\_IPHONE\_SUPPORT**  
     osclconfig, 22  
     osclconfig\_unix\_android.h, 842  
**OSCL\_HAS\_LARGE\_FILE\_SUPPORT**  
     osclconfig\_io.h, 815  
**OSCL\_HAS\_MSWIN\_FILE\_IO\_SUPPORT**  
     osclconfig\_io.h, 815  
**OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT**  
     osclconfig, 22  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_MSWIN\_SUPPORT**  
     osclconfig, 22  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_NATIVE\_FILE\_CACHE\_-  
    ENABLE**  
     osclconfig\_io.h, 815  
**OSCL\_HAS\_NON\_PREEMPTIVE\_-  
    THREAD\_SUPPORT**  
     osclconfig\_proc\_unix\_android.h, 834  
     osclconfig\_proc\_unix\_common.h, 836  
**OSCL\_HAS\_PACKED\_STRUCT**  
     osclconfig.h, 803  
**OSCL\_HAS\_PRAGMA\_PACK**  
     osclconfig, 22  
**OSCL\_HAS\_PTHREAD\_SUPPORT**  
     osclconfig, 22  
     osclconfig\_proc\_unix\_android.h, 834  
     osclconfig\_proc\_unix\_common.h, 836  
**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, 815  
**OSCL\_HAS\_RUNTIME\_LIB\_LOADING\_-  
    SUPPORT**  
     osclconfig\_lib.h, 824  
**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, 834  
 osclconfig\_proc\_unix\_common.h, 836  
**OSCL\_HAS\_SETJMP\_H**  
 osclconfig\_error.h, 807  
**OSCL\_HAS\_SINGLETON\_SUPPORT**  
 osclbase, 34  
**OSCL\_HAS\_SOCKET\_SUPPORT**  
 osclconfig\_io.h, 815  
**OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_-FUNCTION**  
 osclconfig, 23  
 osclconfig\_io.h, 815  
**OSCL\_HAS\_SYMBIAN\_DNS\_SERVER**  
 osclconfig, 23  
 osclconfig\_io.h, 815  
**OSCL\_HAS\_SYMBIAN\_ERRORTRAP**  
 osclconfig, 23  
 osclconfig\_error.h, 807  
**OSCL\_HAS\_SYMBIAN\_MATH**  
 osclconfig, 23  
 osclconfig\_util.h, 847  
**OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS**  
 osclconfig, 23  
 osclconfig\_memory.h, 827  
**OSCL\_HAS\_SYMBIAN\_SCHEDULER**  
 osclconfig, 23  
 osclconfig\_proc\_unix\_android.h, 834  
 osclconfig\_proc\_unix\_common.h, 836  
**OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER**  
 osclconfig, 23  
 osclconfig\_io.h, 815  
**OSCL\_HAS\_SYMBIAN\_SUPPORT**  
 osclconfig, 23  
 osclconfig\_unix\_android.h, 842  
 osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_SYMBIAN\_TIMERS**  
 osclconfig, 23  
 osclconfig\_util.h, 847  
**OSCL\_HAS\_THREAD\_SUPPORT**  
 osclconfig\_proc\_unix\_android.h, 834  
 osclconfig\_proc\_unix\_common.h, 836  
**OSCL\_HAS\_TLS\_SUPPORT**  
 osclconfig\_unix\_android.h, 842  
 osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_UNICODE\_SUPPORT**  
 osclconfig\_unix\_android.h, 842  
 osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_UNIX\_SUPPORT**  
 osclconfig, 23  
 osclconfig\_unix\_android.h, 842  
 osclconfig\_unix\_common.h, 846  
**OSCL\_HAS\_UNIX\_TIME\_FUNCS**  
 osclconfig, 23  
 osclconfig\_time.h, 837  
**oscl\_heapbase.h**, 703  
**OSCL\_HeapString**, 196  
 osclutil, 74, 75  
**OSCL\_HeapString**  
 chartype, 197  
 optype, 197  
**OSCL\_String**, 197  
 other\_chartype, 197  
**OSCL\_HeapStringA**, 198  
 OSCL\_HeapStringA, 199, 200  
**OSCL\_HeapStringA**  
~OSCL\_HeapStringA, 200  
 chartype, 199  
 get\_cstr, 200  
 get\_maxsize, 200  
 get\_size, 201  
 get\_str, 201  
 operator=, 201  
 optype, 199  
 OSCL\_HeapStringA, 199, 200  
**OSCL\_String**, 202  
 other\_chartype, 199  
 set, 201, 202  
**OSCL\_IMPORT\_REF**  
 osclconfig.h, 803  
**oscl\_init.h**, 704  
**OSCL\_INLINE**  
 osclbase, 34  
**Oscl\_Int64\_Utils**, 203  
 get\_int64\_lower32, 204  
 get\_int64\_middle32, 204  
 get\_int64\_upper32, 204  
 get\_uint64\_lower32, 204  
 get\_uint64\_middle32, 204  
 get\_uint64\_upper32, 204  
 set\_int64, 204  
 set\_uint64, 204  
**oscl\_int64\_utils.h**, 705  
 \_OsclInteger64Transport, 705  
**OSCL\_INTEGERS\_WORD\_ALIGNED**  
 osclconfig, 23  
**OSCL\_IO\_EXTENSION\_MAXLEN**  
 oscilio, 97  
**OSCL\_IO\_FILENAME\_MAXLEN**  
 oscilio, 97  
**oscl\_ip\_socket.h**, 706  
**OSCL\_IPPROTO\_IP**  
 osclconfig\_io.h, 815  
**OSCL\_IPPROTO\_TCP**  
 osclconfig\_io.h, 815  
**OSCL\_IPPROTO\_UDP**  
 osclconfig\_io.h, 815  
**oscl\_isdigit**

---

osclutil, 69 <b>OSCL_IsErrnoSupported</b> osclerror, 93 <b>oscl_isLetter</b> osclbase, 38 <b>OSCL_JUMP_MAX_JUMP_MARKS</b> osclerror, 89 <b>OSCL_LAST_CATCH</b> osclerror, 89 <b>OSCL_LEAVE</b> osclerror, 89 <b>Oscl_Less</b> , 205 operator(), 205 <b>OSCL_LIB_READ_DEBUG_LIBS</b> osclconfig_lib.h, 824 <b>Oscl_Linked_List</b> , 206 ~Oscl_Linked_List, 206 add_element, 207 add_to_front, 207 check_list, 207 clear, 207 dequeue_element, 207 get_element, 207 get_first, 208 get_index, 208 get_next, 208 get_num_elements, 208 insert_element, 208 move_to_end, 209 move_to_front, 209 Oscl_Linked_List, 206 remove_element, 209 <b>oscl_linked_list.h</b> , 707 <b>Oscl_Linked_List_Base</b> , 211 ~Oscl_Linked_List_Base, 212 add_element, 212 add_to_front, 212 check_list, 212 construct, 212 destroy, 212 get_element, 212 get_first, 213 get_index, 213 get_next, 213 head, 215 insert_element, 213 iterator, 215 move_to_end, 213 move_to_front, 214 num_elements, 215 remove_element, 214 sizeof_T, 215 tail, 215 <b>oscl_lock_base.h</b> , 708	<b>oscl_log</b> osclutil, 75 <b>oscl_log10</b> osclutil, 75 <b>OSCL_MALLOC</b> osclmemory, 57 <b>oscl_malloc</b> osclmemory, 57 <b>Oscl_Map</b> , 216 begin, 219 clear, 219 const_iterator, 218 const_reference, 218 count, 219 empty, 219 end, 219 equal_range, 219 erase, 220 find, 220 insert, 220 iterator, 218 key_comp, 221 key_compare, 218 key_type, 218 lower_bound, 221 max_size, 221 operator=, 221 operator[], 221 Oscl_Map, 218 pair_citerator_citerator, 218 pair_iterator_bool, 218 pair_iterator_iterator, 218 pointer, 218 reference, 218 self, 218 size, 221 size_type, 218 upper_bound, 221, 222 value_comp, 222 value_type, 218 <b>oscl_map.h</b> , 709 <b>OSCL_DISABLE_WARNING_-TRUNCATE_DEBUG_MESSAGE</b> , 709 <b>Oscl_Map::value_compare</b> , 223 comp, 223 operator(), 223 <b>Oscl_Map&lt; Key, T, Alloc, Compare &gt;</b> , 223 value_compare, 223 <b>Oscl_Map&lt; Key, T, Alloc, Compare &gt;</b> <b>Oscl_Map::value_compare</b> , 223 <b>oscl_math.h</b> , 710 <b>OSCL_MAX</b> osclbase, 34
--	--

**OSCL\_MAX\_TRAP\_LEVELS**  
     osclerror, 90  
**oscl\_media\_data.h**, 711  
**oscl\_media\_status.h**, 712  
**oscl\_mem.h**, 713  
     operator delete, 715  
     operator new, 715  
**OSCL\_DISABLE\_WARNING\_-**  
     **TRUNCATE\_DEBUG\_MESSAGE**,  
     715  
**oscl\_mem\_aligned\_size**  
     osclmemory, 60  
**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, 61  
**oscl\_memcpy**  
     osclmemory, 61  
**OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN**  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**oscl\_memmove**  
     osclmemory, 61  
**oscl\_memmove32**  
     osclmemory, 61  
**oscl\_memset**  
     osclmemory, 62  
**oscl\_memsize\_t**  
     osclconfig\_ansi\_memory.h, 804  
**OSCL\_MIN**  
     osclbase, 34  
**oscl\_mkdir**  
     osclio, 100  
**Oscl\_MTLinked\_List**, 225  
     ~Oscl\_MTLinked\_List, 225  
     add\_element, 226  
     add\_to\_front, 226  
     dequeue\_element, 226  
     get\_element, 226  
     get\_index, 226  
     move\_to\_end, 226  
     move\_to\_front, 227  
**Oscl\_MTLinked\_List**, 225  
     remove\_element, 227  
     the\_list, 227  
**oscl\_mutex.h**, 723  
     OsclNoYieldMutex, 723  
**oscl\_namestring.h**, 724  
**OSCL\_NATIVE\_INT64\_TYPE**  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_NATIVE\_UINT64\_TYPE**  
     osclconfig.h, 803  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_NATIVE\_WCHAR\_TYPE**  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846  
**OSCL\_NEW**  
     osclmemory, 57  
**oscl\_opaque\_type.h**, 725  
**Oscl\_Opaque\_Type\_Alloc**, 229  
     ~Oscl\_Opaque\_Type\_Alloc, 229  
     allocate, 229  
     construct, 229  
     deallocate, 229  
     destroy, 229  
**Oscl\_Opaque\_Type\_Alloc\_LL**, 231  
     ~Oscl\_Opaque\_Type\_Alloc\_LL, 231  
     allocate, 231  
     compare\_data, 231  
     construct, 231  
     deallocate, 231  
     destroy, 232  
     get\_data, 232  
     get\_next, 232  
     set\_next, 232  
**Oscl\_Opaque\_Type\_Compare**, 233  
     ~Oscl\_Opaque\_Type\_Compare, 233  
     compare\_EQ, 233  
     compare\_LT, 233  
     swap, 233  
**OSCL\_PACKED\_STRUCT\_BEGIN**  
     osclconfig.h, 803  
**OSCL\_PACKED\_STRUCT\_END**  
     osclconfig.h, 803  
**OSCL\_PACKED\_VAR**  
     osclbase, 34  
     osclconfig.h, 803  
**Oscl\_Pair**, 235  
     first, 235  
     Oscl\_Pair, 235  
     second, 235  
**OSCL\_PERF\_SUMMARY\_LOGGING**  
     osclproc, 105

**OSCL\_PLACEMENT\_NEW**  
 osclmemory, 57

**oscl\_pow**  
 osclutil, 75

**oscl\_pqueue.h**, 726

**oscl\_pqueue\_test**  
 OsclPriorityQueue, 473

**oscl\_procstatus.h**, 727

**Oscl\_Queue**, 236  
 ~Oscl\_Queue, 237  
 back, 237  
 clear, 237  
 const\_reference, 237  
 front, 238  
 Oscl\_Queue, 237  
 pointer, 237  
 pop, 238  
 push, 238  
 reference, 237  
 size\_type, 237  
 value\_type, 237

**oscl\_queue.h**, 728

**Oscl\_Queue\_Base**, 239  
 ~Oscl\_Queue\_Base, 239  
 bufsize, 241  
 capacity, 240  
 clear, 240  
 construct, 240  
 destroy, 240  
 elems, 241  
 empty, 240  
 ifront, 241  
 irear, 241  
 numelems, 241  
 pop, 240  
 push, 240  
 reserve, 240  
 size, 240  
 sizeof\_T, 241

**oscl\_rand.h**, 729

**OSCL\_RAND\_MAX**  
 osclconfig\_util.h, 847

**Oscl\_Rb\_Tree**, 242  
 ~Oscl\_Rb\_Tree, 244  
 begin, 244  
 clear, 244  
 const\_iterator, 244  
 const\_pointer, 244  
 const\_reference, 244  
 count, 244  
 difference\_type, 244  
 empty, 244  
 end, 244  
 equal\_range, 244

erase, 244  
 find, 244  
 insert\_unique, 244  
 iterator, 244  
 key\_type, 244  
 link\_type, 244  
 lower\_bound, 244  
 max\_size, 244  
 operator=, 244  
 Oscl\_Rb\_Tree, 244  
 pointer, 244  
 reference, 244  
 size, 244  
 size\_type, 244  
 upper\_bound, 244  
 value\_type, 244

**Oscl\_Rb\_Tree\_Base**, 246  
 base\_link\_type, 246  
 rebalance, 246  
 rebalance\_for\_erase, 246  
 rotate\_left, 246  
 rotate\_right, 246

**Oscl\_Rb\_Tree\_Const\_Iterator**, 247  
 base\_link\_type, 248  
 const\_iterator, 248  
 link\_type, 248  
 node, 248  
 operator \*, 248  
 operator!=, 248  
 operator++, 248  
 operator-, 248  
 operator->, 248  
 operator==, 248  
 Oscl\_Rb\_Tree\_Const\_Iterator, 248  
 pointer, 248  
 reference, 248  
 self, 248  
 value\_type, 248

**Oscl\_Rb\_Tree\_Iterator**, 250  
 base\_link\_type, 251  
 iterator, 251  
 link\_type, 251  
 node, 251  
 operator \*, 251  
 operator!=, 251  
 operator++, 251  
 operator-, 251  
 operator->, 251  
 operator==, 251  
 Oscl\_Rb\_Tree\_Iterator, 251  
 pointer, 251  
 reference, 251  
 self, 251  
 value\_type, 251

**Oscl\_Rb\_Tree\_Node**, [253](#)  
     link\_type, [253](#)  
     value, [253](#)  
     value\_type, [253](#)  
**Oscl\_Rb\_Tree\_Node\_Base**  
     black, [254](#)  
     red, [254](#)  
**Oscl\_Rb\_Tree\_Node\_Base**, [254](#)  
     base\_link\_type, [254](#)  
     color, [255](#)  
     color\_type, [254](#)  
     left, [255](#)  
     maximum, [255](#)  
     minimum, [255](#)  
     parent, [255](#)  
     RedBl, [254](#)  
     right, [255](#)  
**OSCL\_READSET\_FLAG**  
     oscl\_socket\_serv\_imp\_pv.h, [767](#)  
**OSCL\_REALLOC**  
     osclmemory, [57](#)  
**oscl\_realloc**  
     osclmemory, [57](#)  
**oscl\_refcounter.h**, [730](#)  
**oscl\_refcounter\_memfrag.h**, [731](#)  
**oscl\_registry\_access\_client.h**, [732](#)  
**oscl\_registry\_client.h**, [733](#)  
**oscl\_registry\_client\_impl.h**, [734](#)  
**oscl\_registry\_serv\_impl.h**, [735](#)  
**oscl\_registry\_serv\_impl\_global.h**, [736](#)  
**oscl\_registry\_serv\_impl\_tls.h**, [737](#)  
**oscl\_registry\_types.h**, [738](#)  
**OSCL\_REINTERPRET\_CAST**  
     osclbase, [34](#)  
**OSCL\_RELEASE\_BUILD**  
     osclconfig.h, [803](#)  
**oscl\_rename**  
     osclio, [100, 101](#)  
**OSCL\_REQUEST\_ERR\_CANCEL**  
     osclproc, [106](#)  
**OSCL\_REQUEST\_ERR\_GENERAL**  
     osclproc, [106](#)  
**OSCL\_REQUEST\_ERR\_NONE**  
     osclproc, [106](#)  
**OSCL\_REQUEST\_PENDING**  
     osclproc, [106](#)  
**oscl\_rmdir**  
     osclio, [101](#)  
**oscl\_scheduler.h**, [739](#)  
**oscl\_scheduler\_ao.h**, [740](#)  
**oscl\_scheduler\_aobase.h**, [741](#)  
**oscl\_scheduler\_readyq.h**, [742](#)  
**oscl\_scheduler\_threadcontext.h**, [743](#)  
**oscl\_scheduler\_tuneables.h**, [744](#)  
**oscl\_scheduler\_types.h**, [745](#)  
**OSCL\_SD\_BOTH**  
     osclconfig\_io.h, [815](#)  
**OSCL\_SD\_RECEIVE**  
     osclconfig\_io.h, [815](#)  
**OSCL\_SD\_SEND**  
     osclconfig\_io.h, [815](#)  
**Oscl\_Select1st**, [256](#)  
     operator(), [256](#)  
**oscl\_semaphore.h**, [746](#)  
**OSCL\_SetLastError**  
     osclerror, [93](#)  
**oscl\_shared\_ptr.h**, [747](#)  
**oscl\_sin**  
     osclutil, [76](#)  
**oscl\_singleton.h**, [748](#)  
**OSCL\_SINGLETON\_ID\_CPM\_PLUGIN**,  
     [749](#)  
**OSCL\_SINGLETON\_ID\_LAST**, [749](#)  
**OSCL\_SINGLETON\_ID\_OMX**, [749](#)  
**OSCL\_SINGLETON\_ID\_-**  
     OMXMASTERCORE, [749](#)  
**OSCL\_SINGLETON\_ID\_OSCLMEM**,  
     [749](#)  
**OSCL\_SINGLETON\_ID\_-**  
     OSCLREGISTRY, [749](#)  
**OSCL\_SINGLETON\_ID\_-**  
     PAYLOADPARSER, [749](#)  
**OSCL\_SINGLETON\_ID\_-**  
     PVERRORTRAP, [749](#)  
**OSCL\_SINGLETON\_ID\_PVLOGGER**,  
     [749](#)  
**OSCL\_SINGLETON\_ID\_-**  
     PVMFRECOGNIZER, [749](#)  
**OSCL\_SINGLETON\_ID\_-**  
     PVSCHEDULER, [749](#)  
**OSCL\_SINGLETON\_ID\_-**  
     SDPMEDIAPARSER, [749](#)  
**OSCL\_SINGLETON\_ID\_TEST**, [749](#)  
**OSCL\_SINGLETON\_ID\_TICKCOUNT**,  
     [749](#)  
**OSCL\_SINGLETON\_ID\_-**  
     WMDRMLOCK, [749](#)  
**OSCL\_SINGLETON\_ID\_CPM\_PLUGIN**  
     oscl\_singleton.h, [749](#)  
**OSCL\_SINGLETON\_ID\_LAST**  
     oscl\_singleton.h, [749](#)  
**OSCL\_SINGLETON\_ID\_OMX**  
     oscl\_singleton.h, [749](#)  
**OSCL\_SINGLETON\_ID\_-**  
     OMXMASTERCORE  
     oscl\_singleton.h, [749](#)  
**OSCL\_SINGLETON\_ID\_OSCLMEM**  
     oscl\_singleton.h, [749](#)

OSCL\_SINGLETON\_ID\_OSCLREGISTRY  
     oscl\_singleton.h, 749  
 OSCL\_SINGLETON\_ID\_PAYLOADPARSER  
     oscl\_singleton.h, 749  
 OSCL\_SINGLETON\_ID\_PVERRORTRAP  
     oscl\_singleton.h, 749  
 OSCL\_SINGLETON\_ID\_PVLOGGER  
     oscl\_singleton.h, 749  
 OSCL\_SINGLETON\_ID\_-  
     PVMFRECOGNIZER  
     oscl\_singleton.h, 749  
 OSCL\_SINGLETON\_ID\_PVSCHEDULER  
     oscl\_singleton.h, 749  
 OSCL\_SINGLETON\_ID\_-  
     SDPMEDIAPARSER  
     oscl\_singleton.h, 749  
 OSCL\_SINGLETON\_ID\_TEST  
     oscl\_singleton.h, 749  
 OSCL\_SINGLETON\_ID\_TICKCOUNT  
     oscl\_singleton.h, 749  
 OSCL\_SINGLETON\_ID\_WMDRMLOCK  
     oscl\_singleton.h, 749  
 oscl\_snprintf  
     osclutil, 76  
 oscl\_snprintf.h, 750  
 OSCL SOCK\_DGRAM  
     osclconfig\_io.h, 815  
 OSCL SOCK\_STREAM  
     osclconfig\_io.h, 815  
 oscl\_socket.h, 751  
 oscl\_socket\_accept.h, 752  
 oscl\_socket\_bind.h, 753  
 oscl\_socket\_connect.h, 754  
 oscl\_socket\_imp.h, 755  
 oscl\_socket\_imp\_base.h, 756  
 oscl\_socket\_imp\_pv.h, 757  
     PVSOCK\_ERR\_BAD\_PARAM, 757  
     PVSOCK\_ERR\_NOT\_IMPLEMENTED,  
         757  
     PVSOCK\_ERR\_NOT\_SUPPORTED, 757  
     PVSOCK\_ERR\_SERV\_NOT\_-  
         CONNECTED, 757  
     PVSOCK\_ERR\_SOCK\_NO\_SERV, 757  
     PVSOCK\_ERR\_SOCK\_NOT\_-  
         CONNECTED, 757  
     PVSOCK\_ERR\_SOCK\_NOT\_OPEN, 757  
 oscl\_socket\_listen.h, 758  
     OSCL\_SOCKET\_LISTEN\_H\_-  
         INCLUDEDd, 758  
 OSCL\_SOCKET\_LISTEN\_H\_INCLUDEDd  
     oscl\_socket\_listen.h, 758  
 oscl\_socket\_method.h, 759  
     MSEC\_TO\_MICROSEC, 759  
 oscl\_socket\_recv.h, 760  
     oscl\_socket\_recv\_from.h, 761  
     oscl\_socket\_request.h, 762  
     oscl\_socket\_send.h, 763  
     oscl\_socket\_send\_to.h, 764  
     oscl\_socket\_serv\_imp.h, 765  
     oscl\_socket\_serv\_imp\_base.h, 766  
     oscl\_socket\_serv\_imp\_pv.h, 767  
         OSCL\_EXCEPTSET\_FLAG, 767  
         OSCL\_READSET\_FLAG, 767  
         OSCL\_WRITESET\_FLAG, 767  
 oscl\_socket\_serv\_imp\_reqlist.h, 768  
 oscl\_socket\_shutdown.h, 769  
 oscl\_socket\_stats.h  
     EOsclSocket\_DataRecv, 771  
     EOsclSocket\_DataSent, 771  
     EOsclSocket\_Except, 770  
     EOsclSocket\_OS, 770  
     EOsclSocket\_Readable, 770  
     EOsclSocket\_RequestAO\_Canceled, 770  
     EOsclSocket\_RequestAO\_Error, 770  
     EOsclSocket\_RequestAO\_Success, 770  
     EOsclSocket\_RequestAO\_Timeout, 770  
     EOsclSocket\_ServPoll, 770  
     EOsclSocket\_ServRequestCancelIssued,  
         771  
     EOsclSocket\_ServRequestComplete, 771  
     EOsclSocket\_ServRequestIssued, 770  
     EOsclSocket\_Writable, 770  
     EOsclSocketServ\_LastEvent, 770  
     EOsclSocketServ\_LoopsckError, 771  
     EOsclSocketServ\_LoopsckOk, 771  
     EOsclSocketServ\_SelectActivity, 770  
     EOsclSocketServ\_SelectNoActivity, 770  
     EOsclSocketServ\_SelectRescheduleAsap,  
         770  
     EOsclSocketServ\_SelectReschedulePoll,  
         770  
 oscl\_socket\_stats.h, 770  
     TOsclSocketServStatEvent, 770  
     TOsclSocketStatEvent, 770  
 oscl\_socket\_tuneables.h, 772  
     PV\_OSCL\_SOCKET\_1MB\_RECV\_BUF,  
         772  
     PV\_OSCL\_SOCKET\_SERVER\_-  
         LOGGER\_OUTPUT, 772  
     PV\_OSCL\_SOCKET\_STATS\_LOGGING,  
         772  
     PV\_SOCKET\_REQUEST\_AO\_-  
         PRIORITY, 772  
     PV\_SOCKET\_SERVER, 772  
     PV\_SOCKET\_SERVER\_AO\_-  
         INTERVAL\_MSEC, 773  
     PV\_SOCKET\_SERVER\_AO\_PRIORITY,  
         773

---

PV_SOCKET_SERVER_IS_THREAD, 773	OSCL_SOL_SOCKET osclconfig_io.h, 815
PV_SOCKET_SERVER_SELECT, 773	OSCL_SOL_TCP osclconfig_io.h, 815
PV_SOCKET_SERVER_SELECT_- LOOPBACK_SOCKET, 773	OSCL_SOL_UDP osclconfig_io.h, 815
PV_SOCKET_SERVER_SELECT_- TIMEOUT_MSEC, 773	oscl_sqrt osclutil, 76
PV_SOCKET_SERVER_THREAD_- PRIORITY, 773	OSCL_StackString, 257 osclutil, 76, 77
PV_SOCKET_SERVI_STATS, 773	OSCL_StackString chartype, 258 otype, 258 OSCL_String, 258 other_chartype, 258
oscl_socket_types.h	oscl_stat osclio, 101, 102
EPVIPAddMembership, 775	OSCL_STAT_BUF osclio, 97
EPVIMulticastTTL, 775	oscl_stat_buf, 259 mode, 259 perms, 259
EPVIPProtoIP, 775	oscl_statsfs osclio, 102
EPVIPProtoTCP, 775	OSCL_STATIC_CAST osclbase, 34
EPVIPTOS, 775	oscl_stdstring.h, 776
EPVSocket, 775	oscl_str_escape_xml osclutil, 77
EPVSocket_Last, 775	oscl_str_is_valid_utf8 osclutil, 77
EPVSocketAccept, 775	oscl_str_need_escape_xml osclutil, 78
EPVSocketBind, 775	oscl_str_ptr_len.h, 778
EPVSocketBothShutdown, 775	oscl_str_truncate_utf8 osclutil, 78
EPVSocketCancel, 774	oscl_str_unescape_uri osclutil, 78, 79
EPVSocketConnect, 775	oscl_streat osclbase, 39
EPVSocketFailure, 774	oscl_strchr osclbase, 39, 40
EPVSocketListen, 775	oscl_strcmp osclbase, 40
EPVSocketNotImplemented, 775	OSCL_StrError osclerror, 93
EPVSocketPending, 774	OSCL_String, 260 ~OSCL_String, 261
EPVSocketRecv, 775	append_rep, 261
EPVSocketRecvFrom, 775	chartype, 261
EPVSocketRecvShutdown, 775	get_cstr, 261
EPVSocketSend, 775	get_maxsize, 261
EPVSocketSendShutdown, 775	get_size, 262
EPVSocketSendTo, 775	get_str, 262
EPVSocketShutdown, 775	
EPVSocketSuccess, 774	
EPVSocketTimeout, 774	
EPVSockReuseAddr, 775	
oscl_socket_types.h, 774	
PVNETWORKADDRESS_LEN, 774	
TPVSocketEvent, 774	
TPVSocketFxn, 775	
TPVSocketOptionLevel, 775	
TPVSocketOptionName, 775	
TPVSocketShutdown, 775	
OSCL_SOCKOPT_IP_ADDMEMBERSHIP osclconfig_io.h, 815	
OSCL_SOCKOPT_IP_MULTICAST_TTL osclconfig_io.h, 815	
OSCL_SOCKOPT_IP_TOS osclconfig_io.h, 815	
OSCL_SOCKOPT_SOL_REUSEADDR osclconfig_io.h, 815	
OSCL_SOL_IP osclconfig_io.h, 815	

---

hash, 262  
 is\_writable, 262  
 operator!=, 262  
 operator+=, 262  
 operator<, 262  
 operator<=, 263  
 operator=, 263  
 operator==, 263  
 operator>, 263  
 operator>=, 263  
 operator[], 263  
 OSCL\_FastString, 178  
 OSCL\_HeapString, 197  
 OSCL\_HeapStringA, 202  
 OSCL\_StackString, 258  
 OSCL\_String, 261  
 read, 263  
 set\_len, 263  
 set\_rep, 263, 264  
 setrep\_to\_char, 264  
 write, 264  
 oscl\_string.h, 779  
 oscl\_string\_containers.h, 780  
 oscl\_string\_rep.h, 781  
 oscl\_string\_uri.h, 782  
 oscl\_string\_utf8.h, 783  
 oscl\_string\_utils.h, 784  
 oscl\_string\_xml.h, 785  
 oscl\_strlen  
     osclbase, 40  
 oscl\_strncat  
     osclbase, 41  
 oscl\_strcmp  
     osclbase, 41, 42  
 oscl\_strncpy  
     osclbase, 42  
 oscl strrchr  
     osclbase, 43  
 oscl\_strset  
     osclbase, 43  
 oscl strstr  
     osclbase, 43, 44  
 Oscl\_Tag, 265  
     ~Oscl\_Tag, 265  
     operator<, 265  
     Oscl\_Tag, 265  
     tag, 265  
     tagAllocator, 265  
 Oscl\_Tag\_Base, 267  
     operator(), 268  
     size\_type, 268  
     tag\_ancestor, 268  
     tag\_base\_type, 268  
     tag\_base\_unit, 268  
     tag\_cmp, 268  
     tag\_copy, 268  
     tag\_depth, 268  
     tag\_len, 268  
 Oscl\_TagTree, 269  
     Oscl\_TagTree, 270  
 Oscl\_TagTree  
     ~Oscl\_TagTree, 270  
     begin, 270  
     children\_type, 270  
     clear, 271  
     count, 271  
     empty, 271  
     end, 271  
     erase, 271  
     find, 271  
     insert, 272  
     map\_type, 270  
     node\_ptr, 270  
     node\_type, 270  
     operator=, 272  
     operator[], 272  
     Oscl\_TagTree, 270  
     pair\_iterator\_bool, 270  
     size, 272  
     size\_type, 270  
     tag\_base\_type, 270  
     tag\_type, 270  
     value\_type, 270  
 oscl\_tagtree.h, 786  
     OSCL\_DISABLE\_WARNING\_-  
         TRUNCATE\_DEBUG\_MESSAGE,  
         786  
 Oscl\_TagTree::const\_iterator, 273  
 Oscl\_TagTree::const\_iterator  
     const\_iterator, 274  
     mapit, 274  
     mapiter, 274  
     operator \*, 274  
     operator!=, 274  
     operator++, 274  
     operator-, 274  
     operator->, 274  
     operator==, 274  
     pointer, 274  
     reference, 274  
     self, 274  
 Oscl\_TagTree::iterator, 276  
 Oscl\_TagTree::iterator  
     iterator, 277  
     mapit, 277  
     mapiter, 277  
     operator \*, 277  
     operator!=, 277

operator++, 277  
 operator-, 277  
 operator->, 277  
 operator==, 277  
 pointer, 277  
 reference, 277  
 self, 277  
**Oscl\_TagTree::Node**, 279  
**Oscl\_TagTree::Node**  
 children, 280  
 children\_type, 280  
 depth, 280  
 Node, 280  
 parent, 280  
 sort\_children, 280  
 tag, 280  
 value, 280  
**Oscl\_TAlloc**, 281  
   ~Oscl\_TAlloc, 282  
   address, 282  
   alloc\_and\_construct, 282  
   alloc\_and\_construct\_fl, 282  
   allocate, 282  
   allocate\_fl, 282  
   const\_pointer, 282  
   const\_reference, 282  
   construct, 282  
   deallocate, 282  
   destroy, 282  
   destruct\_and\_dealloc, 282  
   pointer, 282  
   reference, 282  
   size\_type, 282  
   value\_type, 282  
**Oscl\_TAlloc::rebind**, 284  
   other, 284  
**oscl\_tan**  
   osclutil, 79  
**OSCL\_TCHAR**  
   osclbase, 35  
**oscl\_tcp\_socket.h**, 787  
**OSCL\_TEMPLATED\_DESTRUCTOR\_CALL**  
   osclbase, 34  
   osclconfig.h, 803  
**oscl\_thread.h**  
   EOscIThreadTerminate\_Join, 789  
   EOscIThreadTerminate\_Kill, 789  
   EOscIThreadTerminate\_NOP, 789  
   Start\_on\_creation, 788  
   Suspend\_on\_creation, 788  
   ThreadPriorityAboveNormal, 789  
   ThreadPriorityBelowNormal, 789  
   ThreadPriorityHighest, 789  
   ThreadPriorityLow, 788  
 ThreadPriorityLowest, 788  
 ThreadPriorityNormal, 789  
 ThreadPriorityTimeCritical, 789  
**oscl\_thread.h**, 788  
   OsclThread\_State, 788  
   OsclThreadPriority, 788  
   TOscIThreadFuncPtr, 788  
   TOscIThreadTerminate, 789  
**OSCL\_THREAD DECL**  
   osclconfig\_proc\_unix\_android.h, 834  
   osclconfig\_proc\_unix\_common.h, 836  
**oscl\_tickcount.h**, 790  
**oscl\_time.h**, 791  
**oscl\_timer.h**, 793  
**oscl\_tls.h**, 794  
**OSCL\_TLS\_BASE\_SLOTS**  
   osclbase, 34  
**OSCL\_TLS\_EXTERNAL\_SLOTS**  
   osclbase, 34  
**OSCL\_TLS\_GET\_FUNC**  
   osclconfig\_unix\_android.h, 842  
   osclconfig\_unix\_common.h, 846  
**OSCL\_TLS\_ID\_BASE\_LAST**  
   osclbase, 46  
**OSCL\_TLS\_ID\_ERRORHOOK**  
   osclbase, 46  
**OSCL\_TLS\_ID\_MAGICNUM**  
   osclbase, 46  
**OSCL\_TLS\_ID\_OSCLREGISTRY**  
   osclbase, 46  
**OSCL\_TLS\_ID\_PAYLOADPARSER**  
   osclbase, 46  
**OSCL\_TLS\_ID\_PVERRORTRAP**  
   osclbase, 46  
**OSCL\_TLS\_ID\_PVLOGGER**  
   osclbase, 46  
**OSCL\_TLS\_ID\_PVMFRECOGNIZER**  
   osclbase, 46  
**OSCL\_TLS\_ID\_PVSCHEDULER**  
   osclbase, 46  
**OSCL\_TLS\_ID\_SDPMEDIAPARSER**  
   osclbase, 46  
**OSCL\_TLS\_ID\_SQLITE3**  
   osclbase, 46  
**OSCL\_TLS\_ID\_TEST**  
   osclbase, 46  
**OSCL\_TLS\_ID\_WMDRM**  
   osclbase, 46  
**OSCL\_TLS\_IS\_KEYED**  
   osclconfig\_unix\_android.h, 842  
   osclconfig\_unix\_common.h, 846  
**OSCL\_TLS\_KEY\_CREATE\_FUNC**  
   osclconfig\_unix\_android.h, 842  
   osclconfig\_unix\_common.h, 846

**OSCL\_TLS\_KEY\_DELETE\_FUNC**  
 osclconfig\_unix\_android.h, 842  
 osclconfig\_unix\_common.h, 846

**OSCL\_TLS\_MAX\_SLOTS**  
 osclbase, 34

**OSCL\_TLS\_STORE\_FUNC**  
 osclconfig\_unix\_android.h, 842  
 osclconfig\_unix\_common.h, 846

**oscl\_tolower**  
 osclbase, 44

**OSCL\_TRAP\_ALLOC\_NEW**  
 osclmemory, 57

**OSCL\_TRAP\_AUDIT\_NEW**  
 osclmemory, 58

**OSCL\_TRAP\_NEW**  
 osclmemory, 58

**OSCL\_TRAPSTACK\_POP**  
 osclerror, 90

**OSCL\_TRAPSTACK\_POPDEALLOC**  
 osclerror, 90

**OSCL\_TRAPSTACK\_PUSH**  
 osclerror, 90

**oscl\_tree.h**, 795
 

- OSCL\_DISABLE\_WARNING\_-**
- TRUNCATE\_DEBUG\_MESSAGE, 795

**OSCL\_TRY**  
 osclerror, 90

**OSCL\_TRY\_NO\_TLS**  
 osclerror, 90

**OSCL\_TStrPtrLen**  
 osclutil, 69

**oscl\_types.h**, 796

**oscl\_udp\_socket.h**, 797

**oscl\_UnicodeToUTF8**  
 osclutil, 79

**OSCL\_UNSIGNED\_CONST**  
 osclbase, 34  
 osclconfig.h, 803

**OSCL\_UNUSED\_ARG**  
 osclbase, 34

**OSCL\_UNUSED\_RETURN**  
 osclbase, 34

**oscl\_utf8conv.h**, 798

**oscl\_UTF8ToUnicode**  
 osclutil, 80

**oscl\_uuid.h**, 799
 

- BYTES\_IN\_UUID\_ARRAY, 799
- EMPTY\_UUID, 799
- OsclUid32, 799

**oscl\_uuid\_utils.h**, 800
 

- PV\_CHAR\_CLOSE\_BRACKET, 800
- PV\_CHAR\_COMMA, 800

**Oscl\_Vector**, 285
 

- ~Oscl\_Vector, 286
- back, 287
- begin, 287
- clear, 287
- const\_iterator, 286
- const\_reference, 286
- destroy, 287
- end, 287
- erase, 287
- front, 288
- insert, 288
- iterator, 286
- operator=, 288
- operator[], 288
- Oscl\_Vector, 286
- pointer, 286
- pop\_back, 288
- push\_back, 289
- push\_front, 289
- reference, 286
- value\_type, 286

**oscl\_vector.h**, 801
 

- Oscl\_Vector\_Base, 290
- ~Oscl\_Vector\_Base, 291
- assign\_vector, 291
- bufsize, 293
- capacity, 291
- construct, 291
- destroy, 291
- elems, 293
- empty, 291
- erase, 291, 292
- insert, 292
- numelems, 293
- OsclPriorityQueueBase, 293
- pop\_back, 292
- push\_back, 292
- push\_front, 293
- reserve, 293
- size, 293
- sizeof\_T, 293

**OSCL\_VIRTUAL\_BASE**  
 osclbase, 34

**oscl\_vsnprintf**  
 osclutil, 80, 82

**oscl\_wchar**  
 osclbase, 35

**OSCL\_wFastString**, 294
 

- OSCL\_wFastString, 295

**OSCL\_wFastString**  
 ~OSCL\_wFastString, 295

**chartype**, 294

**get\_cstr**, 295

**get\_maxsize**, 295

get\_size, 295  
 get\_str, 295  
 operator=, 295  
 optype, 294  
 OSCL\_wFastString, 295  
 OSCL\_wString, 296  
 other\_chartype, 295  
 set, 296  
 set\_length, 296  
 OSCL\_wHeapString, 297  
 osclutil, 82  
 OSCL\_wHeapString  
   chartype, 298  
   optype, 298  
   OSCL\_wString, 298  
   other\_chartype, 298  
 OSCL\_wHeapStringA, 299  
   OSCL\_wHeapStringA, 300  
 OSCL\_wHeapStringA  
   ~OSCL\_wHeapStringA, 300  
   chartype, 300  
   get\_cstr, 300  
   get\_maxsize, 300  
   get\_size, 300  
   get\_str, 301  
   operator=, 301  
   optype, 300  
   OSCL\_wHeapStringA, 300  
   OSCL\_wString, 301  
   other\_chartype, 300  
   set, 301  
 OSCL\_WRITESET\_FLAG  
   oscl\_socket\_serv\_imp\_pv.h, 767  
 OSCL\_wStackString, 302  
   osclutil, 82  
 OSCL\_wStackString  
   chartype, 303  
   optype, 303  
   OSCL\_wString, 303  
   other\_chartype, 303  
 OSCL\_wString, 304  
   OSCL\_wFastString, 296  
   OSCL\_wHeapString, 298  
   OSCL\_wHeapStringA, 301  
   OSCL\_wStackString, 303  
   OSCL\_wString, 305  
 OSCL\_wString  
   ~OSCL\_wString, 305  
   append\_rep, 305  
   chartype, 305  
   get\_cstr, 305  
   get\_maxsize, 305  
   get\_size, 305  
   get\_str, 305  
   hash, 305  
   is\_writable, 306  
   operator!=, 306  
   operator+=, 306  
   operator<, 306  
   operator<=, 306  
   operator=, 306  
   operator==, 306  
   operator>, 306  
   operator>=, 306  
   operator[], 306  
   OSCL\_wString, 305  
   read, 306  
   set\_len, 307  
   set\_rep, 307  
   setrep\_to\_wide\_char, 307  
   write, 307  
 OSCL\_ZEROIZE  
   osclproc, 105  
 OsclAccept  
   osclconfig\_io.h, 815  
 OsclAcceptMethod, 308  
 OsclAcceptMethod  
   ~OsclAcceptMethod, 308  
   Accept, 308  
   AcceptRequest, 308  
   DiscardAcceptedSocket, 308  
   GetAcceptedSocket, 308  
   NewL, 308  
 OsclAcceptRequest, 309  
   OsclAcceptRequest, 309  
   OsclSocketI, 539  
 OsclAcceptRequest  
   Accept, 309  
   OsclAcceptRequest, 309  
 OsclActiveObject, 310  
   EPriorityHigh, 311  
   EPriorityHighest, 311  
   EPriorityIdle, 311  
   EPriorityLow, 311  
   EPriorityNominal, 311  
   OsclActiveObject, 311  
   OsclExecSchedulerCommonBase, 397  
   PVActiveBase, 615  
   PVActiveStats, 616  
   PVThreadContext, 635  
 OsclActiveObject  
   ~OsclActiveObject, 311  
   AddToScheduler, 311  
   Cancel, 311  
   DoCancel, 312  
   IsBusy, 312  
   OsclActiveObject, 311  
   OsclActivePriority, 311

PendComplete, 312  
 PendForExec, 312  
 Priority, 312  
 RemoveFromScheduler, 312  
 RunError, 312  
 RunIfNotReady, 313  
 SetBusy, 313  
 SetStatus, 313  
 Status, 313  
 StatusRef, 313  
**OsclActivePriority**  
*OsclActiveObject*, 311  
**OsclAllocDestructDealloc**, 314  
**OsclAllocDestructDealloc**  
*~OsclAllocDestructDealloc*, 314  
**OsclAny**  
*osclbase*, 35  
**OsclAOStatus**, 315  
*OsclAOStatus*, 315  
**OsclAOStatus**  
*operator!=*, 315  
*operator<*, 315  
*operator<=*, 315  
*operator=*, 315  
*operator==*, 315  
*operator>*, 315  
*operator>=*, 315  
*OsclAOStatus*, 315  
*Value*, 315  
**OsclAsyncFile**, 316  
**OsclAsyncFile**  
*~OsclAsyncFile*, 317  
*Close*, 317  
*Delete*, 317  
*EndOfFile*, 317  
*Flush*, 317  
*iNumOfRun*, 318  
*iNumOfRunErr*, 318  
*NewL*, 317  
*Open*, 317, 318  
*Read*, 318  
*Seek*, 318  
*Size*, 318  
*Tell*, 318  
*Write*, 318  
**OsclAsyncFileBuffer**, 319  
**OsclAsyncFileBuffer**  
*~OsclAsyncFileBuffer*, 320  
*Buffer*, 320  
*CleanInUse*, 320  
*HasThisOffset*, 320  
*Id*, 320  
*IsInUse*, 320  
*IsValid*, 320  
*Length*, 320  
*NewL*, 320  
*Offset*, 320  
*SetInUse*, 320  
*SetOffset*, 320  
*StartAsyncRead*, 320  
*UpdateData*, 320  
**OsclAuditCB**, 321  
*OsclAuditCB*, 321  
**OsclAuditCB**  
*OsclAuditCB*, 321  
*pAudit*, 321  
*pStatsNode*, 321  
**OsclBase**  
*OsclSingletonRegistry*, 534  
*OsclTLSRegistry*, 596  
*osclbase*  
*\_OSCL\_Abort*, 36  
*ALLOC\_AND\_CONSTRUCT*, 32  
*ALLOCATE*, 32  
*big\_endian\_to\_host*, 36  
*Bind*, 36  
*c\_bool*, 34  
*CTIME\_BUFFER\_SIZE*, 46  
*CtimeStrBuf*, 34  
*EPV\_ARM\_GNUC*, 32  
*EPV\_ARM\_MSEVC*, 32  
*EPV\_ARM\_RVCT*, 32  
*host\_to\_big\_endian*, 36  
*host\_to\_little\_endian*, 36  
*int64*, 35  
*ISO8601TIME\_BUFFER\_SIZE*, 46  
*ISO8601timeStrBuf*, 35  
*ISO8601ToRFC822*, 36  
*little\_endian\_to\_host*, 37  
*mbchar*, 35  
*MICROSECONDS*, 35  
*MILLISECONDS*, 35  
*MSEC\_PER\_SEC*, 46  
*NULL*, 32  
*NULL\_TERM\_CHAR*, 32  
*octet*, 35  
*operator+*, 37  
*operator-*, 37  
*operator==*, 37  
*OSCL\_ABS*, 32  
*OSCL\_ASSERT*, 32  
*OSCL\_Assert*, 37  
*oscl\_CIstrcmp*, 37, 38  
*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,  
     33  
 OSCL\_DLL\_ENTRY\_POINT, 33  
 OSCL\_DLL\_ENTRY\_POINT\_DEFAULT,  
     33  
 OSCL\_DYNAMIC\_CAST, 33  
 OSCL\_HAS\_SINGLETON\_SUPPORT, 34  
 OSCL\_INLINE, 34  
 oscl\_isLetter, 38  
 OSCL\_MAX, 34  
 OSCL\_MIN, 34  
 OSCL\_PACKED\_VAR, 34  
 OSCL\_REINTERPRET\_CAST, 34  
 OSCL\_STATIC\_CAST, 34  
 oscl\_streat, 39  
 oscl\_strchr, 39, 40  
 oscl\_strcmp, 40  
 oscl\_strlen, 40  
 oscl\_strncat, 41  
 oscl\_strncmp, 41, 42  
 oscl\_strncpy, 42  
 oscl strrchr, 43  
 oscl\_strset, 43  
 oscl strstr, 43, 44  
 OSCL\_TCHAR, 35  
 OSCL\_TEMPLATED\_DESTRUCTOR\_-  
     CALL, 34  
 OSCL\_TLS\_BASE\_SLOTS, 34  
 OSCL\_TLS\_EXTERNAL\_SLOTS, 34  
 OSCL\_TLS\_ID\_BASE\_LAST, 46  
 OSCL\_TLS\_ID\_ERRORHOOK, 46  
 OSCL\_TLS\_ID\_MAGICNUM, 46  
 OSCL\_TLS\_ID\_OSCLREGISTRY, 46  
 OSCL\_TLS\_ID\_PAYLOADPARSER, 46  
 OSCL\_TLS\_ID\_PVERRORTRAP, 46  
 OSCL\_TLS\_ID\_PVLOGGER, 46  
 OSCL\_TLS\_ID\_PVMFRECOGNIZER, 46  
 OSCL\_TLS\_ID\_PVSCHEDULER, 46  
 OSCL\_TLS\_ID\_SDPMEDIAPARSER, 46  
 OSCL\_TLS\_ID\_SQLITE3, 46  
 OSCL\_TLS\_ID\_TEST, 46  
 OSCL\_TLS\_ID\_WMDRM, 46  
 OSCL\_TLS\_MAX\_SLOTS, 34  
 oscl\_tolower, 44  
 OSCL\_UNSIGNED\_CONST, 34  
 OSCL\_UNUSED\_ARG, 34  
 OSCL\_UNUSED\_RETURN, 34  
 OSCL\_VIRTUAL\_BASE, 34  
 oscl\_wchar, 35  
 OsclAny, 35  
 OsclFloat, 35  
  
 PV8601TIME\_BUFFER\_SIZE, 46  
 PV8601timeStrBuf, 35  
 PV8601ToRFC822, 44  
 PVMEM\_INST\_LEVEL, 34  
 PVosclBase\_Cleanup, 45  
 PVosclBase\_Init, 45  
 RFC822ToPV8601, 45  
 SECONDS, 35  
 TimeUnits, 35  
 TOscITlsKey, 35  
 uint, 35  
 uint64, 35  
 unix\_ntp\_offset, 46  
 USEC\_PER\_SEC, 46  
 OsclBasicDateStruct  
     osclconfig\_time.h, 837  
 OsclBasicTimeStruct  
     osclconfig\_time.h, 837  
 OsclBind  
     osclconfig\_io.h, 816  
 OsclBindMethod, 322  
 OsclBindMethod  
     ~OsclBindMethod, 322  
     Bind, 322  
     BindRequest, 322  
     NewL, 322  
 OsclBindRequest, 323  
     OsclBindRequest, 323  
 OsclBindRequest  
     Bind, 323  
     OsclBindRequest, 323  
 OsclBinIStream, 324  
     OsclBinIStream, 324  
 OsclBinIStream  
     ~OsclBinIStream, 324  
     get, 324  
     OsclBinIStream, 324  
     Read\_uint8, 324  
 OsclBinIStreamBigEndian, 326  
     OsclBinIStreamBigEndian, 327  
 OsclBinIStreamBigEndian  
     operator>>, 327  
     OsclBinIStreamBigEndian, 327  
     Read, 327  
     Read\_uint16, 327  
     Read\_uint32, 327  
 OsclBinIStreamLittleEndian, 329  
     OsclBinIStreamLittleEndian, 330  
 OsclBinIStreamLittleEndian  
     operator>>, 330  
     OsclBinIStreamLittleEndian, 330  
     Read\_uint16, 330  
     Read\_uint32, 330  
 OsclBinOStream, 331

OsclBinOStream, 331  
 OsclBinOStream  
   ~OsclBinOStream, 331  
   OsclBinOStream, 331  
   write, 331  
 OsclBinOStreamBigEndian, 332  
   OsclBinOStreamBigEndian, 333  
 OsclBinOStreamBigEndian  
   operator<<, 333  
   OsclBinOStreamBigEndian, 333  
   WriteUnsignedLong, 333  
   WriteUnsignedShort, 333  
 OsclBinOStreamLittleEndian, 334  
   OsclBinOStreamLittleEndian, 335  
 OsclBinOStreamLittleEndian  
   operator<<, 335  
   OsclBinOStreamLittleEndian, 335  
   WriteUnsignedLong, 335  
   WriteUnsignedShort, 335  
 OsclBinStream, 336  
   EOF\_STATE, 337  
   FAIL\_STATE, 337  
   GOOD\_STATE, 337  
   OsclBinStream, 337  
 OsclBinStream  
   Attach, 337  
   eof, 337  
   fail, 338  
   firstFragPtr, 339  
   fragsLeft, 339  
   good, 338  
   HaveRoomInCurrentBlock, 338  
   length, 339  
   nextFragPtr, 339  
   numFrags, 339  
   OsclBinStream, 337  
   pBasePosition, 339  
   PositionInBlock, 338  
   pPosition, 339  
   ReserveSpace, 338  
   Seek, 338  
   seekFromCurrentPosition, 338  
   specialFragBuffer, 339  
   state, 339  
   state\_t, 337  
   tellg, 338  
 OsclBuf, 340  
   OsclBuf, 341  
 OsclBuf  
   Delete, 341  
   Des, 341  
   DesC, 341  
   iBuffer, 341  
   iLength, 341  
     iMaxLength, 341  
     Length, 341  
     NewL, 341  
     OsclBuf, 341  
 OsclCloseSocket  
   osclconfig\_io.h, 816  
 OsclCoeActiveScheduler  
   OsclExecSchedulerBase, 391  
   OsclExecSchedulerCommonBase, 397  
   PVThreadContext, 635  
 OsclCoeActiveSchedulerBase  
   PVThreadContext, 635  
 OsclCompareLess, 342  
 OsclCompareLess  
   compare, 342  
 OsclComponentFactory  
   osclutil, 69  
 OsclComponentRegistry, 343  
   OsclComponentRegistry, 344  
 OsclComponentRegistry  
   ~OsclComponentRegistry, 344  
   CloseSession, 344  
   FindExact, 344  
   FindHierarchical, 344  
   iComponentIdCounter, 344  
   iData, 344  
   iMutex, 344  
   iNumSessions, 344  
   OpenSession, 344  
   OsclComponentRegistry, 344  
   Register, 344  
   Unregister, 344  
 OsclComponentRegistryData, 345  
 OsclComponentRegistryData  
   Find, 345  
   iVec, 345  
 OsclComponentRegistryElement, 346  
   OsclComponentRegistryElement, 346  
 OsclComponentRegistryElement  
   ~OsclComponentRegistryElement, 346  
   iComponentId, 346  
   iFactory, 346  
   iId, 346  
   Match, 346  
   operator=, 346  
   OsclComponentRegistryElement, 346  
 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\_ANDROID\_SUPPORT, 22  
 OSCL\_HAS\_BERKELEY\_SOCKETS, 22  
 OSCL\_HAS\_IPHONE\_SUPPORT, 22  
 OSCL\_HAS\_MSWIN\_PARTIAL\_-  
   SUPPORT, 22  
 OSCL\_HAS\_MSWIN\_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, 802  
   \_\_TFS\_\_, 803  
   OSCL\_EXPORT\_REF, 803  
   OSCL\_HAS\_ANDROID\_FILE\_IO\_-  
     SUPPORT, 803  
   OSCL\_HAS\_ANDROID\_SUPPORT, 803  
   OSCL\_HAS\_PACKED\_STRUCT, 803  
   OSCL\_IMPORT\_REF, 803  
   OSCL\_NATIVE\_UINT64\_TYPE, 803  
   OSCL\_PACKED\_STRUCT\_BEGIN, 803  
   OSCL\_PACKED\_STRUCT\_END, 803  
   OSCL\_PACKED\_VAR, 803  
   OSCL\_RELEASE\_BUILD, 803  
   OSCL\_TEMPLATED\_DESTRUCTOR\_-  
     CALL, 803  
   OSCL\_UNSIGNED\_CONST, 803  
   PVLOGGER\_INST\_LEVEL, 803  
 osclconfig\_ansi\_memory.h, 804  
   OSCL\_HAS\_ANSI\_MEMORY\_FUNCS,  
     804  
   oscl\_memsize\_t, 804  
 osclconfig\_check.h, 805  
 osclconfig\_compiler\_warnings.h, 806  
   OSCL\_FUNCTION\_PTR, 806  
 osclconfig\_error.h, 807  
   OSCL\_HAS\_ERRNO\_H, 807  
   OSCL\_HAS\_EXCEPTIONS, 807  
   OSCL\_HAS\_SETJMP\_H, 807  
   OSCL\_HAS\_SYMBIAN\_ERRORTRAP,  
     807  
   osclconfig\_error\_check.h, 808  
   osclconfig\_global\_new\_delete.h, 809  
   osclconfig\_global\_placement\_new.h, 810  
     operator new, 810  
 osclconfig\_io.h, 811  
   OSCL\_AF\_INET, 815  
   OSCL\_FILE\_BUFFER\_MAX\_SIZE, 815  
   OSCL\_HAS\_ANSI\_64BIT\_FILE\_IO\_-  
     SUPPORT, 815  
   OSCL\_HAS\_ANSI\_FILE\_IO\_SUPPORT,  
     815  
   OSCL\_HAS\_BERKELEY\_SOCKETS,  
     815  
   OSCL\_HAS\_GLOB, 815  
   OSCL\_HAS\_LARGE\_FILE\_SUPPORT,  
     815  
   OSCL\_HAS\_MSWIN\_FILE\_IO\_-  
     SUPPORT, 815  
   OSCL\_HAS\_NATIVE\_FILE\_CACHE\_-  
     ENABLE, 815  
   OSCL\_HAS\_PV\_FILE\_CACHE, 815  
   OSCL\_HAS\_SOCKET\_SUPPORT, 815  
   OSCL\_HAS\_SYMBIAN\_-  
     COMPATIBLE\_IO\_FUNCTION,  
       815  
   OSCL\_HAS\_SYMBIAN\_DNS\_SERVER,  
     815  
   OSCL\_HAS\_SYMBIAN\_SOCKET\_-  
     SERVER, 815  
   OSCL IPPROTO\_IP, 815  
   OSCL IPPROTO\_TCP, 815  
   OSCL IPPROTO\_UDP, 815  
   OSCL\_SD\_BOTH, 815  
   OSCL\_SD\_RECEIVE, 815  
   OSCL\_SD\_SEND, 815  
   OSCL SOCK\_DGRAM, 815  
   OSCL SOCK\_STREAM, 815

OSCL\_SOCKOPT\_IP\_-  
     ADDMEMBERSHIP, 815  
 OSCL\_SOCKOPT\_IP\_MULTICAST\_-  
     TTL, 815  
 OSCL\_SOCKOPT\_IP\_TOS, 815  
 OSCL\_SOCKOPT\_SOL\_REUSEADDR,  
     815  
 OSCL\_SOL\_IP, 815  
 OSCL\_SOL\_SOCKET, 815  
 OSCL\_SOL\_TCP, 815  
 OSCL\_SOL\_UDP, 815  
 OsclAccept, 815  
 OsclBind, 816  
 OsclCloseSocket, 816  
 OsclConnect, 816  
 OsclConnectComplete, 816  
 OsclGetAsyncSockErr, 816  
 OsclGetDottedAddr, 816  
 OsclGetDottedAddrVector, 817  
 OsclGethostbyname, 817  
 OsclGetPeerName, 817  
 OsclJoin, 817  
 OsclListen, 818  
 OsclMakeInAddr, 818  
 OsclMakeSockAddr, 818  
 OsclPipe, 818  
 OsclReadFD, 818  
 OsclRecv, 818  
 OsclRecvFrom, 818  
 OsclSend, 819  
 OsclSendTo, 819  
 OsclSetNonBlocking, 819  
 OsclSetRecvBufferSize, 819  
 OsclSetSockOpt, 819  
 OsclShutdown, 819  
 OsclSocket, 820  
 OsclSocketCleanup, 820  
 OsclSocketSelect, 820  
 OsclSocketStartup, 820  
 OsclUnMakeInAddr, 820  
 OsclUnMakeSockAddr, 821  
 OsclValidInetAddr, 821  
 OsclWriteFD, 821  
 TIpmReq, 821  
 TOsclFileOffset, 821  
 TOsclHostent, 821  
 TOsclSockAddr, 821  
 TOsclSockAddrLen, 821  
 TOsclSocket, 821  
 osclconfig\_io\_check.h, 822  
     \_\_verify\_\_TOsclFileOffset\_\_defined\_\_,  
         822  
 osclconfig\_ix86.h, 823  
 osclconfig\_lib.h, 824  
 OSCL\_HAS\_RUNTIME\_LIB\_-  
     LOADING\_SUPPORT, 824  
 OSCL\_LIB\_READ\_DEBUG\_LIBS, 824  
 PV\_DYNAMIC\_LOADING\_CONFIG\_-  
     FILE\_PATH, 824  
 PV\_RUNTIME\_LIB\_FILENAME\_-  
     EXTENSION, 824  
 osclconfig\_lib\_check.h, 825  
 osclconfig\_limits\_typedefs.h, 826  
     OSCL\_CHAR\_IS\_SIGNED, 826  
     OSCL\_CHAR\_IS\_UNSIGNED, 826  
 osclconfig\_memory.h, 827  
     OSCL\_BYPASS\_MEMMGT, 827  
     OSCL\_HAS\_GLOBAL\_NEW\_DELETE,  
         827  
     OSCL\_HAS\_HEAP\_BASE\_SUPPORT,  
         827  
     OSCL\_HAS\_SYMBIAN\_MEMORY\_-  
         FUNCS, 827  
     PVMEM\_INST\_LEVEL, 827  
 osclconfig\_memory\_check.h, 828  
 osclconfig\_no\_os.h, 829  
 osclconfig\_proc.h, 830  
 osclconfig\_proc\_check.h, 831  
     \_\_verify\_\_TOsclConditionObject\_\_-  
         defined\_\_, 831  
     \_\_verify\_\_TOsclMutexObject\_\_defined\_\_,  
         831  
     \_\_verify\_\_TOsclSemaphoreObject\_\_-  
         defined\_\_, 831  
     \_\_verify\_\_TOsclThreadFuncArg\_\_-  
         defined\_\_, 831  
     \_\_verify\_\_TOsclThreadFuncRet\_\_-  
         defined\_\_, 831  
     \_\_verify\_\_TOsclThreadId\_\_defined\_\_, 831  
     \_\_verify\_\_TOsclThreadObject\_\_defined\_\_  
         , 831  
 osclconfig\_proc\_unix\_android.h, 833  
     OSCL\_HAS\_NON\_PREEMPTIVE\_-  
         THREAD\_SUPPORT, 834  
     OSCL\_HAS\_PTHREAD\_SUPPORT, 834  
     OSCL\_HAS\_SEM\_TIMEDWAIT\_-  
         SUPPORT, 834  
     OSCL\_HAS\_SYMBIAN\_SCHEDULER,  
         834  
     OSCL\_HAS\_THREAD\_SUPPORT, 834  
     OSCL\_THREAD\_DECL, 834  
 TOsclConditionObject, 834  
 TOsclMutexObject, 834  
 TOsclSemaphoreObject, 834  
 TOsclThreadFuncArg, 834  
 TOsclThreadFuncRet, 834  
 TOsclThreadId, 834  
 TOsclThreadObject, 834

osclconfig\_proc\_unix\_common.h, 835  
   OSCL\_HAS\_NON\_PREEMPTIVE\_-  
     THREAD\_SUPPORT, 836  
   OSCL\_HAS\_PTHREAD\_SUPPORT, 836  
   OSCL\_HAS\_SEM\_TIMEDWAIT\_-  
     SUPPORT, 836  
   OSCL\_HAS\_SYMBIAN\_SCHEDULER,  
     836  
   OSCL\_HAS\_THREAD\_SUPPORT, 836  
   OSCL\_THREAD\_DECL, 836  
   TOsclConditionObject, 836  
   TOsclMutexObject, 836  
   TOsclSemaphoreObject, 836  
   TOsclThreadFuncArg, 836  
   TOsclThreadFuncRet, 836  
   TOsclThreadId, 836  
   TOsclThreadObject, 836  
 osclconfig\_time.h, 837  
   OSCL\_HAS\_UNIX\_TIME\_FUNCS, 837  
   OsclBasicDateStruct, 837  
   OsclBasicTimeStruct, 837  
 osclconfig\_time\_check.h, 838  
   \_\_Validate\_\_BasicTimeDateStruct\_\_, 838  
   \_\_Validate\_\_BasicTimeStruct\_\_, 838  
 osclconfig\_unix\_android.h, 839  
   \_STRLIT, 842  
   \_STRLIT\_CHAR, 842  
   \_STRLIT\_WCHAR, 842  
   INT64, 842  
   INT64\_HILO, 842  
   OSCL\_DISABLE\_INLINES, 842  
   OSCL\_HAS\_ANSI\_MATH\_SUPPORT,  
     842  
   OSCL\_HAS\_ANSI\_STDIO\_SUPPORT,  
     842  
   OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT,  
     842  
   OSCL\_HAS\_ANSI\_STRING\_SUPPORT,  
     842  
   OSCL\_HAS\_ANSI\_WIDE\_STRING\_-  
     SUPPORT, 842  
   OSCL\_HAS\_BASIC\_LOCK, 842  
   OSCL\_HAS\_GLOBAL\_VARIABLE\_-  
     SUPPORT, 842  
   OSCL\_HAS\_IPHONE\_SUPPORT, 842  
   OSCL\_HAS\_MSWIN\_PARTIAL\_-  
     SUPPORT, 842  
   OSCL\_HAS\_MSWIN\_SUPPORT, 842  
   OSCL\_HAS\_SYMBIAN\_SUPPORT, 842  
   OSCL\_HAS\_TLS\_SUPPORT, 842  
   OSCL\_HAS\_UNICODE\_SUPPORT, 842  
   OSCL\_HAS\_UNIX\_SUPPORT, 842  
   OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN,  
     842  
   OSCL\_NATIVE\_INT64\_TYPE, 842  
   OSCL\_NATIVE\_UINT64\_TYPE, 842  
   OSCL\_NATIVE\_WCHAR\_TYPE, 842  
   OSCL\_TLS\_GET\_FUNC, 842  
   OSCL\_TLS\_IS\_KEYED, 842  
   OSCL\_TLS\_KEY\_CREATE\_FUNC, 842  
   OSCL\_TLS\_KEY\_DELETE\_FUNC, 842  
   OSCL\_TLS\_STORE\_FUNC, 842  
   TOsclBasicLockObject, 842  
   TOsclTlsKey, 842  
   UINT64, 842  
   UINT64\_HILO, 842  
 osclconfig\_unix\_common.h, 843  
   \_STRLIT, 846  
   \_STRLIT\_CHAR, 846  
   \_STRLIT\_WCHAR, 846  
   INT64, 846  
   INT64\_HILO, 846  
   OSCL\_DISABLE\_INLINES, 846  
   OSCL\_HAS\_ANSI\_MATH\_SUPPORT,  
     846  
   OSCL\_HAS\_ANSI\_STDIO\_SUPPORT,  
     846  
   OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT,  
     846  
   OSCL\_HAS\_ANSI\_STRING\_SUPPORT,  
     846  
   OSCL\_HAS\_ANSI\_WIDE\_STRING\_-  
     SUPPORT, 846  
   OSCL\_HAS\_BASIC\_LOCK, 846  
   OSCL\_HAS\_GLOBAL\_VARIABLE\_-  
     SUPPORT, 846  
   OSCL\_HAS\_MSWIN\_PARTIAL\_-  
     SUPPORT, 846  
   OSCL\_HAS\_MSWIN\_SUPPORT, 846  
   OSCL\_HAS\_SYMBIAN\_SUPPORT, 846  
   OSCL\_HAS\_TLS\_SUPPORT, 846  
   OSCL\_HAS\_UNICODE\_SUPPORT, 846  
   OSCL\_HAS\_UNIX\_SUPPORT, 846  
   OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN,  
     846  
   OSCL\_NATIVE\_INT64\_TYPE, 846  
   OSCL\_NATIVE\_UINT64\_TYPE, 846  
   OSCL\_NATIVE\_WCHAR\_TYPE, 846  
   OSCL\_TLS\_GET\_FUNC, 846  
   OSCL\_TLS\_IS\_KEYED, 846  
   OSCL\_TLS\_KEY\_CREATE\_FUNC, 846  
   OSCL\_TLS\_KEY\_DELETE\_FUNC, 846  
   OSCL\_TLS\_STORE\_FUNC, 846  
   TOsclBasicLockObject, 846  
   TOsclTlsKey, 846  
   UINT64, 846  
   UINT64\_HILO, 846  
 osclconfig\_util.h, 847

**OSCL\_CLOCK\_HAS\_DRIFT\_-CORRECTION**, [847](#)  
**OSCL\_HAS\_SYMBIAN\_MATH**, [847](#)  
**OSCL\_HAS\_SYMBIAN\_TIMERS**, [847](#)  
**OSCL\_RAND\_MAX**, [847](#)  
**SLEEP\_ONE\_SEC**, [847](#)  
**osclconfig\_util\_check.h**, [848](#)  
**OsclConnect**  
*osclconfig\_io.h*, [816](#)  
**OsclConnectComplete**  
*osclconfig\_io.h*, [816](#)  
**OsclConnectMethod**, [348](#)  
**OsclConnectMethod**  
*~OsclConnectMethod*, [348](#)  
*Connect*, [348](#)  
*ConnectRequest*, [348](#)  
*NewL*, [348](#)  
**OsclConnectRequest**, [349](#)  
*OsclConnectRequest*, [349](#)  
*OsclSocketI*, [539](#)  
**OsclConnectRequest**  
*Connect*, [349](#)  
*OsclConnectRequest*, [349](#)  
**OsclDestructDealloc**, [350](#)  
**OsclDestructDealloc**  
*~OsclDestructDealloc*, [350](#)  
*destruct\_and\_dealloc*, [350](#)  
**OsclDNS**, [351](#)  
*OsclSocketServ*, [555](#)  
**OsclDNS**  
*~OsclDNS*, [351](#)  
*CancelGetHostByName*, [351](#)  
*GetHostByName*, [352](#)  
*NewL*, [352](#)  
*OsclDNSRequestAO*, [352](#)  
**OsclDNSI**, [353](#)  
*OsclDNSRequestAO*, [365](#)  
*OsclSocketServI*, [557](#)  
**OsclDNSI**  
*~OsclDNSI*, [353](#)  
*Close*, [353](#)  
*DNSRequestParam*, [354](#)  
*GetHostByName*, [353](#)  
*GetHostByNameResponseContainsAlias-Info*, [354](#)  
*GetHostByNameSuccess*, [354](#)  
*GetNextHost*, [354](#)  
*GetNextHostSuccess*, [354](#)  
*NewL*, [354](#)  
*Open*, [354](#)  
*OsclDNSRequest*, [354](#)  
*OsclGetHostByNameRequest*, [354](#)  
**OsclDNSIBase**, [355](#)  
*OsclDNSIBase*, [356](#)

**OsclDNSIBase**  
*~OsclDNSIBase*, [356](#)  
*CancelFxn*, [356](#)  
*CancelGetHostByName*, [356](#)  
*Close*, [356](#)  
*GetHostByName*, [356](#)  
*GetHostByNameResponseContainsAlias-Info*, [356](#)  
*GetHostByNameSuccess*, [356](#)  
*GetNextHost*, [356](#)  
*GetNextHostSuccess*, [356](#)  
*iAlloc*, [357](#)  
*iSocketServ*, [357](#)  
*IsReady*, [356](#)  
*Open*, [356](#)  
*OsclDNSIBase*, [356](#)  
*OsclDNSRequest*, [357](#)  
*OsclGetHostByNameRequest*, [357](#)  
**OsclDNSMethod**, [358](#)  
*OsclDNSMethod*, [359](#)  
*OsclDNSRequestAO*, [365](#)  
**OsclDNSMethod**  
*Abort*, [359](#)  
*AbortAll*, [359](#)  
*CancelMethod*, [359](#)  
*ConstructL*, [359](#)  
*iAlloc*, [360](#)  
*iDNFSfxn*, [360](#)  
*iDNSObserver*, [360](#)  
*iDNSRequestAO*, [360](#)  
*iId*, [360](#)  
*iLogger*, [360](#)  
*MethodDone*, [359](#)  
*OsclDNSMethod*, [359](#)  
*Run*, [359](#)  
*StartMethod*, [359](#)  
**OsclDNSObserver**, [361](#)  
**OsclDNSObserver**  
*~OsclDNSObserver*, [361](#)  
*HandleDNSEvent*, [361](#)  
**OsclDNSRequest**, [362](#)  
*OsclDNSI*, [354](#)  
*OsclDNSIBase*, [357](#)  
*OsclDNSRequest*, [362](#)  
*OsclDNSRequestAO*, [365](#)  
**OsclDNSRequest**  
*~OsclDNSRequest*, [362](#)  
*Activate*, [362](#)  
*CancelRequest*, [362](#)  
*Complete*, [362](#)  
*iActive*, [362](#)  
*iDNSRequestAO*, [362](#)  
*iDNSRequestParam*, [362](#)  
*OsclDNSRequest*, [362](#)

OsclDNSRequestAO, 363  
   OsclDNS, 352  
   OsclDNSRequestAO, 364  
 OsclDNSRequestAO  
   Abort, 364  
   Cancelled, 364  
   ConstructL, 364  
   DoCancel, 364  
   Failure, 364  
   GetHostNameParam, 365  
   GetSocketError, 364  
   iDNSI, 365  
   iDNSMethod, 365  
   iLogger, 365  
   iSocketError, 365  
   NewRequest, 364  
   OsclDNSI, 365  
   OsclDNSMethod, 365  
   OsclDNSRequest, 365  
   OsclDNSRequestAO, 364  
   RequestDone, 364  
   Run, 364  
   Serv, 365  
   Success, 365  
 OsclDoubleLink, 366  
   OsclDoubleLink, 366  
 OsclDoubleLink  
   iNext, 366  
   InsertAfter, 366  
   InsertBefore, 366  
   iPrev, 366  
   OsclDoubleLink, 366  
   Remove, 366  
 OsclDoubleList, 367  
   OsclDoubleList, 367  
 OsclDoubleList  
   Head, 367  
   InsertHead, 367  
   InsertTail, 367  
   IsHead, 367  
   IsTail, 367  
   OsclDoubleList, 367  
   Tail, 367  
 OsclDoubleListBase, 368  
   OsclDoubleListBase, 369  
 OsclDoubleListBase  
   getHead, 369  
   getOffset, 369  
   iHead, 369  
   Insert, 369  
   InsertHead, 369  
   InsertTail, 369  
   iOffset, 369  
   IsEmpty, 369  
   OsclDoubleListBase, 369  
   Reset, 369  
   SetOffset, 369  
 OsclDoubleRunner, 370  
   OsclDoubleRunner, 370  
 OsclDoubleRunner  
   iHead, 370  
   iNext, 370  
   iOffset, 370  
   operator T \*, 370  
   operator++, 370  
   operator--, 370  
   OsclDoubleRunner, 370  
   Set, 370  
   SetToHead, 370  
   SetToTail, 370  
 OsclErrAlreadyExists  
   osclerror, 92  
 OsclErrAlreadyInstalled  
   osclerror, 92  
 OsclErrArgument  
   osclerror, 92  
 OsclErrBadHandle  
   osclerror, 92  
 OsclErrBusy  
   osclerror, 92  
 OsclErrCancelled  
   osclerror, 92  
 OsclErrCorrupt  
   osclerror, 92  
 OsclErrGeneral  
   osclerror, 92  
 OsclErrInvalidState  
   osclerror, 92  
 OsclErrNoHandler  
   osclerror, 92  
 OsclErrNoMemory  
   osclerror, 92  
 OsclErrNone  
   osclerror, 92  
 OsclErrNoResources  
   osclerror, 92  
 OsclErrNotInstalled  
   osclerror, 92  
 OsclErrNotReady  
   osclerror, 92  
 OsclErrNotSupported  
   osclerror, 92  
 OsclError, 372  
   OsclErrorTrapImp, 378  
   OsclExecSchedulerCommonBase, 397  
   OsclTrapStack, 599  
 OsclError  
   Leave, 372

LeaveIfError, 372  
 LeaveIfNull, 372  
 Pop, 372  
 PopDealloc, 372, 373  
 PushL, 373  
**osclerror**  
   \_PV\_TRAP, 88  
   \_PV\_TRAP\_NO\_TLS, 88  
   internalLeave, 88  
   OSCL\_BAD\_ALLOC\_EXCEPTION\_CODE, 88  
   OSCL\_CATCH, 88  
   OSCL\_CATCH\_ANY, 88  
   OSCL\_ERR\_NONE, 89  
   OSCL\_FIRST\_CATCH, 89  
   OSCL\_FIRST\_CATCH\_ANY, 89  
   OSCL\_GetLastError, 93  
   OSCL\_IsErrnoSupported, 93  
   OSCL\_JUMP\_MAX\_JUMP\_MARKS, 89  
   OSCL\_LAST\_CATCH, 89  
   OSCL\_LEAVE, 89  
   OSCL\_MAX\_TRAP\_LEVELS, 90  
   OSCL\_SetLastError, 93  
   OSCL\_StrError, 93  
   OSCL\_TRAPSTACK\_POP, 90  
   OSCL\_TRAPSTACK\_POPDEALLOC, 90  
   OSCL\_TRAPSTACK\_PUSH, 90  
   OSCL\_TRY, 90  
   OSCL\_TRY\_NO\_TLS, 90  
   OsclErrAlreadyExists, 92  
   OsclErrAlreadyInstalled, 92  
   OsclErrArgument, 92  
   OsclErrBadHandle, 92  
   OsclErrBusy, 92  
   OsclErrCancelled, 92  
   OsclErrCorrupt, 92  
   OsclErrGeneral, 92  
   OsclErrInvalidState, 92  
   OsclErrNoHandler, 92  
   OsclErrNoMemory, 92  
   OsclErrNone, 92  
   OsclErrNoResources, 92  
   OsclErrNotInstalled, 92  
   OsclErrNotReady, 92  
   OsclErrNotSupported, 92  
   OsclErrOverflow, 92  
   OsclErrSystemCallFailed, 92  
   OsclErrThreadContextIncorrect, 92  
   OsclErrTimeout, 92  
   OsclErrUnderflow, 92  
   OsclFailure, 92  
   OsclLeaveCode, 93  
   OsclPending, 92  
   OsclReturnCode, 93  
  
 OsclSuccess, 92  
 OsclTrapOperation, 93  
 PVError\_DoLeave, 92  
 PVERROR\_IMP\_JUMPS, 92  
 PVERRORTRAP\_REGISTRY, 92  
 PVERRORTRAP\_REGISTRY\_ID, 93  
 OsclErrorAllocator, 374  
   OsclErrorAllocator, 374  
 OsclErrorAllocator  
   allocate, 374  
   deallocate, 374  
   operator delete, 375  
   operator new, 375  
   OsclErrorAllocator, 374  
 OsclErrorTrap, 376  
   OsclErrorTrapImp, 378  
   OsclTrapStack, 599  
 OsclErrorTrap  
   Cleanup, 376  
   GetErrorTrapImp, 376  
   Init, 376  
 OsclErrorTrapImp, 377  
   OsclJump, 421  
   OsclTrapStack, 599  
 OsclErrorTrapImp  
   CPVInterfaceProxy, 378  
   iJumpData, 378  
   iLeave, 378  
   iTrapStack, 378  
   OsclError, 378  
   OsclErrorTrap, 378  
   OsclExecScheduler, 378  
   OsclExecSchedulerCommonBase, 378  
   OsclJump, 378  
   OsclJumpMark, 378  
   OsclScheduler, 378  
   OsclTrapStack, 378  
   Trap, 377  
   TrapNoTls, 377  
   UnTrap, 377  
 OsclErrOverflow  
   osclerror, 92  
 OsclErrSystemCallFailed  
   osclerror, 92  
 OsclErrThreadContextIncorrect  
   osclerror, 92  
 OsclErrTimeout  
   osclerror, 92  
 OsclErrUnderflow  
   osclerror, 92  
 OsclException, 379  
   OsclException, 379  
 OsclException  
   getLeaveCode, 379



IsInstalled, 396  
 IsStarted, 396  
 iStopper, 399  
 iStopperCrit, 399  
 iSuspended, 399  
 iThreadContext, 399  
 iTime, 399  
 iTimeCompareThreshold, 399  
 iTotPercent, 399  
 iTotTicksTemp, 399  
 OsclActiveObject, 397  
 OsclCoeActiveScheduler, 397  
 OsclError, 397  
 OsclExecScheduler, 397  
 OsclExecSchedulerCommonBase, 395  
 OsclReadyQ, 397  
 OsclScheduler, 397  
 OsclTimerCompare, 397  
 OsclTimerObject, 399  
 PendComplete, 396  
 PVActiveBase, 399  
 PVActiveStats, 399  
 PVSchedulerStopper, 399  
 PVThreadContext, 399  
 RequestCanceled, 396  
 ResetLogPerf, 396  
 ResumeScheduler, 396  
 SetScheduler, 396  
 ShowStats, 396  
 ShowSummaryStats, 396  
 StartNativeScheduler, 396  
 StartScheduler, 396  
 StopScheduler, 396  
 SuspendScheduler, 397  
 TOtherExecStats, 394  
 UninstallScheduler, 397  
 UpdateTimers, 397  
 UpdateTimersMsec, 397  
 WaitForReadyAO, 397  
 OsclExtractFilenameFromFullPath  
   OsclFileManager, 407  
 OsclFailure  
   osclerror, 92  
 OsclFileCache, 401  
   Oscl\_File, 186  
   OsclFileCache, 402  
 OsclFileCache  
   ~OsclFileCache, 402  
   \_fixedCaches, 402  
   \_movableCache, 402  
   AddFixedCache, 402  
   Close, 402  
   EndOfFile, 402  
   FileSize, 402  
   Flush, 402  
   Open, 402  
   OsclFileCache, 402  
   OsclFileCacheBuffer, 402  
   Read, 402  
   Seek, 402  
   Tell, 402  
   Write, 402  
   OsclFileCacheBuffer, 403  
     Oscl\_File, 186  
     OsclFileCache, 402  
     OsclFileCacheBuffer, 404  
 OsclFileCacheBuffer  
   capacity, 404  
   Contains, 404  
   currentPos, 404  
   endPos, 404  
   filePosition, 404  
   FillFromFile, 404  
   iContainer, 404  
   isFixed, 404  
   IsUpdated, 404  
   OsclFileCacheBuffer, 404  
   pBuffer, 404  
   Preceeds, 404  
   PrepRead, 404  
   PrepWrite, 404  
   SetPosition, 404  
   updateEnd, 404  
   updateStart, 404  
   usableSize, 404  
   WriteUpdatesToFile, 404  
 OsclFileHandle, 405  
   OsclFileHandle, 405  
 OsclFileHandle  
   Handle, 405  
   Oscl\_File, 405  
   OsclFileHandle, 405  
 OsclFileManager, 406  
   OSCL\_FILE\_ATTRIBUTE\_ARCHIVE,  
     406  
   OSCL\_FILE\_ATTRIBUTE\_-  
      DIRECTORY, 406  
   OSCL\_FILE\_ATTRIBUTE\_HIDDEN, 406  
   OSCL\_FILE\_ATTRIBUTE\_NORMAL,  
     406  
   OSCL\_FILE\_ATTRIBUTE\_READONLY,  
     406  
   OSCL\_FILE\_ATTRIBUTE\_SYSTEM,  
     406  
 OsclFileManager  
   OSCL\_FILE\_ATTRIBUTE\_TYPE, 406  
   OsclExtractFilenameFromFullPath, 407  
   OsclGetFileAttributes, 407

OsclGetFileCreationTime, 407, 408  
 OsclGetFileLastAccessTime, 408  
 OsclGetFileLastWriteTime, 409  
 OsclGetFileSize, 409  
 OsclFileStats, 411  
   OsclFileStats, 411  
 OsclFileStats  
   End, 411  
   Log, 411  
   LogAll, 411  
   OsclFileStats, 411  
   Start, 411  
 OsclFileStatsItem, 412  
 OsclFileStatsItem  
   iOpCount, 412  
   iParam, 412  
   iParam2, 412  
   iStartTick, 412  
   iTTotalTicks, 412  
 OsclFloat  
   osclbase, 35  
 OsclGetAsyncSockErr  
   osclconfig\_io.h, 816  
 OsclGetDottedAddr  
   osclconfig\_io.h, 816  
 OsclGetDottedAddrVector  
   osclconfig\_io.h, 817  
 OsclGetFileAttributes  
   OsclFileManager, 407  
 OsclGetFileCreationTime  
   OsclFileManager, 407, 408  
 OsclGetFileLastAccessTime  
   OsclFileManager, 408  
 OsclGetFileLastWriteTime  
   OsclFileManager, 409  
 OsclGetFileSize  
   OsclFileManager, 409  
 OsclGetHostbyname  
   osclconfig\_io.h, 817  
 OsclGetHostNameMethod, 413  
   OsclGetHostNameRequest, 414  
 OsclGetHostNameMethod  
   ~OsclGetHostNameMethod, 413  
   GetHostName, 413  
   NewL, 413  
 OsclGetHostNameRequest, 414  
   OsclIDNSI, 354  
   OsclDNSIBase, 357  
 OsclGetHostNameRequest  
   OsclGetHostNameMethod, 414  
 OsclGetPeerName  
   osclconfig\_io.h, 817  
 OsclInit, 415  
 OsclInit  
   Cleanup, 415  
   Init, 415  
 OsclInteger64Transport, 416  
 OsclInteger64Transport  
   iHigh, 416  
   iLow, 416  
 osclio  
   EOsclFileOp\_Close, 98  
   EOsclFileOp\_EndOfFile, 98  
   EOsclFileOp\_Flush, 98  
   EOsclFileOp\_Last, 99  
   EOsclFileOp\_NativeClose, 98  
   EOsclFileOp\_NativeEndOfFile, 99  
   EOsclFileOp\_NativeFlush, 99  
   EOsclFileOp\_NativeOpen, 98  
   EOsclFileOp\_NativeRead, 98  
   EOsclFileOp\_NativeSeek, 99  
   EOsclFileOp\_NativeSetSize, 99  
   EOsclFileOp\_NativeSize, 99  
   EOsclFileOp\_NativeTell, 99  
   EOsclFileOp\_NativeWrite, 99  
   EOsclFileOp\_Open, 98  
   EOsclFileOp\_Read, 98  
   EOsclFileOp\_Seek, 98  
   EOsclFileOp\_SetSize, 98  
   EOsclFileOp\_Size, 98  
   EOsclFileOp\_Tell, 98  
   EOsclFileOp\_Write, 98  
   EPVDNSCancel, 99  
   EPVDNSFailure, 99  
   EPVDNSGetHostByName, 99  
   EPVDNSPending, 99  
   EPVDNSSuccess, 99  
   EPVDNSTimeout, 99  
   oscl\_chdir, 99  
   OSCL\_FILE\_CHAR\_PATH\_-  
     DELIMITER, 97  
   OSCL\_FILE\_STATS\_LOGGER\_NODE,  
     97  
   OSCL\_FILE\_WCHAR\_PATH\_-  
     DELIMITER, 97  
   OSCL\_FILEMGMT\_E\_ALREADY\_-  
     EXISTS, 98  
   OSCL\_FILEMGMT\_E\_NO\_MATCH, 98  
   OSCL\_FILEMGMT\_E\_NOT\_EMPTY, 98  
   OSCL\_FILEMGMT\_E\_NOT\_-  
     IMPLEMENTED, 98  
   OSCL\_FILEMGMT\_E\_OK, 98  
   OSCL\_FILEMGMT\_E\_PATH\_NOT\_-  
     FOUND, 98  
   OSCL\_FILEMGMT\_E\_PATH\_TOO\_-  
     LONG, 98  
   OSCL\_FILEMGMT\_E\_PERMISSION\_-  
     DENIED, 98

OSCL\_FILEMGMT\_E\_SYS\_SPECIFIC,  
   98  
 OSCL\_FILEMGMT\_E\_UNKNOWN, 98  
 OSCL\_FILEMGMT\_ERR\_TYPE, 98  
 OSCL\_FILEMGMT\_MODE\_DIR, 98  
 OSCL\_FILEMGMT\_MODES, 98  
 OSCL\_FILEMGMT\_PERMS, 98  
 OSCL\_FILEMGMT\_PERMS\_EXECUTE,  
   98  
 OSCL\_FILEMGMT\_PERMS\_READ, 98  
 OSCL\_FILEMGMT\_PERMS\_WRITE, 98  
 OSCL\_FSSTAT, 97  
 oscl\_getcwd, 99, 100  
 OSCL\_IO\_EXTENSION\_MAXLEN, 97  
 OSCL\_IO\_FILENAME\_MAXLEN, 97  
 oscl\_mkdir, 100  
 oscl\_rename, 100, 101  
 oscl\_rmdir, 101  
 oscl\_stat, 101, 102  
 OSCL\_STAT\_BUF, 97  
 oscl\_statfs, 102  
 TOsclFileHandle, 97  
 TOsclFileOffsetInt32, 97  
 TOsclFileOp, 98  
 TPVDNSEvent, 99  
 TPVDNSFxn, 99  
 OsclIpMReq, 417  
   OsclIpMReq, 417  
 OsclIpMReq  
   interfaceAddr, 417  
   multicastAddr, 417  
   OsclIpMReq, 417  
 OsclIPSocketI, 418  
   OsclIPSocketI, 419  
 OsclIPSocketI  
   ~OsclIPSocketI, 419  
   Alloc, 419  
   Bind, 419  
   Close, 419  
   ConstructL, 419  
   GetPeerName, 419  
   GetRecvData, 419  
   GetSendData, 419  
   iAddress, 420  
   iAlloc, 420  
   iId, 420  
   iLogger, 420  
   iObserver, 420  
   iSocket, 420  
   iSocketServ, 420  
   Join, 419  
   OsclIPSocketI, 419  
   OsclSocketMethod, 420  
   OsclSocketRequestAO, 420  
     SetOptionToReuseAddress, 419  
     SetRecvBufferSize, 419  
     SetTOS, 419  
     SocketServ, 419  
     ThreadLogoff, 419  
     ThreadLogon, 419  
 OsclJoin  
   osclconfig\_io.h, 817  
 OsclJump, 421  
   OsclErrorTrapImp, 378  
 OsclJump  
   ~OsclJump, 421  
   Jump, 421  
   OsclErrorTrapImp, 421  
   StaticJump, 421  
   Top, 421  
 OsclJumpMark  
   OsclErrorTrapImp, 378  
 OsclLeaveCode  
   osclerror, 93  
 OsclListen  
   osclconfig\_io.h, 818  
 OsclListenMethod, 422  
 OsclListenMethod  
   ~OsclListenMethod, 422  
   Listen, 422  
   ListenRequest, 422  
   NewL, 422  
 OsclListenRequest, 423  
   OsclListenRequest, 423  
 OsclListenRequest  
   Listen, 423  
   OsclListenRequest, 423  
 OsclLockBase, 424  
 OsclLockBase  
   ~OsclLockBase, 424  
   Lock, 424  
   Unlock, 424  
 OsclMakeInAddr  
   osclconfig\_io.h, 818  
 OsclMakeSockAddr  
   osclconfig\_io.h, 818  
 OsclMem, 425  
   OsclMemGlobalAuditObject, 441  
 OsclMem  
   Cleanup, 425  
   Init, 425  
 OsclMemAllocator, 426  
 OsclMemAllocator  
   allocate, 426  
   allocate\_fl, 426  
   deallocate, 426  
 OsclMemAllocDestructDealloc, 427  
 OsclMemAllocDestructDealloc

allocate, 427  
 allocate\_fl, 427  
 deallocate, 427  
 destruct\_and\_dealloc, 427  
**OsclMemAudit**, 429  
     **OsclMemAudit**, 429  
**OsclMemAudit**  
     ~**OsclMemAudit**, 429  
     GetLock, 430  
     MM\_AddTag, 430  
     MM\_allocate, 430  
     MM\_CreateAllocNodeInfo, 430  
     MM\_deallocate, 430  
     MM\_GetAllocNo, 430  
     MM\_GetAllocNodeInfo, 430  
     MM\_GetExistingTag, 431  
     MM\_GetMode, 431  
     MM\_GetNumAllocNodes, 431  
     MM\_GetOverheadStats, 431  
     MM\_GetPostfillPattern, 431  
     MM\_GetPrefillPattern, 431  
     MM\_GetRefCount, 431  
     MM\_GetRootNode, 432  
     MM\_GetStats, 432  
     MM\_GetStatsInDepth, 432  
     MM\_GetTagName, 432  
     MM\_GetTreeNodes, 432  
     MM\_ReleaseAllocNodeInfo, 432  
     MM\_SetFailurePoint, 432  
     MM\_SetMode, 433  
     MM\_SetPostfillPattern, 433  
     MM\_SetPrefillPattern, 433  
     MM\_SetTagLevel, 433  
     MM\_UnsetFailurePoint, 433  
     MM\_Validate, 433  
     **OsclMemAudit**, 429  
     **OsclMemGlobalAuditObject**, 434  
**OSCLMemAutoPtr**, 435  
     **OSCLMemAutoPtr**, 436  
**OSCLMemAutoPtr**  
     ~**OSCLMemAutoPtr**, 436  
     \_Ownership, 438  
     allocate, 437  
     deallocate, 437  
     get, 437  
     operator \*, 437  
     operator->, 437  
     operator=, 437  
     **OSCLMemAutoPtr**, 436  
     release, 437  
     setWithoutOwnership, 437  
     takeOwnership, 438  
**OsclMemBasicAllocator**, 439  
**OsclMemBasicAllocator**  
     allocate, 439  
     deallocate, 439  
**OsclMemBasicAllocDestructDealloc**, 440  
**OsclMemBasicAllocDestructDealloc**  
     allocate, 440  
     deallocate, 440  
     destruct\_and\_dealloc, 440  
**OsclMemGlobalAuditObject**, 441  
     **OsclMemAudit**, 434  
**OsclMemGlobalAuditObject**  
     audit\_type, 441  
     getGlobalMemAuditObject, 441  
     **OsclMem**, 441  
**OsclMemInit**  
     osclmemory, 62  
**osclmemory**  
     \_OSCL\_CLEANUP\_BASE\_CLASS, 50  
     \_OSCL\_TRAP\_NEW, 50  
     \_oscl\_audit\_calloc, 59  
     \_oscl\_audit\_free, 59  
     \_oscl\_audit\_malloc, 59  
     \_oscl\_audit\_new, 59  
     \_oscl\_audit\_realloc, 60  
     \_oscl\_calloc, 60  
     \_oscl\_default\_audit\_calloc, 60  
     \_oscl\_default\_audit\_malloc, 60  
     \_oscl\_default\_audit\_new, 60  
     \_oscl\_default\_audit\_realloc, 60  
     \_oscl\_free, 60  
     \_oscl\_malloc, 60  
     \_oscl\_realloc, 60  
     ALLOC\_NODE\_FLAG, 62  
     COMPUTE\_MEM\_ALIGN\_SIZE, 51  
     DEFAULT\_MM\_AUDIT\_MODE, 52  
     DEFAULT\_POSTFILL\_PATTERN, 52  
     DEFAULT\_PREFILL\_PATTERN, 52  
     FENCE\_PATTERN, 52  
     MEM\_ALIGN\_SIZE, 52  
     MIN\_FENCE\_SIZE, 52  
     MM\_ALLOC\_MAX\_QUERY\_-  
         FILENAME\_LEN, 52  
     MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN,  
         52  
     MM\_AllocNodeAutoPtr, 59  
     MM\_AUDIT\_ALLOC\_NODE\_-  
         ENABLE\_FLAG, 52  
     MM\_AUDIT\_ALLOC\_NODE\_-  
         SUPPORT, 52  
     MM\_AUDIT\_FAILURE\_SIMULATION\_-  
         SUPPORT, 52  
     MM\_AUDIT\_FENCE\_SUPPORT, 52  
     MM\_AUDIT\_FILL\_SUPPORT, 52  
     MM\_AUDIT\_INCLUDE\_ALL\_HEAP\_-  
         VALIDATION, 52

MM\_AUDIT\_POSTFILL\_FLAG, 52  
 MM\_AUDIT\_PREFILL\_FLAG, 52  
 MM\_AUDIT\_SUPPRESS\_FILENAME\_-  
   FLAG, 52  
 MM\_AUDIT\_VALIDATE\_ALL\_HEAP\_-  
   FLAG, 52  
 MM\_AUDIT\_VALIDATE\_BLOCK, 52  
 MM\_AUDIT\_VALIDATE\_ON\_FREE\_-  
   FLAG, 52  
 MM\_StatsNodeTagTreeType, 59  
 MMAuditCharAutoPtr, 59  
 MMAuditUint8AutoPtr, 59  
 operator delete, 60  
 operator delete[], 60  
 operator new, 60  
 operator new[], 60  
 OSCL\_ALLOC\_DELETE, 52  
 OSCL\_ALLOC\_NEW, 53  
 OSCL\_ARRAY\_DELETE, 53  
 OSCL\_ARRAY\_NEW, 53  
 OSCL\_AUDIT\_ARRAY\_NEW, 53  
 OSCL\_AUDIT\_CALLOC, 54  
 OSCL\_AUDIT\_MALLOC, 54  
 OSCL\_AUDIT\_NEW, 54  
 OSCL\_AUDIT\_REALLOC, 55  
 OSCL\_CALLOC, 55  
 oscl\_calloc, 55  
 OSCL\_CLEANUP\_BASE\_CLASS, 55  
 OSCL\_DEFAULT\_FREE, 56  
 OSCL\_DEFAULT\_MALLOC, 56  
 OSCL\_DELETE, 56  
 OSCL\_DISABLE\_WARNING\_-  
   RETURN\_TYPE\_NOT\_UDT, 56  
 OSCL\_DISABLE\_WARNING\_-  
   TRUNCATE\_DEBUG\_MESSAGE,  
   56  
 OSCL\_FREE, 56  
 oscl\_free, 56  
 OSCL\_HAS\_GLOBAL\_NEW\_DELETE,  
   56  
 OSCL\_MALLOC, 57  
 oscl\_malloc, 57  
 oscl\_mem\_aligned\_size, 60  
 oscl\_memcmp, 61  
 oscl\_memcpy, 61  
 oscl\_memmove, 61  
 oscl\_memmove32, 61  
 oscl\_memset, 62  
 OSCL\_NEW, 57  
 OSCL\_PLACEMENT\_NEW, 57  
 OSCL\_REALLOC, 57  
 oscl\_realloc, 57  
 OSCL\_TRAP\_ALLOC\_NEW, 57  
 OSCL\_TRAP\_AUDIT\_NEW, 58  
 OSCL\_TRAP\_NEW, 58  
 OsclMemInit, 62  
 OsclMemStatsNodeAutoPtr, 59  
 OsclTagTreeType, 59  
 TagTree\_Allocator, 59  
 OsclMemoryFragment, 442  
 OsclMemoryFragment  
   len, 442  
   ptr, 442  
 OsclMemPoolFixedChunkAllocator, 443  
   OsclMemPoolFixedChunkAllocator, 444  
 OsclMemPoolFixedChunkAllocator  
   ~OsclMemPoolFixedChunkAllocator, 444  
   addRef, 444  
   allocate, 444  
   CancelFreeChunkAvailableCallback, 444  
   createmempool, 444  
   deallocate, 445  
   destroymempool, 445  
   enablenullpointerreturn, 445  
   iCheckNextAvailableFreeChunk, 446  
   iChunkAlignment, 446  
   iChunkSize, 446  
   iChunkSizeMemAligned, 446  
   iEnableNullPtrReturn, 446  
   iFreeMemChunkList, 446  
   iMemPool, 446  
   iMemPoolAligned, 446  
   iMemPoolAllocator, 446  
   iNextAvailableContextData, 446  
   iNumChunk, 446  
   iObserver, 446  
   iRefCount, 446  
   notifyfreechunkavailable, 445  
   OsclMemPoolFixedChunkAllocator, 444  
   removeRef, 445  
 OsclMemPoolFixedChunkAllocatorObserver,  
   447  
 OsclMemPoolFixedChunkAllocatorObserver  
   ~OsclMemPoolFixedChunkAllocatorObserver,  
   447  
   freechunkavailable, 447  
 OsclMemPoolResizableAllocator, 448  
   OsclMemPoolResizableAllocator, 449  
 OsclMemPoolResizableAllocator  
   ~OsclMemPoolResizableAllocator, 449  
   addnewmempoolbuffer, 449  
   addRef, 449  
   allocate, 450  
   allocateblock, 450  
   CancelFreeChunkAvailableCallback, 450  
   CancelFreeMemoryAvailableCallback, 450  
   deallocate, 450  
   deallocateblock, 450

destroyallmempoolbuffers, 450  
 enablenullpointerreturn, 450  
 findfreeblock, 451  
 getAllocatedSize, 451  
 getAvailableSize, 451  
 getBufferSize, 451  
 getLargestContiguousFreeBlockSize, 451  
 getMemPoolBufferAllocatedSize, 451  
 getMemPoolBufferSize, 451  
 iBlockInfoAlignedSize, 453  
 iBufferInfoAlignedSize, 453  
 iCheckFreeMemoryAvailable, 453  
 iCheckNextAvailable, 453  
 iEnableNullPtrReturn, 453  
 iExpectedNumBlocksPerBuffer, 453  
 iFreeMemContextData, 453  
 iFreeMemPoolObserver, 453  
 iMaxNewMemPoolBufferSz, 453  
 iMemPoolBufferAllocator, 453  
 iMemPoolBufferList, 453  
 iMemPoolBufferNumLimit, 453  
 iMemPoolBufferSize, 453  
 iNextAvailableContextData, 453  
 iObserver, 453  
 iRefCount, 453  
 iRequestedAvailableFreeMemSize, 453  
 iRequestedNextAvailableSize, 453  
 memoryPoolBufferMgmtOverhead, 451  
 notifyfreeblockavailable, 451  
 notifyfreememoryavailable, 451  
 OsclMemPoolResizableAllocator, 449  
 removeRef, 452  
 setMaxSzForNewMemPoolBuffer, 452  
 trim, 452  
 validateblock, 452  
**OsclMemPoolResizableAllocator::MemPoolBlockInfo**,  
 454  
**OsclMemPoolResizableAllocator::MemPool-**  
**BlockInfo**  
 iBlockBuffer, 454  
 iBlockPostFence, 454  
 iBlockPreFence, 454  
 iBlockSize, 454  
 iNextFreeBlock, 454  
 iParentBuffer, 454  
 iPrevFreeBlock, 454  
**OsclMemPoolResizableAllocator::MemPoolBufferInfo**,  
 455  
**OsclMemPoolResizableAllocator::MemPool-**  
**BufferInfo**  
 iAllocatedSz, 455  
 iBufferPostFence, 455  
 iBufferPreFence, 455  
 iBufferSize, 455  
 iEndAddr, 455  
 iNextFreeBlock, 455  
 iNumOutstanding, 455  
 iStartAddr, 455  
**OsclMemPoolResizableAllocatorMemoryObserver**,  
 456  
**OsclMemPoolResizableAllocatorMemory-**  
**Observer**  
 ~OsclMemPoolResizableAllocatorMemoryObserver,  
 456  
 freememoryavailable, 456  
**OsclMemPoolResizableAllocatorObserver**, 457  
**OsclMemPoolResizableAllocatorObserver**  
 ~OsclMemPoolResizableAllocatorObserver,  
 457  
 freeblockavailable, 457  
**OsclMemStatsNode**, 458  
 OsclMemStatsNode, 458  
**OsclMemStatsNode**  
 ~OsclMemStatsNode, 458  
 operator delete, 458  
 operator new, 458  
 OsclMemStatsNode, 458  
 pMMFIParam, 458  
 pMMStats, 458  
 reset, 458  
 tag, 458  
**OsclMemStatsNodeAutoPtr**  
 osclmemory, 59  
**OsclMutex**, 459  
 OsclMutex, 459  
**OsclMutex**  
 ~OsclMutex, 459  
 Close, 459  
 Create, 459  
 Lock, 460  
 OsclMutex, 459  
 TryLock, 460  
 Unlock, 460  
**OsclNameString**, 461  
 OsclNameString, 461  
**OsclNameString**  
 MaxLen, 461  
 OsclNameString, 461  
 Set, 461  
 Str, 461  
**OsclNativeFile**, 462  
**Oscl\_FileServer**, 194  
**OsclNativeFile**, 463  
**OsclNativeFile**  
 ~OsclNativeFile, 463  
 Close, 463  
 EndOfFile, 463  
 Flush, 463

GetError, 463  
 GetReadAsyncNumElements, 463  
 HasAsyncRead, 463  
 Mode, 463  
 Open, 463  
 OsclNativeFile, 463  
 Read, 463  
 ReadAsync, 463  
 ReadAsyncCancel, 463  
 Seek, 464  
 SetSize, 464  
 Size, 464  
 Tell, 464  
 Write, 464  
 OsclNativeFileParams, 465  
   OsclNativeFileParams, 465  
 OsclNativeFileParams  
   iAsyncReadBufferSize, 465  
   iNativeAccessMode, 465  
   iNativeBufferSize, 465  
   OsclNativeFileParams, 465  
 OsclNetworkAddress, 466  
   OsclNetworkAddress, 466  
 OsclNetworkAddress  
   ipAddr, 466  
   operator==, 466  
   OsclNetworkAddress, 466  
   port, 466  
 OsclNoYieldMutex  
   oscl\_mutex.h, 723  
 OsclNullLock, 467  
 OsclNullLock  
   ~OsclNullLock, 467  
   Lock, 467  
   Unlock, 467  
 OsclPending  
   osclerror, 92  
 OsclPipe  
   osclconfig\_io.h, 818  
 OsclPriorityLink, 468  
 OsclPriorityLink  
   iPriority, 468  
 OsclPriorityList, 469  
   OsclPriorityList, 469  
 OsclPriorityList  
   Head, 469  
   Insert, 469  
   IsHead, 469  
   IsTail, 469  
   OsclPriorityList, 469  
   Tail, 469  
 OsclPriorityQueue, 470  
   OsclPriorityQueue, 471  
 OsclPriorityQueue  
   ~OsclPriorityQueue, 471  
   c, 473  
   comp, 473  
   compare\_EQ, 471  
   compare\_LT, 471  
   const\_reference, 471  
   container\_type, 471  
   empty, 472  
   find\_heap, 472  
   iterator, 471  
   oscl\_pqueue\_test, 473  
   OsclPriorityQueue, 471  
   pop, 472  
   pop\_heap, 472  
   push, 472  
   push\_heap, 472  
   remove, 472  
   reserve, 472  
   size, 472  
   swap, 472  
   top, 472  
   validate, 473  
   value\_type, 471  
   vec, 473  
 OsclPriorityQueueBase, 474  
   Oscl\_Vector\_Base, 293  
 OsclPriorityQueueBase  
   ~OsclPriorityQueueBase, 474  
   construct, 474  
   find\_heap, 474  
   pop\_heap, 474  
   push\_heap, 474  
   remove, 474  
 osclproc  
   EPVThreadContext\_InThread, 106  
   EPVThreadContext\_NonOsclThread, 106  
   EPVThreadContext\_OsclThread, 106  
   EPVThreadContext\_Undetermined, 106  
   OSCL\_PERF\_SUMMARY\_LOGGING, 105  
   OSCL\_REQUEST\_ERR\_CANCEL, 106  
   OSCL\_REQUEST\_ERR\_GENERAL, 106  
   OSCL\_REQUEST\_ERR\_NONE, 106  
   OSCL\_REQUEST\_PENDING, 106  
   OSCL\_ZEROIZE, 105  
   OsclPtrAdd, 106  
   OsclPtrSub, 106  
   PV\_SCHED\_CHECK\_Q, 105  
   PV\_SCHED\_ENABLE\_AO\_STATS, 105  
   PV\_SCHED\_ENABLE\_LOOP\_STATS, 105  
   PV\_SCHED\_ENABLE\_PERF\_LOGGING, 105

---

PV\_SCHED\_ENABLE\_THREAD\_CONTEXT\_CHECKS, 105  
 PV\_SCHED\_FAIR\_SCHEDULING, 105  
 PV\_SCHED\_LOG\_Q, 105  
 PVEEXECNAMELEN, 105  
 PVSCHEDNAMELEN, 105  
 QUE\_ITER\_BEGIN, 105  
 QUE\_ITER\_END, 105  
 TOsclReady, 106  
 TPVThreadContext, 106  
**OsclProcStatus**, 475  
   ALREADY\_SUSPENDED\_ERROR, 475  
   BAD\_THREADID\_ADDR\_ERROR, 475  
   EXCEED\_MAX\_COUNT\_VARIABLE\_ERROR, 476  
   EXCEED\_MAX\_SEM\_COUNT\_ERROR, 476  
   INVALID\_ACCESS\_ERROR, 476  
   INVALID\_ARGUMENT\_ERROR, 476  
   INVALID\_FUNCTION\_ERROR, 476  
   INVALID\_HANDLE\_ERROR, 476  
   INVALID\_OPERATION\_ERROR, 476  
   INVALID\_PARAM\_ERROR, 475  
   INVALID\_POINTER\_ERROR, 476  
   INVALID\_PRIORITY\_ERROR, 475  
   INVALID\_THREAD\_ERROR, 475  
   INVALID\_THREAD\_ID\_ERROR, 475  
   MAX\_THRDS\_REACHED\_ERROR, 475  
   MUTEX\_LOCKED\_ERROR, 476  
   NO\_PERMISSION\_ERROR, 475  
   NOT\_ENOUGH\_MEMORY\_ERROR, 475  
   NOT\_ENOUGH\_RESOURCES\_ERROR, 475  
   NOT\_IMPLEMENTED, 476  
   NOT\_SUSPENDED\_ERROR, 475  
   OTHER\_ERROR, 475  
   OUTOFGMEMORY\_ERROR, 475  
   PSHARED\_ATTRIBUTE\_SETTING\_ERROR, 476  
   PSHARED\_NOT\_ZERO\_ERROR, 476  
   RELOCK\_MUTEX\_ERROR, 476  
   SEM\_NOT\_SIGNALLED\_ERROR, 476  
   SUCCESS\_ERROR, 475  
   SYSTEM\_RESOURCES\_UNAVAILABLE\_ERROR, 476  
   THREAD\_1\_INACTIVE\_ERROR, 475  
   THREAD\_BLOCK\_ERROR, 476  
   THREAD\_NOT\_OWN\_MUTEX\_ERROR, 476  
   TOO\_MANY\_THREADS\_ERROR, 475  
   WAIT\_ABANDONED\_ERROR, 476  
   WAIT\_TIMEOUT\_ERROR, 476  
**OsclProcStatus**  
   eOsclProcError, 475  
**OsclPtr**  
   Append, 477  
   Length, 477  
   OsclPtr, 477  
   Ptr, 477  
   Set, 477  
   SetLength, 477  
   Zero, 477  
**OsclPtrAdd**  
   osclproc, 106  
**OsclPtrC**, 479  
   OsclPtrC, 480  
**OsclPtrC**  
   Left, 480  
   Length, 480  
   OsclPtrC, 480  
   Ptr, 480  
   Right, 480  
   Set, 480  
   SetLength, 480  
   Zero, 480  
**OsclPtrSub**  
   osclproc, 106  
**OsclRand**, 481  
**OsclRand**  
   Rand, 481  
   Seed, 481  
**OsclReadFD**  
   osclconfig\_io.h, 818  
**OsclReadyAlloc**, 482  
**OsclReadyAlloc**  
   allocate, 482  
   allocate\_fl, 482  
   deallocate, 482  
**OsclReadyCompare**, 483  
   PVActiveBase, 615  
**OsclReadyCompare**  
   compare, 483  
**OsclReadyQ**, 484  
   OsclExecSchedulerCommonBase, 397  
   PVActiveBase, 615  
   PVActiveStats, 616  
**OsclReadyQ**  
   Callback, 485  
   Construct, 485  
   Depth, 485  
   IsIn, 485  
   PendComplete, 485  
   PopTop, 485  
   RegisterForCallback, 485  
   Remove, 485  
   ThreadLogoff, 485

ThreadLogon, 485  
 TimerCallback, 485  
 Top, 485  
 WaitAndPopTop, 485  
 WaitForRequestComplete, 485  
**OsclReadySetPosition**  
 PVActiveBase, 615  
**OsclRecv**  
 osclconfig\_io.h, 818  
**OsclRecvFrom**  
 osclconfig\_io.h, 818  
**OsclRecvFromMethod**, 486  
**OsclRecvFromMethod**  
 ~OsclRecvFromMethod, 486  
 GetRecvData, 486  
 NewL, 486  
 RecvFrom, 486  
 RecvFromRequest, 486  
**OsclRecvFromRequest**, 488  
 OsclRecvFromRequest, 488  
 OsclSocketI, 539  
**OsclRecvFromRequest**  
 GetRecvData, 488  
 OsclRecvFromRequest, 488  
 RecvFrom, 488  
 Success, 488  
**OsclRecvMethod**, 490  
**OsclRecvMethod**  
 ~OsclRecvMethod, 490  
 GetRecvData, 490  
 NewL, 490  
 Recv, 490  
 RecvRequest, 490  
**OsclRecvRequest**, 491  
 OsclRecvRequest, 491  
 OsclSocketI, 539  
**OsclRecvRequest**  
 GetRecvData, 491  
 OsclRecvRequest, 491  
 Recv, 491  
 Success, 491  
**OsclRefCounter**, 492  
**OsclRefCounter**  
 ~OsclRefCounter, 492  
 addRef, 492  
 getCount, 492  
 removeRef, 492  
**OsclRefCounterDA**, 494  
 OsclRefCounterDA, 494  
**OsclRefCounterDA**  
 ~OsclRefCounterDA, 494  
 addRef, 495  
 getCount, 495  
 OsclRefCounterDA, 494  
 removeRef, 495  
 getCount, 495  
 OsclRefCounterMemFrag, 496  
 OsclRefCounterMemFrag, 496  
**OsclRefCounterMemFrag**  
 ~OsclRefCounterMemFrag, 496  
 getCapacity, 497  
 getCount, 497  
 getMemFrag, 497  
 getMemFragPtr, 497  
 getMemFragSize, 497  
 getRefCounter, 497  
 operator=, 497  
 OsclRefCounterMemFrag, 496  
**OsclRefCounterMTDA**, 498  
 OsclRefCounterMTDA, 498  
**OsclRefCounterMTDA**  
 ~OsclRefCounterMTDA, 498  
 addRef, 499  
 getCount, 499  
 OsclRefCounterMTDA, 498  
 removeRef, 499  
**OsclRefCounterMTSA**, 500  
 OsclRefCounterMTSA, 500  
**OsclRefCounterMTSA**  
 ~OsclRefCounterMTSA, 500  
 addRef, 501  
 getCount, 501  
 OsclRefCounterMTSA, 500  
 removeRef, 501  
**OsclRefCounterSA**, 502  
 OsclRefCounterSA, 502  
**OsclRefCounterSA**  
 ~OsclRefCounterSA, 502  
 addRef, 503  
 getCount, 503  
 OsclRefCounterSA, 502  
 removeRef, 503  
**OsclRegistryAccessClient**, 504  
 OsclRegistryAccessClient, 504  
 OsclRegistryClientImpl, 512  
 OsclRegistryServTlsImpl, 515  
**OsclRegistryAccessClient**  
 ~OsclRegistryAccessClient, 504  
 Close, 504  
 Connect, 504  
 GetFactories, 504  
 GetFactory, 504  
 OsclRegistryAccessClient, 504  
**OsclRegistryAccessClientImpl**, 506  
**OsclRegistryAccessClientTlsImpl**, 507  
**OsclRegistryAccessElement**, 508  
**OsclRegistryAccessElement**  
 iFactory, 508  
 iMimeTypeString, 508

**OsclRegistryClient**, 509  
 OsclRegistryClient, 509  
 OsclRegistryClientImpl, 512  
 OsclRegistryServTlsImpl, 515  
**OsclRegistryClient**  
 ~OsclRegistryClient, 509  
 Close, 509  
 Connect, 509  
 OsclRegistryClient, 509  
 Register, 509  
 UnRegister, 510  
**OsclRegistryClientImpl**, 511  
**OsclRegistryClientImpl**  
 Close, 512  
 Connect, 512  
 GetFactories, 512  
 GetFactory, 512  
 OsclRegistryAccessClient, 512  
 OsclRegistryClient, 512  
 Register, 512  
 UnRegister, 512  
**OsclRegistryClientTlsImpl**, 513  
**OsclRegistryServTlsImpl**, 514  
 OsclRegistryServTlsImpl, 515  
**OsclRegistryServTlsImpl**  
 ~OsclRegistryServTlsImpl, 515  
 Close, 515  
 Connect, 515  
 GetFactories, 515  
 GetFactory, 515  
 OsclRegistryAccessClient, 515  
 OsclRegistryClient, 515  
 OsclRegistryServTlsImpl, 515  
 Register, 515  
 UnRegister, 515  
**OsclReturnCode**  
 oscrror, 93  
**OsclScheduler**, 516  
 OsclErrorTrapImp, 378  
 OsclExecScheduler, 390  
 OsclExecSchedulerCommonBase, 397  
**OsclScheduler**  
 Cleanup, 516  
 Init, 516  
**OsclSchedulerCommonBase**  
 PVActiveBase, 615  
**OsclSchedulerObserver**, 517  
**OsclSchedulerObserver**  
 ~OsclSchedulerObserver, 517  
 OsclSchedulerReadyCallback, 517  
 OsclSchedulerTimerCallback, 517  
**OsclSchedulerReadyCallback**  
 OsclSchedulerObserver, 517  
**OsclSchedulerTimerCallback**

OsclSchedulerObserver, 517  
 OsclSchedulerReadyCallback, 517  
 OsclSchedulerTimerCallback, 517  
 OsclSchedulerObserver, 517  
 OsclSchedulerTimerCallback

OsclSchedulerObserver, 517  
 OsclScopedLock, 518  
 OsclScopedLock, 518  
**OsclScopedLock**  
 ~OsclScopedLock, 518  
 OsclScopedLock, 518  
**OsclSelect**, 519  
 OsclSelect, 520  
**OsclSelect**  
 iErrAlloc, 520  
 iHeapCheck, 520  
 iOsclBase, 520  
 iOsclErrorTrap, 520  
 iOsclLogger, 520  
 iOsclMemory, 520  
 iOsclScheduler, 520  
 iOutputFile, 520  
 iSchedulerAlloc, 520  
 iSchedulerName, 520  
 iSchedulerReserve, 520  
 OsclSelect, 520  
**OsclSemaphore**, 521  
 OsclSemaphore, 521  
**OsclSemaphore**  
 ~OsclSemaphore, 521  
 Close, 521  
 Create, 521  
 OsclSemaphore, 521  
 Signal, 522  
 TryWait, 522  
 Wait, 522  
**OsclSend**  
 osclconfig\_io.h, 819  
**OsclSendMethod**, 523  
**OsclSendMethod**  
 ~OsclSendMethod, 523  
 GetSendData, 523  
 NewL, 523  
 Send, 523  
 SendRequest, 523  
**OsclSendRequest**, 524  
 OsclSendRequest, 524  
 OsclSocketI, 539  
**OsclSendRequest**  
 GetSendData, 524  
 OsclSendRequest, 524  
 Send, 524  
 Success, 524  
**OsclSendTo**  
 osclconfig\_io.h, 819  
**OsclSendToMethod**, 525  
**OsclSendToMethod**  
 ~OsclSendToMethod, 525  
 GetSendData, 525

NewL, [525](#)  
 SendTo, [525](#)  
 SendToRequest, [525](#)  
**OsclSendToRequest**, [526](#)  
 OsclSendToRequest, [526](#)  
 OsclSocketI, [539](#)  
**OsclSendToRequest**  
 GetSendData, [526](#)  
 OsclSendToRequest, [526](#)  
 SendTo, [526](#)  
 Success, [526](#)  
**OsclSetNonBlocking**  
 osclconfig\_io.h, [819](#)  
**OsclSetRecvBufferSize**  
 osclconfig\_io.h, [819](#)  
**OsclSetSockOpt**  
 osclconfig\_io.h, [819](#)  
**OsclSharedPtr**, [527](#)  
 OsclSharedPtr, [528](#)  
**OsclSharedPtr**  
 ~OsclSharedPtr, [528](#)  
 get\_count, [528](#)  
 GetRefCounter, [528](#)  
 GetRep, [528](#)  
 operator \*, [528](#)  
 operator TheClass \*, [529](#)  
 operator->, [529](#)  
 operator=, [529](#)  
 OsclSharedPtr, [528](#)  
 Unbind, [529](#)  
**OsclShutdown**  
 osclconfig\_io.h, [819](#)  
**OsclShutdownMethod**, [530](#)  
**OsclShutdownMethod**  
 ~OsclShutdownMethod, [530](#)  
 NewL, [530](#)  
 Shutdown, [530](#)  
 ShutdownRequest, [530](#)  
**OsclShutdownRequest**, [531](#)  
 OsclShutdownRequest, [531](#)  
 OsclSocketI, [539](#)  
**OsclShutdownRequest**  
 OsclShutdownRequest, [531](#)  
 Shutdown, [531](#)  
**OsclSingleton**, [532](#)  
 OsclSingleton, [532](#)  
**OsclSingleton**  
 ~OsclSingleton, [532](#)  
 \_Ptr, [533](#)  
 operator \*, [532](#)  
 operator->, [532](#)  
 OsclSingleton, [532](#)  
 set, [532](#)  
**OsclSingletonRegistry**, [534](#)  
**OsclSingletonRegistry**  
 getInstance, [534](#)  
 lockAndGetInstance, [534](#)  
**OsclBase**, [534](#)  
 registerInstance, [534](#)  
 registerInstanceAndUnlock, [534](#)  
**OsclSocket**  
 osclconfig\_io.h, [820](#)  
**OsclSocketCleanup**  
 osclconfig\_io.h, [820](#)  
**OsclSocketI**, [535](#)  
 OsclSocketRequestAO, [553](#)  
 OsclSocketServI, [557](#)  
**OsclSocketI**  
 ~OsclSocketI, [536](#)  
 Accept, [536](#)  
 Bind, [536](#)  
 Close, [536](#)  
 Connect, [536](#)  
 GetPeerName, [536](#)  
 Join, [537](#)  
 Listen, [537](#)  
 Logger, [537](#)  
 MakeAddr, [537](#)  
 MakeMulticastGroupInformation, [537](#)  
 NewL, [537](#)  
 Open, [537](#)  
 OsclAcceptRequest, [539](#)  
 OsclConnectRequest, [539](#)  
 OsclRecvFromRequest, [539](#)  
 OsclRecvRequest, [539](#)  
 OsclSendRequest, [539](#)  
 OsclSendToRequest, [539](#)  
 OsclShutdownRequest, [539](#)  
 OsclTCPSocket, [539](#)  
 OsclUDPSocket, [539](#)  
 ProcessAccept, [537](#)  
 ProcessConnect, [538](#)  
 ProcessRecv, [538](#)  
 ProcessRecvFrom, [538](#)  
 ProcessSend, [538](#)  
 ProcessSendTo, [538](#)  
 ProcessShutdown, [538](#)  
 Recv, [538](#)  
 RecvFrom, [538](#)  
 RecvFromSuccess, [538](#)  
 RecvSuccess, [538](#)  
 Send, [538](#)  
 SendSuccess, [538](#)  
 SendTo, [538](#)  
 SendToSuccess, [538](#)  
 SetRecvBufferSize, [538](#)  
 SetSockOpt, [539](#)  
 Shutdown, [539](#)

Socket, 539  
 ThreadLogoff, 539  
 ThreadLogon, 539  
**OsclSocketIBase**, 540  
     **OsclSocketIBase**, 541  
**OsclSocketIBase**  
     ~**OsclSocketIBase**, 541  
     Accept, 541  
     Bind, 541  
     BindAsync, 541  
     CancelAccept, 542  
     CancelBind, 542  
     CancelConnect, 542  
     CancelFxn, 542  
     CancelListen, 542  
     CancelRecv, 542  
     CancelRecvFrom, 542  
     CancelSend, 542  
     CancelSendTo, 542  
     CancelShutdown, 542  
     Close, 542  
     Connect, 542  
     GetShutdown, 542  
     HasAsyncBind, 542  
     HasAsyncListen, 542  
     iAlloc, 544  
     iSocketServ, 544  
     IsOpen, 542  
     Join, 542  
     Listen, 542  
     ListenAsync, 542  
     Open, 543  
     **OsclSocketIBase**, 541  
     **OsclSocketMethod**, 544  
     **OsclSocketRequest**, 544  
     **OsclSocketRequestAO**, 544  
     **OsclTCPSocket**, 544  
     **OsclUDPSocket**, 544  
     Recv, 543  
     RecvFrom, 543  
     RecvFromSuccess, 543  
     RecvSuccess, 543  
     Send, 543  
     SendSuccess, 543  
     SendTo, 543  
     SendToSuccess, 543  
     Shutdown, 544  
**OsclSocketMethod**, 545  
     **OsclIPSocketI**, 420  
     **OsclSocketIBase**, 544  
     **OsclSocketMethod**, 546  
     **OsclSocketRequestAO**, 553  
**OsclSocketMethod**  
     ~**OsclSocketMethod**, 546  
     Abort, 546  
     AbortAll, 546  
     Alloc, 546  
     CancelMethod, 546  
     ConstructL, 546  
     iContainer, 547  
     iSocketFxn, 547  
     iSocketRequestAO, 547  
     MethodDone, 546  
     **OsclSocketMethod**, 546  
     Run, 546  
     StartMethod, 547  
     ThreadLogoff, 547  
     ThreadLogon, 547  
**OsclSocketObserver**, 548  
**OsclSocketObserver**  
     ~**OsclSocketObserver**, 548  
     HandleSocketEvent, 548  
**OsclSocketRequest**, 549  
     **OsclSocketIBase**, 544  
     **OsclSocketRequest**, 549  
     **OsclSocketRequestAO**, 553  
     **OsclSocketServI**, 557  
**OsclSocketRequest**  
     Activate, 549  
     CancelRequest, 549  
     Complete, 549  
     Fxn, 549  
     iParam, 549  
     iSocketI, 549  
     iSocketRequestAO, 549  
     **OsclSocketRequest**, 549  
**OsclSocketRequestAO**, 550  
     **OsclIPSocketI**, 420  
     **OsclSocketIBase**, 544  
     **OsclSocketRequestAO**, 551  
**OsclSocketRequestAO**  
     ~**OsclSocketRequestAO**, 551  
     Abort, 551  
     Alloc, 551  
     CleanupParam, 551  
     ConstructL, 551  
     DoCancel, 551  
     GetSocketError, 551  
     iContainer, 553  
     Id, 552  
     iParam, 553  
     iParamSize, 553  
     iSocketError, 553  
     NewRequest, 552  
     **OsclSocketI**, 553  
     **OsclSocketMethod**, 553  
     **OsclSocketRequest**, 553  
     **OsclSocketRequestAO**, 551

RequestDone, 552  
 Run, 552  
 SocketI, 552  
 SocketObserver, 552  
 Success, 552  
 OsclSocketSelect  
     osclconfig\_io.h, 820  
 OsclSocketServ, 554  
     OsclSocketServI, 557  
 OsclSocketServ  
     ~OsclSocketServ, 554  
     Close, 554  
     Connect, 554  
     NewL, 555  
     OsclDNS, 555  
     OsclTCPSocket, 555  
     OsclUDPSocket, 555  
 OsclSocketServI, 556  
     OsclSocketServRequestList, 560  
 OsclSocketServI  
     Close, 556  
     Connect, 556  
     IsServerThread, 557  
     LoopbackSocket, 557  
     NewL, 557  
     OsclDNSI, 557  
     OsclSocketI, 557  
     OsclSocketRequest, 557  
     OsclSocketServ, 557  
     OsclSocketServRequestList, 557  
     OsclTCPSocketI, 557  
     OsclUDPSocketI, 557  
 OsclSocketServIBase, 558  
     ESocketServ\_Connected, 558  
     ESocketServ\_Error, 559  
     ESocketServ\_Idle, 558  
     OsclSocketServIBase, 559  
 OsclSocketServIBase  
     ~OsclSocketServIBase, 559  
     Close, 559  
     Connect, 559  
     iAlloc, 559  
     iLogger, 559  
     iServerError, 559  
     iServState, 559  
     IsServConnected, 559  
     OsclSocketServIBase, 559  
     State, 559  
     TSocketServState, 558  
 OsclSocketServRequestList, 560  
     OsclSocketServI, 557  
     OsclSocketServRequestList, 560  
 OsclSocketServRequestList  
     Add, 560  
     Close, 560  
     Open, 560  
     OsclSocketServI, 560  
     OsclSocketServRequestList, 560  
     Remove, 560  
     StartCancel, 560  
     WaitOnRequests, 560  
     Wakeup, 560  
 OsclSocketServRequestQElem, 562  
     OsclSocketServRequestQElem, 562  
 OsclSocketServRequestQElem  
     iCancel, 562  
     iSelect, 562  
     iSocketRequest, 562  
     OsclSocketServRequestQElem, 562  
 OsclSocketStartup  
     osclconfig\_io.h, 820  
 OsclSocketTOS, 563  
     EPVCritic\_Ecp, 563  
     EPVFlash, 563  
     EPVHiRel, 563  
     EPVHiThrpt, 563  
     EPVImmediate, 563  
     EPVInetControl, 563  
     EPVLDelay, 563  
     EPVNetControl, 563  
     EPVNoTOS, 563  
     EPVOverrideFlash, 563  
     EPVPriority, 563  
     EPVRoutine, 563  
     OsclSocketTOS, 564  
 OsclSocketTOS  
     ClearTOS, 564  
     GetTOS, 564  
     OsclSocketTOS, 564  
     SetPrecedence, 564  
     SetPriority, 564  
     TPVServicePrecedence, 563  
     TPVServicePriority, 563  
 OsclSuccess  
     osclerror, 92  
 OsclTagTreeType  
     osclmemory, 59  
 OsclTCPSocket, 565  
     OsclSocketI, 539  
     OsclSocketIBase, 544  
     OsclSocketServ, 555  
 OsclTCPSocket  
     ~OsclTCPSocket, 566  
     Accept, 566  
     Bind, 566  
     BindAsync, 566  
     CancelAccept, 567  
     CancelBind, 567

CancelConnect, [567](#)  
 CancelListen, [567](#)  
 CancelRecv, [567](#)  
 CancelSend, [567](#)  
 CancelShutdown, [567](#)  
 Close, [568](#)  
 Connect, [568](#)  
 GetAcceptedSocketL, [568](#)  
 GetPeerName, [568](#)  
 GetRecvData, [569](#)  
 GetSendData, [569](#)  
 Listen, [569](#)  
 ListenAsync, [569](#)  
 NewL, [569](#)  
 Recv, [570](#)  
 Send, [570](#)  
 SetOptionToReuseAddress, [570](#)  
 SetTOS, [571](#)  
 Shutdown, [571](#)  
 ThreadLogoff, [571](#)  
 ThreadLogon, [571](#)  
**OscITCPSocketI**, [572](#)  
   **OscISocketServI**, [557](#)  
**OscITCPSocketI**  
   ~OscITCPSocketI, [573](#)  
   Accept, [573](#)  
   BindAsync, [573](#)  
   CancelAccept, [573](#)  
   CancelBind, [573](#)  
   CancelConnect, [573](#)  
   CancelListen, [573](#)  
   CancelRecv, [573](#)  
   CancelSend, [573](#)  
   CancelShutdown, [573](#)  
   Close, [573](#)  
   Connect, [573](#)  
   GetAcceptedSocketL, [573](#)  
   GetRecvData, [573](#)  
   GetSendData, [573](#)  
   Listen, [573](#)  
   ListenAsync, [574](#)  
   NewL, [574](#)  
   Recv, [574](#)  
   Send, [574](#)  
   Shutdown, [574](#)  
   ThreadLogoff, [574](#)  
   ThreadLogon, [574](#)  
**OscIThread**, [575](#)  
   **OscIThread**, [575](#)  
**OscIThread**  
   ~OscIThread, [575](#)  
   CanTerminate, [575](#)  
   CompareId, [576](#)  
   Create, [576](#)  
     Exit, [576](#)  
     GetId, [576](#)  
     GetPriority, [577](#)  
     OsclThread, [575](#)  
     Resume, [577](#)  
     SetPriority, [577](#)  
     SleepMillisec, [577](#)  
     Suspend, [578](#)  
     Terminate, [578](#)  
   **OscIThread\_State**  
     oscl\_thread.h, [788](#)  
   **OscIThreadLock**, [579](#)  
     **OscIThreadLock**, [579](#)  
   **OscIThreadLock**  
     ~OscIThreadLock, [579](#)  
     Lock, [579](#)  
     OsclThreadLock, [579](#)  
     Unlock, [579](#)  
   **OscIThreadPriority**  
     oscl\_thread.h, [788](#)  
   **OscITickCount**, [580](#)  
   **OscITickCount**  
     MsecToTicks, [580](#)  
     TickCount, [580](#)  
     TickCountFrequency, [580](#)  
     TickCountPeriod, [580](#)  
     TicksToMsec, [580](#)  
   **OSCLTICKCOUNT\_MAX\_TICKS**  
     osclutil, [69](#)  
   **OscITimer**, [582](#)  
     **OscITimer**, [583](#)  
   **OscITimer**  
     ~OscITimer, [583](#)  
     callback\_timer\_type, [583](#)  
     CallbackTimer< Alloc >, [584](#)  
     Cancel, [583](#)  
     Clear, [583](#)  
     OscITimer, [583](#)  
     Request, [583](#)  
     SetExactFrequency, [583](#)  
     SetFrequency, [584](#)  
     SetObserver, [584](#)  
     TimerBaseElapsed, [584](#)  
   **OscITimerCompare**, [585](#)  
     OscISchedulerCommonBase, [397](#)  
   **OscITimerCompare**  
     compare, [585](#)  
   **OscITimerObject**, [586](#)  
     OscISchedulerCommonBase, [399](#)  
     OscITimerObject, [587](#)  
     PVActiveBase, [615](#)  
     PVActiveStats, [616](#)  
     PVThreadContext, [635](#)  
   **OscITimerObject**

~OsclTimerObject, 587  
 AddToScheduler, 587  
 After, 587  
 Cancel, 587  
 DoCancel, 587  
 IsBusy, 588  
 OsclTimerObject, 587  
 Priority, 588  
 RemoveFromScheduler, 588  
 RunError, 588  
 RunIfNotReady, 588  
 SetBusy, 588  
 SetStatus, 588  
 Status, 589  
 StatusRef, 589  
 OsclTimerObserver, 590  
 OsclTimerObserver  
   ~OsclTimerObserver, 590  
   TimeoutOccurred, 590  
 OsclTimerQ, 591  
 OsclTimerQ  
   Add, 591  
   Construct, 591  
   IsIn, 591  
   Pop, 591  
   PopTop, 591  
   Remove, 591  
   Top, 591  
 OsclTLS, 592  
   OsclTLS, 592  
 OsclTLS  
   ~OsclTLS, 592  
   \_Ptr, 593  
   operator \*, 592  
   operator->, 592  
   OsclTLS, 592  
   set, 592  
 OsclTLSEx, 594  
   OsclTLSEx, 594  
 OsclTLSEx  
   ~OsclTLSEx, 594  
   \_Ptr, 595  
   operator \*, 594  
   operator->, 594  
   OsclTLSEx, 594  
   set, 594  
 OsclTLSRegistry, 596  
 OsclTLSRegistry  
   getInstance, 596  
   OsclBase, 596  
   registerInstance, 596  
 OsclTLSRegistryEx, 597  
 OsclTLSRegistryEx  
   getInstance, 597  
   registerInstance, 597  
   OsclTrapItem, 598  
   OsclTrapItem, 598  
 OsclTrapItem  
   OsclTrapItem, 598  
   OsclTrapStack, 598  
   OsclTrapStackItem, 598  
 OsclTrapOperation  
   osclerror, 93  
 OsclTrapStack, 599  
   OsclErrorTrapImp, 378  
   OsclTrapItem, 598  
 OsclTrapStackItem  
   OsclError, 599  
   OsclErrorTrap, 599  
   OsclErrorTrapImp, 599  
 OsclTrapStackItem, 600  
   OsclTrapItem, 598  
   OsclTrapStackItem, 600  
 OsclTrapStackItem  
   iCBase, 600  
   iNext, 600  
   iTAny, 600  
   iTrapOperation, 600  
   OsclTrapStackItem, 600  
 OsclUDPSocket, 601  
   OsclSocketI, 539  
   OsclSocketIBase, 544  
   OsclSocketServ, 555  
 OsclUDPSocket  
   ~OsclUDPSocket, 602  
   Bind, 602  
   BindAsync, 602  
   CancelBind, 602  
   CancelRecvFrom, 602  
   CancelSendTo, 602  
   Close, 603  
   GetPeerName, 603  
   GetRecvData, 603  
   GetSendData, 603  
   Join, 603  
   JoinMulticastGroup, 604  
   NewL, 604  
   RecvFrom, 604  
   SendTo, 605  
   SetMulticastTTL, 605  
   SetOptionToReuseAddress, 605  
   SetRecvBufferSize, 606  
   SetTOS, 606  
   ThreadLogoff, 606  
   ThreadLogon, 606  
 OsclUDPSocketI, 607  
   OsclSocketServI, 557  
 OsclUDPSocketI

---

~OsclUDPSocketI, 608  
 BindAsync, 608  
 CancelBind, 608  
 CancelRecvFrom, 608  
 CancelSendTo, 608  
 Close, 608  
 GetRecvData, 608  
 GetSendData, 608  
 JoinMulticastGroup, 608  
 NewL, 608  
 RecvFrom, 608  
 SendTo, 608  
 SetMulticastTTL, 608  
 ThreadLogoff, 608  
 ThreadLogon, 608  
**OsclUid32**  
     oscl\_uuid.h, 799  
**OsclUnMakeInAddr**  
     osclconfig\_io.h, 820  
**OsclUnMakeSockAddr**  
     osclconfig\_io.h, 821  
**osclutil**  
     ~OSCL\_HeapString, 84  
     ~OSCL\_StackString, 84  
     ~OSCL\_wHeapString, 84  
     ~OSCL\_wStackString, 84  
     APPEND\_MEDIA\_AT\_END, 84  
     BufferFreeFuncPtr, 69  
     EOSCL\_StringOp\_CompressASCII, 70  
     EOSCL\_StringOp\_UTF16ToUTF8, 70  
     EOSCL\_wStringOp\_ExpandASCII, 70  
     EOSCL\_wStringOp\_UTF8ToUTF16, 70  
     extract\_string, 70  
     get\_cstr, 70  
     get\_maxsize, 71  
     get\_size, 71  
     get\_str, 72  
     GetBufferState, 72  
     GetFragment, 72  
     MAX\_NUMBER\_OF\_BYTE\_PER\_UTF8,  
         69  
     MediaTimestamp, 69  
     operator=, 72–74  
     oscl\_abs, 74  
     OSCL\_ASCII\_CASE\_MAGIC\_BIT, 84  
     oscl\_asin, 74  
     oscl\_atan, 74  
     oscl\_cos, 74  
     oscl\_exp, 74  
     oscl\_floor, 74  
     OSCL\_HeapString, 74, 75  
     oscl\_isdigit, 69  
     oscl\_log, 75  
     oscl\_log10, 75  
     oscl\_pow, 75  
     oscl\_sin, 76  
     oscl\_snprintf, 76  
     oscl\_sqrt, 76  
     OSCL\_StackString, 76, 77  
     oscl\_str\_escape\_xml, 77  
     oscl\_str\_is\_valid\_utf8, 77  
     oscl\_str\_need\_escape\_xml, 78  
     oscl\_str\_truncate\_utf8, 78  
     oscl\_str\_unescape\_uri, 78, 79  
     oscl\_tan, 79  
     OSCL\_TStrPtrLen, 69  
     oscl\_UncodeToUTF8, 79  
     oscl\_UTF8ToUnicode, 80  
     oscl\_vsnprintf, 80, 82  
     OSCL\_wHeapString, 82  
     OSCL\_wStackString, 82  
     OsclComponentFactory, 69  
     OSCLTICKCOUNT\_MAX\_TICKS, 69  
     PV\_atof, 82  
     PV\_atoi, 82  
     set, 82–84  
     skip\_to\_line\_term, 84  
     skip\_to\_whitespace, 84  
     skip\_whitespace, 84  
     skip\_whitespace\_and\_line\_term, 84  
     StrCSumPtrLen, 69  
     StrPtrLen, 69  
     TOSCL\_StringOp, 70  
     TOSCL\_wStringOp, 70  
     WStrPtrLen, 69  
**OsclUuid**, 610  
     OsclUuid, 611  
**OsclUuid**  
     data1, 611  
     data2, 611  
     data3, 611  
     data4, 611  
     operator!=, 611  
     operator=, 611  
     operator==, 611  
     OsclUuid, 611  
**OsclValidInetAddr**  
     osclconfig\_io.h, 821  
**OsclWriteFD**  
     osclconfig\_io.h, 821  
**other**  
     Oscl\_TAlloc::rebind, 284  
**other\_chartype**  
     OSCL\_FastString, 176  
     OSCL\_HeapString, 197  
     OSCL\_HeapStringA, 199  
     OSCL\_StackString, 258  
     OSCL\_wFastString, 295

OSCL\_wHeapString, 298  
 OSCL\_wHeapStringA, 300  
 OSCL\_wStackString, 303  
**OTHER\_ERROR**  
 OsclProcStatus, 475  
**OUTOFMEMORY\_ERROR**  
 OsclProcStatus, 475  
 overwrite  
 CFastRep, 128  
  
 pad  
 MM\_AllocBlockFence, 147  
 MM\_AllocBlockHdr, 148  
 pair\_citerator\_citerator  
 Oscl\_Map, 218  
 pair\_iterator\_bool  
 Oscl\_Map, 218  
 Oscl\_TagTree, 270  
 pair\_iterator\_iterator  
 Oscl\_Map, 218  
**pAllocInfo**  
 MM\_AllocNode, 151  
 parent  
 Oscl\_Rb\_Tree\_Node\_Base, 255  
 Oscl\_TagTree::Node, 280  
**pAudit**  
 OsclAuditCB, 321  
**pBasePosition**  
 OsclBinStream, 339  
**pBuffer**  
 OsclFileCacheBuffer, 404  
 peakNumAllocs  
 MM\_Stats\_t, 165  
 peakNumBytes  
 MM\_Stats\_t, 165  
**PendComplete**  
 OsclActiveObject, 312  
 OsclExecSchedulerCommonBase, 396  
 OsclReadyQ, 485  
**PendForExec**  
 OsclActiveObject, 312  
**per\_allocation\_overhead**  
 MM\_AuditOverheadStats, 161  
**perms**  
 oscl\_stat\_buf, 259  
**PersistHostAddress**  
 GetHostNameParam, 136  
**pFileName**  
 MM\_AllocInfo, 150  
**pMemBlock**  
 MM\_AllocInfo, 150  
 MM\_AllocQueryInfo, 152  
**pMMFIParam**  
 OsclMemStatsNode, 458  
  
 pMMStats  
 OsclMemStatsNode, 458  
**pNext**  
 MM\_AllocNode, 151  
**pNode**  
 MM\_AllocBlockHdr, 148  
**pointer**  
 MemAllocator, 146  
 Oscl\_Map, 218  
 Oscl\_Queue, 237  
 Oscl\_Rb\_Tree, 244  
 Oscl\_Rb\_Tree\_Const\_Iterator, 248  
 Oscl\_Rb\_Tree\_Iterator, 251  
 Oscl\_TagTree::const\_iterator, 274  
 Oscl\_TagTree::iterator, 277  
 Oscl\_TAlloc, 282  
 Oscl\_Vector, 286  
**Pop**  
 OsclError, 372  
 OsclTimerQ, 591  
**pop**  
 Oscl\_Queue, 238  
 Oscl\_Queue\_Base, 240  
 OsclPriorityQueue, 472  
**pop\_back**  
 Oscl\_Vector, 288  
 Oscl\_Vector\_Base, 292  
**pop\_heap**  
 OsclPriorityQueue, 472  
 OsclPriorityQueueBase, 474  
**PopDealloc**  
 OsclError, 372, 373  
**PopTop**  
 OsclReadyQ, 485  
 OsclTimerQ, 591  
**port**  
 OsclNetworkAddress, 466  
**PositionInBlock**  
 OsclBinStream, 338  
**pPosition**  
 OsclBinStream, 339  
**pPrev**  
 MM\_AllocNode, 151  
**Preceeds**  
 OsclFileCacheBuffer, 404  
**PreRead**  
 OsclFileCacheBuffer, 404  
**PreWrite**  
 OsclFileCacheBuffer, 404  
**Priority**  
 OsclActiveObject, 312  
 OsclTimerObject, 588  
**ProcessAccept**  
 OsclSocketI, 537

ProcessConnect  
     OsclSocketI, 538  
 ProcessRecv  
     OsclSocketI, 538  
 ProcessRecvFrom  
     OsclSocketI, 538  
 ProcessSend  
     OsclSocketI, 538  
 ProcessSendTo  
     OsclSocketI, 538  
 ProcessShutdown  
     OsclSocketI, 538  
 pRootNode  
     MM\_AllocBlockHdr, 148  
 pruneSubtree  
     MM\_Audit\_Imp, 159  
 PSHARED\_ATTRIBUTE\_SETTING\_ERROR  
     OsclProcStatus, 476  
 PSHARED\_NOT\_ZERO\_ERROR  
     OsclProcStatus, 476  
 pStats  
     MM\_Stats\_CB, 163  
 pStatsNode  
     MM\_AllocInfo, 150  
     OsclAuditCB, 321  
 Ptr  
     OsclPtr, 477  
     OsclPtrC, 480  
 ptr  
     OsclMemoryFragment, 442  
     StrPtrLen, 648  
     WStrPtrLen, 659  
 push  
     Oscl\_Queue, 238  
     Oscl\_Queue\_Base, 240  
     OsclPriorityQueue, 472  
 push\_back  
     Oscl\_Vector, 289  
     Oscl\_Vector\_Base, 292  
 push\_front  
     Oscl\_Vector, 289  
     Oscl\_Vector\_Base, 293  
 push\_heap  
     OsclPriorityQueue, 472  
     OsclPriorityQueueBase, 474  
 PushL  
     OsclError, 373  
 PV8601TIME\_BUFFER\_SIZE  
     osclbase, 46  
 PV8601timeStrBuf  
     osclbase, 35  
 PV8601ToRFC822  
     osclbase, 44  
 PV\_atof  
     osclutil, 82  
 PV\_atoi  
     osclutil, 82  
 PV\_CHAR\_CLOSE\_BRACKET  
     oscl\_uuid\_utils.h, 800  
 PV\_CHAR\_COMMA  
     oscl\_uuid\_utils.h, 800  
 PV\_DNS\_IS\_THREAD  
     oscl\_dns\_tuneables.h, 677  
 PV\_DNS\_SERVER  
     oscl\_dns\_tuneables.h, 677  
 PV\_DYNAMIC\_LOADING\_CONFIG\_FILE\_PATH  
     osclconfig\_lib.h, 824  
 PV\_OSCL\_SOCKET\_1MB\_RECV\_BUF  
     oscl\_socket\_tuneables.h, 772  
 PV\_OSCL\_SOCKET\_SERVER\_LOGGER\_OUTPUT  
     oscl\_socket\_tuneables.h, 772  
 PV\_OSCL\_SOCKET\_STATS\_LOGGING  
     oscl\_socket\_tuneables.h, 772  
 PV\_RUNTIME\_LIB\_FILENAME\_EXTENSION  
     osclconfig\_lib.h, 824  
 PV\_SCHED\_CHECK\_Q  
     osclproc, 105  
 PV\_SCHED\_ENABLE\_AO\_STATS  
     osclproc, 105  
 PV\_SCHED\_ENABLE\_LOOP\_STATS  
     osclproc, 105  
 PV\_SCHED\_ENABLE\_PERF\_LOGGING  
     osclproc, 105  
 PV\_SCHED\_ENABLE\_THREAD\_CONTEXT\_CHECKS  
     osclproc, 105  
 PV\_SCHED\_FAIR\_SCHEDULING  
     osclproc, 105  
 PV\_SCHED\_LOG\_Q  
     osclproc, 105  
 PV\_SOCKET\_REQUEST\_AO\_PRIORITY  
     oscl\_socket\_tuneables.h, 772  
 PV\_SOCKET\_SERVER  
     oscl\_socket\_tuneables.h, 772  
 PV\_SOCKET\_SERVER\_AO\_INTERVAL\_MSEC  
     oscl\_socket\_tuneables.h, 773  
 PV\_SOCKET\_SERVER\_AO\_PRIORITY  
     oscl\_socket\_tuneables.h, 773  
 PV\_SOCKET\_SERVER\_IS\_THREAD  
     oscl\_socket\_tuneables.h, 773  
 PV\_SOCKET\_SERVER\_SELECT  
     oscl\_socket\_tuneables.h, 773  
 PV\_SOCKET\_SERVER\_SELECT\_LOOPBACK\_SOCKET

---

oscl\_socket\_tuneables.h, 773  
**PV\_SOCKET\_SERVER\_SELECT\_-TIMEOUT\_MSEC**  
 oscl\_socket\_tuneables.h, 773  
**PV\_SOCKET\_SERVER\_THREAD\_-PRIORITY**  
 oscl\_socket\_tuneables.h, 773  
**PV\_SOCKET\_SERVI\_STATS**  
 oscl\_socket\_tuneables.h, 773  
**PVActiveBase**, 612  
 OsclExecSchedulerBase, 391  
 OsclExecSchedulerCommonBase, 399  
**PVActiveBase**, 613  
**PVActiveStats**, 616  
 PVThreadContext, 635  
**PVActiveBase**  
 ~PVActiveBase, 613  
 Activate, 613  
 AddToScheduler, 613  
 Cancel, 613  
 Destroy, 613  
 DoCancel, 613  
 iAddedNum, 615  
 iBusy, 615  
 iName, 615  
 iPVAstats, 615  
 iPVRreadyQLink, 615  
 IsAdded, 613  
 IsInAnyQ, 614  
 iStatus, 615  
 iThreadContext, 615  
 OsclActiveObject, 615  
 OsclExecScheduler, 615  
 OsclReadyCompare, 615  
 OsclReadyQ, 615  
 OsclReadySetPosition, 615  
 OsclSchedulerCommonBase, 615  
 OsclTimerObject, 615  
**PVActiveBase**, 613  
**PVActiveStats**, 615  
 RemoveFromScheduler, 614  
 Run, 614  
 RunError, 614  
**PVActiveStats**, 616  
 OsclExecSchedulerCommonBase, 399  
**PVActiveBase**, 615  
**PVActiveStats**  
 OsclActiveObject, 616  
 OsclExecScheduler, 616  
 OsclExecSchedulerCommonBase, 616  
 OsclReadyQ, 616  
 OsclTimerObject, 616  
**PVActiveBase**, 616  
**PVCleanupStack**

\_OsclHeapBase, 111  
**PVError\_DoLeave**  
 oscl\_error\_imp\_fatalerror.h, 685  
 oscl\_error\_imp\_jumps.h, 687  
 osclerror, 92  
**PVERRORTYPE\_JUMPS**  
 osclerror, 92  
**PVERRORTRAP\_REGISTRY**  
 osclerror, 92  
**PVERRORTRAP\_REGISTRY\_ID**  
 osclerror, 93  
**PVEXECNAMELEN**  
 osclproc, 105  
**PVLogger**, 617  
 ~PVLogger, 618  
 AddAppender, 618  
 AddFilter, 618  
 alloc\_type, 618  
 Cleanup, 619  
 DisableAppenderInheritance, 619  
 filter\_status\_type, 618  
 GetLoggerObject, 619  
 GetLogLevel, 619  
 GetNumAppendlers, 619  
 GetParent, 620  
 Init, 620  
 IsActive, 620  
 log\_level\_type, 618  
 LogMsgBuffers, 620  
 LogMsgBuffersV, 620  
 LogMsgString, 621  
 LogMsgStringV, 621  
 message\_id\_type, 618  
 PVLogger, 618  
 PVLoggerRegistry, 622  
 RemoveAppender, 621  
 SetLogLevel, 621  
 SetLogLevelAndPropagate, 622  
 SetParent, 622  
**pvlogger.h**, 849  
 \_PVLOGGER\_LOGBIN, 851  
 \_PVLOGGER\_LOGBIN\_V, 851  
 \_PVLOGGER\_LOGMSG, 851  
 \_PVLOGGER\_LOGMSG\_V, 851  
 PVLOGGER\_ENABLE, 851  
 PVLOGGER\_INST\_LEVEL, 852  
 PVLOGGER\_INST\_LEVEL\_SUPPORT, 852  
 PVLOGGER\_LEVEL\_UNINITIALIZED, 855  
 PVLOGGER\_LOG\_USE\_ONLY, 852  
 PVLOGGER\_LOGBIN, 852  
 PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_HLDBG, 852

PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
   INST\_LLDBG, 853  
 PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
   INST\_MLDBG, 853  
 PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
   INST\_PROF, 853  
 PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
   INST\_REL, 853  
 PVLOGGER\_LOGBIN\_V, 853  
 PVLOGGER\_LOGBIN\_V\_-  
   PVLOGMSG\_INST\_LLDBG, 853  
 PVLOGGER\_LOGBIN\_V\_-  
   PVLOGMSG\_INST\_LLDBG, 853  
 PVLOGGER\_LOGBIN\_V\_-  
   PVLOGMSG\_INST\_PROF, 853  
 PVLOGGER\_LOGBIN\_V\_-  
   PVLOGMSG\_INST\_REL, 853  
 PVLOGGER\_LOGBIN\_V\_-  
   PVLOGMSG\_V\_INST\_MLDBG,  
   853  
 PVLOGGER\_LOGMSG, 853  
 PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
   INST\_LLDBG, 853  
 PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
   INST\_LLDBG, 854  
 PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
   INST\_MLDBG, 854  
 PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
   INST\_PROF, 854  
 PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
   INST\_REL, 854  
 PVLOGGER\_LOGMSG\_V, 854  
 PVLOGGER\_LOGMSG\_V\_-  
   PVLOGMSG\_INST\_LLDBG, 854  
 PVLOGGER\_LOGMSG\_V\_-  
   PVLOGMSG\_INST\_LLDBG, 854  
 PVLOGGER\_LOGMSG\_V\_-  
   PVLOGMSG\_INST\_MLDBG,  
   854  
 PVLOGGER\_LOGMSG\_V\_-  
   PVLOGMSG\_INST\_PROF, 854  
 PVLOGGER\_LOGMSG\_V\_-  
   PVLOGMSG\_INST\_REL, 854  
 PVLOGMSG\_ALERT, 855  
 PVLOGMSG\_CRIT, 855  
 PVLOGMSG\_DEBUG, 855  
 PVLOGMSG\_EMERG, 855  
 PVLOGMSG\_ERR, 855  
 PVLOGMSG\_FATAL\_ERROR, 855  
 PVLOGMSG\_INFO, 856  
 PVLOGMSG\_INST\_LLDBG, 854  
 PVLOGMSG\_INST\_LLDBG, 854  
 PVLOGMSG\_INST\_MLDBG, 854  
 PVLOGMSG\_INST\_PROF, 855  
 PVLOGMSG\_INST\_REL, 855  
 PVLOGMSG\_STACK\_TRACE, 856  
 PVLOGMSG\_STATISTIC, 856  
 PVLOGMSG\_VERBOSE, 856  
 PVLOGMSG\_WARNING, 856  
 pvlogger\_accessories.h, 857  
   PVLOGGER\_FILTER\_ACCEPT, 857  
   PVLOGGER\_FILTER\_NEUTRAL, 857  
   PVLOGGER\_FILTER\_REJECT, 857  
 pvlogger\_c.h, 858  
   PVLOGGER\_C\_INST\_LEVEL, 859  
   pvLogger\_GetLoggerObject, 859  
   pvLogger\_IsActive, 859  
   pvLogger\_LogMsgString, 859  
   PVLOGMSG\_C\_ALERT, 859  
   PVLOGMSG\_C\_CRIT, 859  
   PVLOGMSG\_C\_EMERG, 859  
   PVLOGMSG\_C\_ERR, 859  
   PVLOGMSG\_C\_INFO, 859  
   PVLOGMSG\_C\_INST\_LLDBG, 859  
   PVLOGMSG\_C\_INST\_LLDBG, 859  
   PVLOGMSG\_C\_INST\_MLDBG, 859  
   PVLOGMSG\_C\_INST\_PROF, 859  
   PVLOGMSG\_C\_INST\_REL, 859  
   PVLOGMSG\_C\_NOTICE, 859  
   PVLOGMSG\_C\_STACK\_DEBUG, 859  
   PVLOGMSG\_C\_STACK\_TRACE, 859  
   PVLOGMSG\_C\_WARNING, 859  
 PVLOGGER\_C\_INST\_LEVEL  
   pvlogger\_c.h, 859  
 PVLOGGER\_ENABLE  
   pvlogger.h, 851  
 PVLOGGER\_FILTER\_ACCEPT  
   pvlogger\_accessories.h, 857  
 PVLOGGER\_FILTER\_NEUTRAL  
   pvlogger\_accessories.h, 857  
 PVLOGGER\_FILTER\_REJECT  
   pvlogger\_accessories.h, 857  
 pvLogger\_GetLoggerObject  
   pvlogger\_c.h, 859  
 PVLOGGER\_INST\_LEVEL  
   osclconfig.h, 803  
   pvlogger.h, 852  
 PVLOGGER\_INST\_LEVEL\_SUPPORT  
   pvlogger.h, 852  
 pvLogger\_IsActive  
   pvlogger\_c.h, 859  
 PVLOGGER\_LEVEL\_UNINITIALIZED  
   pvlogger.h, 855  
 PVLOGGER\_LOG\_USE\_ONLY  
   pvlogger.h, 852  
 PVLOGGER\_LOGBIN

pvlogger.h, 852  
**PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_-  
HLDBG**  
 pvlogger.h, 852  
**PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_-  
LLDBG**  
 pvlogger.h, 853  
**PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_-  
MLDBG**  
 pvlogger.h, 853  
**PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_-  
PROF**  
 pvlogger.h, 853  
**PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_-  
REL**  
 pvlogger.h, 853  
**PVLOGGER\_LOGBIN\_V**  
 pvlogger.h, 853  
**PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_-  
INST\_HLDBG**  
 pvlogger.h, 853  
**PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_-  
INST\_LLDBG**  
 pvlogger.h, 853  
**PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_-  
INST\_PROF**  
 pvlogger.h, 853  
**PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_-  
INST\_REL**  
 pvlogger.h, 853  
**PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_V\_-  
INST\_MLDBG**  
 pvlogger.h, 853  
**PVLOGGER\_LOGMSG**  
 pvlogger.h, 853  
**PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
INST\_HLDBG**  
 pvlogger.h, 853  
**PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
INST\_LLDBG**  
 pvlogger.h, 854  
**PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
INST\_MLDBG**  
 pvlogger.h, 854  
**PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
INST\_PROF**  
 pvlogger.h, 854  
**PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
INST\_REL**  
 pvlogger.h, 854  
**PVLOGGER\_LOGMSG\_V**  
 pvlogger.h, 854  
**PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
INST\_HLDBG**

pvlogger.h, 854  
**PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
INST\_LLDBG**  
 pvlogger.h, 854  
**PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
INST\_MLDBG**  
 pvlogger.h, 854  
**PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
INST\_PROF**  
 pvlogger.h, 854  
**PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
INST\_REL**  
 pvlogger.h, 854  
**PvLogger\_LogMsgString**  
 pvlogger\_c.h, 859  
**Pvlogger\_registry.h**, 860  
**PVLoggerAppender**, 623  
**PVLoggerAppender**  
 ~PVLoggerAppender, 623  
 AppendBuffers, 623  
 AppendString, 623  
 message\_id\_type, 623  
**PVLoggerFilter**, 624  
**PVLoggerFilter**  
 ~PVLoggerFilter, 625  
 filter\_status\_type, 624  
 FilterOpaqueMessge, 625  
 FilterString, 625  
 log\_level\_type, 624  
 message\_id\_type, 624  
**PVLoggerLayout**, 626  
**PVLoggerLayout**  
 ~PVLoggerLayout, 626  
 FormatOpaqueMessage, 626  
 FormatString, 626  
 message\_id\_type, 626  
**PVLoggerRegistry**, 628  
**PVLogger**, 622  
**PVLoggerRegistry**, 628  
**PVLoggerRegistry**  
 ~PVLoggerRegistry, 628  
 alloc\_type, 628  
 CreatePVLogger, 629  
 GetPVLoggerObject, 629  
 GetPVLoggerRegistry, 629  
 log\_level\_type, 628  
**PVLoggerRegistry**, 628  
 SetNodeLogLevelExplicit, 629  
**PVLOGMSG\_ALERT**  
 pvlogger.h, 855  
**PVLOGMSG\_C\_ALERT**  
 pvlogger\_c.h, 859  
**PVLOGMSG\_C\_CRIT**  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_EMERG  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_ERR  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_INFO  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_INST\_HLDBG  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_INST\_LLDBG  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_INST\_MLDBG  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_INST\_PROF  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_INST\_REL  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_NOTICE  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_STACK\_DEBUG  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_STACK\_TRACE  
 pvlogger\_c.h, 859

PVLOGMSG\_C\_WARNING  
 pvlogger\_c.h, 859

PVLOGMSG\_CRIT  
 pvlogger.h, 855

PVLOGMSG\_DEBUG  
 pvlogger.h, 855

PVLOGMSG\_EMERG  
 pvlogger.h, 855

PVLOGMSG\_ERR  
 pvlogger.h, 855

PVLOGMSG\_FATAL\_ERROR  
 pvlogger.h, 855

PVLOGMSG\_INFO  
 pvlogger.h, 856

PVLOGMSG\_INST\_HLDBG  
 pvlogger.h, 854

PVLOGMSG\_INST\_LLDBG  
 pvlogger.h, 854

PVLOGMSG\_INST\_MLDBG  
 pvlogger.h, 854

PVLOGMSG\_INST\_PROF  
 pvlogger.h, 855

PVLOGMSG\_INST\_REL  
 pvlogger.h, 855

PVLOGMSG\_NONFATAL\_ERROR  
 pvlogger.h, 856

PVLOGMSG\_NOTICE  
 pvlogger.h, 856

PVLOGMSG\_STACK\_TRACE  
 pvlogger.h, 856

PVLOGMSG\_STATISTIC  
 pvlogger.h, 856

PVLOGMSG\_VERBOSE  
 pvlogger.h, 856

PVLOGMSG\_WARNING  
 pvlogger.h, 856

PVMEM\_INST\_LEVEL  
 osclbase, 34

osclconfig\_memory.h, 827

PVNETHWORKADDRESS\_LEN  
 oscl\_socket\_types.h, 774

PVOscBase\_Cleanup  
 osclbase, 45

PVOscBase\_Init  
 osclbase, 45

PVSCHEDNAMELEN  
 osclproc, 105

PVSchedulerStopper, 631

OsclExecSchedulerCommonBase, 399

PVSchedulerStopper, 631

PVSchedulerStopper  
 ~PVSchedulerStopper, 631

PVSchedulerStopper, 631

PVSOCK\_ERR\_BAD\_PARAM  
 oscl\_socket\_imp\_pv.h, 757

PVSOCK\_ERR\_NOT\_IMPLEMENTED  
 oscl\_socket\_imp\_pv.h, 757

PVSOCK\_ERR\_NOT\_SUPPORTED  
 oscl\_socket\_imp\_pv.h, 757

PVSOCK\_ERR\_SERV\_NOT\_CONNECTED  
 oscl\_socket\_imp\_pv.h, 757

PVSOCK\_ERR\_SOCK\_NO\_SERV  
 oscl\_socket\_imp\_pv.h, 757

PVSOCK\_ERR\_SOCK\_NOT\_CONNECTED  
 oscl\_socket\_imp\_pv.h, 757

PVSOCK\_ERR SOCK NOT OPEN  
 oscl\_socket\_imp\_pv.h, 757

PVSockBufRecv, 632

PVSockBufRecv, 632

PVSockBufRecv  
 iLen, 632

iMaxLen, 632

iPtr, 632

PVSockBufRecv, 632

PVSockBufSend, 633

PVSockBufSend, 633

PVSockBufSend  
 iLen, 633

iPtr, 633

PVSockBufSend, 633

PVThreadContext, 634

OsclExecSchedulerCommonBase, 399

PVThreadContext, 634

PVThreadContext  
 ~PVThreadContext, 634

EnterThreadContext, 634

ExitThreadContext, 634  
 Id, 634  
 IsSameThreadContext, 634  
 OsclActiveObject, 635  
 OsclCoeActiveScheduler, 635  
 OsclCoeActiveSchedulerBase, 635  
 OsclExecScheduler, 635  
 OsclExecSchedulerBase, 635  
 OsclExecSchedulerCommonBase, 635  
 OsclTimerObject, 635  
 PVActiveBase, 635  
 PVThreadContext, 634  
 ThreadHasScheduler, 635

**QUE\_ITER\_BEGIN**  
 osclproc, 105

**QUE\_ITER\_END**  
 osclproc, 105

**Rand**  
 OsclRand, 481

**Read**  
 Oscl\_File, 183  
 OsclAsyncFile, 318  
 OsclBinIStreamBigEndian, 327  
 OsclFileCache, 402  
 OsclNativeFile, 463

**read**  
 OSCL\_String, 263  
 OSCL\_wString, 306

**Read\_uint16**  
 OsclBinIStreamBigEndian, 327  
 OsclBinIStreamLittleEndian, 330

**Read\_uint32**  
 OsclBinIStreamBigEndian, 327  
 OsclBinIStreamLittleEndian, 330

**Read\_uint8**  
 OsclBinIStream, 324

**ReadAsync**  
 OsclNativeFile, 463

**ReadAsyncCancel**  
 OsclNativeFile, 463

**rebalance**  
 Oscl\_Rb\_Tree\_Base, 246

**rebalance\_for\_erase**  
 Oscl\_Rb\_Tree\_Base, 246

**Recv**  
 OsclRecvMethod, 490  
 OsclRecvRequest, 491  
 OsclSocketI, 538  
 OsclSocketIBase, 543  
 OsclTCPSocket, 570  
 OsclTCPSocketI, 574

**RecvFrom**

OsclRecvFromMethod, 486  
 OsclRecvFromRequest, 488  
 OsclSocketI, 538  
 OsclSocketIBase, 543  
 OsclUDPSocket, 604  
 OsclUDPSocketI, 608  
 RecvFromParam, 636  
 RecvFromParam, 636  
 RecvFromParam  
   iAddr, 636  
   iBufRecv, 636  
   iFlags, 636  
   iMultiMaxLen, 636  
   iPacketLen, 636  
   iPacketSource, 636  
 RecvFromParam, 636

**RecvFromRequest**  
 OsclRecvFromMethod, 486

**RecvFromSuccess**  
 OsclSocketI, 538  
 OsclSocketIBase, 543

**RecvParam**, 638  
 RecvParam, 638

**RecvParam**  
   iBufRecv, 638  
   iFlags, 638  
 RecvParam, 638

**RecvRequest**  
 OsclRecvMethod, 490

**RecvSuccess**  
 OsclSocketI, 538  
 OsclSocketIBase, 543

**red**  
 Oscl\_Rb\_Tree\_Node\_Base, 254

**RedBl**  
 Oscl\_Rb\_Tree\_Node\_Base, 254

**refcount**  
 CHeapRep, 130

**reference**  
 Oscl\_Map, 218  
 Oscl\_Queue, 237  
 Oscl\_Rb\_Tree, 244  
 Oscl\_Rb\_Tree\_Const\_Iterator, 248  
 Oscl\_Rb\_Tree\_Iterator, 251  
 Oscl\_TagTree::const\_iterator, 274  
 Oscl\_TagTree::iterator, 277  
 Oscl\_TAlloc, 282  
 Oscl\_Vector, 286

**Register**  
 OsclComponentRegistry, 344  
 OsclRegistryClient, 509  
 OsclRegistryClientImpl, 512  
 OsclRegistryServTlsImpl, 515

**RegisterForCallback**

OsclExecScheduler, 389  
 OsclReadyQ, 485  
**registerInstance**  
 OsclSingletonRegistry, 534  
 OsclTLSRegistry, 596  
 OsclTLSRegistryEx, 597  
**registerInstanceAndUnlock**  
 OsclSingletonRegistry, 534  
**release**  
 OsclExclusiveArrayPtr, 382  
 OsclExclusivePtr, 385  
 OsclExclusivePtrA, 388  
 OSCLMemAutoPtr, 437  
**RELOCK\_MUTEX\_ERROR**  
 OsclProcStatus, 476  
**Remove**  
 OsclDoubleLink, 366  
 OsclReadyQ, 485  
 OsclSocketServRequestList, 560  
 OsclTimerQ, 591  
**remove**  
 OsclPriorityQueue, 472  
 OsclPriorityQueueBase, 474  
**remove\_element**  
 Oscl\_Linked\_List, 209  
 Oscl\_Linked\_List\_Base, 214  
 Oscl\_MTLLinked\_List, 227  
**remove\_ref**  
 CHheapRep, 130  
**removeALLAllocNodes**  
 MM\_Audit\_Imp, 159  
**removeAllocNode**  
 MM\_Audit\_Imp, 159  
**RemoveAppender**  
 PVLogger, 621  
**RemoveFixedCache**  
 Oscl\_File, 183  
**RemoveFromScheduler**  
 OsclActiveObject, 312  
 OsclTimerObject, 588  
 PVActiveBase, 614  
**RemoveRef**  
 DNSRequestParam, 134  
**removeRef**  
 Oscl\_DefAllocWithRefCounter, 174  
 OsclMemPoolFixedChunkAllocator, 445  
 OsclMemPoolResizableAllocator, 452  
 OsclRefCounter, 492  
 OsclRefCounterDA, 495  
 OsclRefCounterMTDA, 499  
 OsclRefCounterMTSA, 501  
 OsclRefCounterSA, 503  
**Request**  
 OsclTimer, 583  
**RequestCanceled**  
 OsclExecSchedulerCommonBase, 396  
**RequestDone**  
 OsclDNSRequestAO, 364  
 OsclSocketRequestAO, 552  
**reserve**  
 Oscl\_Queue\_Base, 240  
 Oscl\_Vector\_Base, 293  
 OsclPriorityQueue, 472  
**ReserveSpace**  
 OsclBinStream, 338  
**Reset**  
 OsclDoubleListBase, 369  
**reset**  
 BufferState, 119  
 MM\_FailInsertParam, 162  
 MM\_Stats\_t, 165  
 OsclMemStatsNode, 458  
**ResetLogPerf**  
 OsclExecSchedulerCommonBase, 396  
**Resume**  
 OsclThread, 577  
**ResumeScheduler**  
 OsclExecSchedulerCommonBase, 396  
**retrieveParentTag**  
 MM\_Audit\_Imp, 159  
**retrieveParentTagLength**  
 MM\_Audit\_Imp, 159  
**RFC822ToPV8601**  
 osclbase, 45  
**Right**  
 OsclPtrC, 480  
**right**  
 Oscl\_Rb\_Tree\_Node\_Base, 255  
**rotate\_left**  
 Oscl\_Rb\_Tree\_Base, 246  
**rotate\_right**  
 Oscl\_Rb\_Tree\_Base, 246  
**Run**  
 CallbackTimer, 124  
 OsclDNSMethod, 359  
 OsclDNSRequestAO, 364  
 OsclSocketMethod, 546  
 OsclSocketRequestAO, 552  
 PVActiveBase, 614  
**RunError**  
 OsclActiveObject, 312  
 OsclTimerObject, 588  
 PVActiveBase, 614  
**RunIfNotReady**  
 OsclActiveObject, 313  
 OsclTimerObject, 588  
**RunSchedulerNonBlocking**  
 OsclExecScheduler, 389

save\_registry  
     TLSStorageOps, 656  
 second  
     Oscl\_Pair, 235  
 SECONDS  
     osclbase, 35  
 Seed  
     OsclRand, 481  
 Seek  
     Oscl\_File, 183  
     OsclAsyncFile, 318  
     OsclBinStream, 338  
     OsclFileCache, 402  
     OsclNativeFile, 464  
 seek\_type  
     Oscl\_File, 180  
 SEEKCUR  
     Oscl\_File, 180  
 SEEKEND  
     Oscl\_File, 180  
 seekFromCurrentPosition  
     OsclBinStream, 338  
 SEEKSET  
     Oscl\_File, 180  
 self  
     Oscl\_Map, 218  
     Oscl\_Rb\_Tree\_Const\_Iterator, 248  
     Oscl\_Rb\_Tree\_Iterator, 251  
     Oscl\_TagTree::const\_iterator, 274  
     Oscl\_TagTree::iterator, 277  
 SEM\_NOT\_SIGNALLED\_ERROR  
     OsclProcStatus, 476  
 Send  
     OsclSendMethod, 523  
     OsclSendRequest, 524  
     OsclSocketI, 538  
     OsclSocketIBase, 543  
     OsclTCPSocket, 570  
     OsclTCPSocketI, 574  
 SendParam, 639  
     SendParam, 639  
 SendParam  
     iBufSend, 639  
     iFlags, 639  
     iXferLen, 639  
     SendParam, 639  
 SendRequest  
     OsclSendMethod, 523  
 SendSuccess  
     OsclSocketI, 538  
     OsclSocketIBase, 543  
 SendTo  
     OsclSendToMethod, 525  
     OsclSendToRequest, 526  
     OsclSocketI, 538  
     OsclSocketIBase, 543  
     SendToParam, 640  
     SendToParam, 640  
     SendToParam  
         ~SendToParam, 640  
         iAddr, 640  
         iBufSend, 640  
         iFlags, 640  
         iXferLen, 640  
         SendToParam, 640  
     SendToRequest  
         OsclSendToMethod, 525  
 SendToSuccess  
     OsclSocketI, 538  
     OsclSocketIBase, 543  
 Serv  
     OsclDNSRequestAO, 365  
 Set  
     OsclDoubleRunner, 370  
     OsclNameString, 461  
     OsclPtr, 477  
     OsclPtrC, 480  
 set  
     CHHeapRep, 130  
     CStackRep, 132  
     OSCL\_FastString, 177, 178  
     OSCL\_HeapStringA, 201, 202  
     OSCL\_wFastString, 296  
     OSCL\_wHeapStringA, 301  
     OsclExclusiveArrayPtr, 382  
     OsclExclusivePtr, 385  
     OsclExclusivePtrA, 388  
     OsclSingleton, 532  
     OsclTLS, 592  
     OsclTLSEx, 594  
     osclutil, 82–84  
     set\_from\_ntp\_time  
         TimeValue, 654  
     set\_from\_system\_time  
         NTPTime, 169  
     set\_int64  
         Oscl\_Int64\_Utils, 204  
     set\_len  
         OSCL\_String, 263  
         OSCL\_wString, 307  
     set\_length  
         OSCL\_FastString, 178  
         OSCL\_wFastString, 296  
     set\_next  
         Oscl\_Opaque\_Type\_Alloc\_LL, 232  
     set\_r

CFastRep, 128  
 set\_rep  
   CHheapRep, 130  
   OSCL\_String, 263, 264  
   OSCL\_wString, 307  
 set\_to\_current\_time  
   NTPTime, 169  
   TimeValue, 654  
 set\_to\_zero  
   TimeValue, 655  
 set\_uint64  
   Oscl\_Int64\_Utils, 204  
 set\_w  
   CFastRep, 128  
 set\_zulu  
   TimeValue, 655  
 setAllocNodeFlag  
   MM\_AllocBlockHdr, 148  
 SetAsyncReadBufferSize  
   Oscl\_File, 183  
 SetBusy  
   OsclActiveObject, 313  
   OsclTimerObject, 588  
 SetCacheObserver  
   Oscl\_File, 184  
 setCheckSum  
   StrCSumPtrLen, 645  
 SetExactFrequency  
   OsclTimer, 583  
 SetFileHandle  
   Oscl\_File, 184  
 SetFrequency  
   OsclTimer, 584  
 SetInUse  
   OsclAsyncFileBuffer, 320  
 SetLength  
   OsclPtr, 477  
   OsclPtrC, 480  
 SetLoggingEnable  
   Oscl\_File, 184  
 SetLogLevel  
   PVLogger, 621  
 SetLogLevelAndPropagate  
   PVLogger, 622  
 setMaxSzForNewMemPoolBuffer  
   OsclMemPoolResizableAllocator, 452  
 SetMulticastTTL  
   OsclUDPSocket, 605  
   OsclUDPSocketI, 608  
 SetNativeAccessMode  
   Oscl\_File, 184  
 SetNativeBufferSize  
   Oscl\_File, 185  
 SetNodeLogLevelExplicit  
   PVLoggerRegistry, 629  
 SetObserver  
   OsclTimer, 584  
 SetOffset  
   OsclAsyncFileBuffer, 320  
   OsclDoubleListBase, 369  
 SetOptionToReuseAddress  
   OsclIPSocketI, 419  
   OsclTCPSocket, 570  
   OsclUDPSocket, 605  
 SetParent  
   PVLogger, 622  
 SetPosition  
   OsclFileCacheBuffer, 404  
 SetPrecedence  
   OsclSocketTOS, 564  
 SetPriority  
   OsclSocketTOS, 564  
   OsclThread, 577  
 setPtrLen  
   StrCSumPtrLen, 645  
   StrPtrLen, 648  
   WStrPtrLen, 659  
 SetPVCacheSize  
   Oscl\_File, 185  
 SetRecvBufferSize  
   OsclIPSocketI, 419  
   OsclSocketI, 538  
   OsclUDPSocket, 606  
 setrep\_to\_char  
   OSCL\_String, 264  
 setrep\_to\_wide\_char  
   OSCL\_wString, 307  
 SetScheduler  
   OsclExecSchedulerCommonBase, 396  
 SetSize  
   Oscl\_File, 185  
   OsclNativeFile, 464  
 SetSockOpt  
   OsclSocketI, 539  
 SetStatus  
   OsclActiveObject, 313  
   OsclTimerObject, 588  
 SetSummaryStatsLoggingEnable  
   Oscl\_File, 185  
 SetTimestamp  
   MediaData, 144  
 SetToHead  
   OsclDoubleRunner, 370  
 SetTOS  
   OsclIPSocketI, 419  
   OsclTCPSocket, 571  
   OsclUDPSocket, 606  
 SetToTail

OsclDoubleRunner, 370  
 setWithoutOwnership  
     OSCLMemAutoPtr, 437  
 ShowStats  
     OsclExecSchedulerCommonBase, 396  
 ShowSummaryStats  
     OsclExecSchedulerCommonBase, 396  
 Shutdown  
     OsclShutdownMethod, 530  
     OsclShutdownRequest, 531  
     OsclSocketI, 539  
     OsclSocketIBase, 544  
     OsclTCPSocket, 571  
     OsclTCPSocketI, 574  
 ShutdownParam, 641  
     ShutdownParam, 641  
 ShutdownParam  
     iHow, 641  
     ShutdownParam, 641  
 ShutdownRequest  
     OsclShutdownMethod, 530  
 Signal  
     OsclSemaphore, 522  
 Size  
     Oscl\_File, 185  
     OsclAsyncFile, 318  
     OsclNativeFile, 464  
 size  
     CFastRep, 128  
     CHheapRep, 130  
     CStackRep, 132  
     MM\_AllocBlockHdr, 148  
     MM\_AllocInfo, 150  
     MM\_AllocQueryInfo, 152  
     Oscl\_Map, 221  
     Oscl\_Queue\_Base, 240  
     Oscl\_Rb\_Tree, 244  
     Oscl\_TagTree, 272  
     Oscl\_Vector\_Base, 293  
     OsclPriorityQueue, 472  
     StrPtrLen, 648  
     WStrPtrLen, 659  
 size\_type  
     Oscl\_Map, 218  
     Oscl\_Queue, 237  
     Oscl\_Rb\_Tree, 244  
     Oscl\_Tag\_Base, 268  
     Oscl\_TagTree, 270  
     Oscl\_TAlloc, 282  
 sizeof\_T  
     Oscl\_Linked\_List\_Base, 215  
     Oscl\_Queue\_Base, 241  
     Oscl\_Vector\_Base, 293  
 skip\_to\_line\_term  
     osclutil, 84  
     skip\_to whitespace  
         osclutil, 84  
     skip\_whitespace  
         osclutil, 84  
     skip\_whitespace\_and\_line\_term  
         osclutil, 84  
     SLEEP\_ONE\_SEC  
         osclconfig\_util.h, 847  
     SleepMillisec  
         OsclThread, 577  
     Socket  
         OsclSocketI, 539  
     SocketI  
         OsclSocketRequestAO, 552  
     SocketObserver  
         OsclSocketRequestAO, 552  
     SocketRequestParam, 642  
         SocketRequestParam, 643  
     SocketRequestParam  
         iFxn, 643  
         SocketRequestParam, 643  
     SocketServ  
         OsclIPSocketI, 419  
     sort\_children  
         Oscl\_TagTree::Node, 280  
     specialFragBuffer  
         OsclBinStream, 339  
     Start  
         OsclFileStats, 411  
     Start\_on\_creation  
         oscl\_thread.h, 788  
     StartAsyncRead  
         OsclAsyncFileBuffer, 320  
     StartCancel  
         OsclSocketServRequestList, 560  
     StartMethod  
         OsclDNSMethod, 359  
         OsclSocketMethod, 547  
     StartNativeScheduler  
         OsclExecSchedulerCommonBase, 396  
     StartScheduler  
         OsclExecSchedulerCommonBase, 396  
     State  
         OsclSocketServIBase, 559  
     state  
         OsclBinStream, 339  
     state\_t  
         OsclBinStream, 337  
     StaticJump  
         OsclJump, 421  
     stats\_overhead  
         MM\_AuditOverheadStats, 161  
     Status

OsclActiveObject, 313  
 OsclTimerObject, 589  
**status\_t**  
 BufFragStatusClass, 123  
**StatusRef**  
 OsclActiveObject, 313  
 OsclTimerObject, 589  
**StopScheduler**  
 OsclExecSchedulerCommonBase, 396  
**Str**  
 OsclNameString, 461  
**StrCSumPtrLen**, 644  
 osclutil, 69  
 StrCSumPtrLen, 645  
**StrCSumPtrLen**  
 checkSum, 645  
 CheckSumType, 645  
 getCheckSum, 645  
 isCIEquivalentTo, 645  
 operator!=, 645  
 operator=, 645  
 operator==, 645  
 setCheckSum, 645  
 setPtrLen, 645  
 StrCSumPtrLen, 645  
**StrPtrLen**, 647  
 osclutil, 69  
 StrPtrLen, 648  
**StrPtrLen**  
 c\_str, 648  
 isCIEquivalentTo, 648  
 isCIPrefixOf, 648  
 isLetter, 648  
 len, 648  
 length, 648  
 operator!=, 648  
 operator=, 648  
 operator==, 648  
 ptr, 648  
 setPtrLen, 648  
 size, 648  
 StrPtrLen, 648  
**Success**  
 OsclDNSRequestAO, 365  
 OsclRecvFromRequest, 488  
 OsclRecvRequest, 491  
 OsclSendRequest, 524  
 OsclSendToRequest, 526  
 OsclSocketRequestAO, 552  
**SUCCESS\_ERROR**  
 OsclProcStatus, 475  
**Suspend**  
 OsclThread, 578  
**Suspend\_on\_creation**  
 oscl\_thread.h, 788  
**SuspendScheduler**  
 OsclExecSchedulerCommonBase, 397  
**swap**  
 Oscl\_Opaque\_Type\_Compare, 233  
 OsclPriorityQueue, 472  
**SYSTEM\_RESOURCES\_UNAVAILABLE\_-ERROR**  
 OsclProcStatus, 476  
**tag**  
 MM\_AllocQueryInfo, 152  
 MM\_Stats\_CB, 163  
 Oscl\_Tag, 265  
 Oscl\_TagTree::Node, 280  
 OsclMemStatsNode, 458  
**tag\_ancestor**  
 Oscl\_Tag\_Base, 268  
**tag\_base\_type**  
 Oscl\_Tag\_Base, 268  
 Oscl\_TagTree, 270  
**tag\_base\_unit**  
 Oscl\_Tag\_Base, 268  
**tag\_cmp**  
 Oscl\_Tag\_Base, 268  
**tag\_copy**  
 Oscl\_Tag\_Base, 268  
**tag\_depth**  
 Oscl\_Tag\_Base, 268  
**tag\_len**  
 Oscl\_Tag\_Base, 268  
**tag\_type**  
 Oscl\_TagTree, 270  
**tagAllocator**  
 Oscl\_Tag, 265  
**TagTree\_Allocator**  
 osclmemory, 59  
**Tail**  
 OsclDoubleList, 367  
 OsclPriorityList, 469  
**tail**  
 Oscl\_Linked\_List\_Base, 215  
**takeOwnership**  
 OSCLMemAutoPtr, 438  
**TDNSRequestParamAllocator**  
 oscl\_dns\_param.h, 675  
**Tell**  
 Oscl\_File, 185  
 OsclAsyncFile, 318  
 OsclFileCache, 402  
 OsclNativeFile, 464  
**tellg**  
 OsclBinStream, 338  
**Terminate**

OsclThread, [578](#)  
 the\_list  
   Oscl\_MTLinked\_List, [227](#)  
 THREAD\_1\_INACTIVE\_ERROR  
   OsclProcStatus, [475](#)  
 THREAD\_BLOCK\_ERROR  
   OsclProcStatus, [476](#)  
 THREAD\_NOT\_OWN\_MUTEX\_ERROR  
   OsclProcStatus, [476](#)  
 ThreadHasScheduler  
   PVThreadContext, [635](#)  
 ThreadLogoff  
   OsclIPSocketI, [419](#)  
   OsclReadyQ, [485](#)  
   OsclSocketI, [539](#)  
   OsclSocketMethod, [547](#)  
   OsclTCPSocket, [571](#)  
   OsclTCPSocketI, [574](#)  
   OsclUDPSocket, [606](#)  
   OsclUDPSocketI, [608](#)  
 ThreadLogon  
   OsclIPSocketI, [419](#)  
   OsclReadyQ, [485](#)  
   OsclSocketI, [539](#)  
   OsclSocketMethod, [547](#)  
   OsclTCPSocket, [571](#)  
   OsclTCPSocketI, [574](#)  
   OsclUDPSocket, [606](#)  
   OsclUDPSocketI, [608](#)  
 ThreadPriorityAboveNormal  
   oscl\_thread.h, [789](#)  
 ThreadPriorityBelowNormal  
   oscl\_thread.h, [789](#)  
 ThreadPriorityHighest  
   oscl\_thread.h, [789](#)  
 ThreadPriorityLow  
   oscl\_thread.h, [788](#)  
 ThreadPriorityLowest  
   oscl\_thread.h, [788](#)  
 ThreadPriorityNormal  
   oscl\_thread.h, [789](#)  
 ThreadPriorityTimeCritical  
   oscl\_thread.h, [789](#)  
 TickCount  
   OsclTickCount, [580](#)  
 TickCountFrequency  
   OsclTickCount, [580](#)  
 TickCountPeriod  
   OsclTickCount, [580](#)  
 TicksToMsec  
   OsclTickCount, [580](#)  
 TimeoutOccurred  
   OsclTimerObserver, [590](#)  
 TimerBaseElapsed  
   CallbackTimerObserver, [126](#)  
   OsclTimer, [584](#)  
 TimerCallback  
   OsclReadyQ, [485](#)  
 timestamp  
   MediaData, [144](#)  
 TimeUnits  
   osclbase, [35](#)  
 TimeValue, [649](#)  
   TimeValue, [651, 652](#)  
 TimeValue  
   get\_ISO8601\_str\_time, [652](#)  
   get\_local\_time, [652](#)  
   get\_pv8601\_str\_time, [652](#)  
   get\_rfc822\_gmtime\_str, [652](#)  
   get\_sec, [653](#)  
   get\_str\_ctime, [653](#)  
   get\_timeval\_ptr, [653](#)  
   get\_timevalue\_in\_usec, [653](#)  
   get\_usec, [653](#)  
   is\_zero, [654](#)  
   is\_zulu, [654](#)  
   NTPTime, [655](#)  
   operator \*=, [654](#)  
   operator!=, [655](#)  
   operator+=, [654](#)  
   operator-=, [654](#)  
   operator<, [655](#)  
   operator<=, [655](#)  
   operator=, [654](#)  
   operator==, [655](#)  
   operator>, [655](#)  
   operator>=, [655](#)  
   set\_from\_ntp\_time, [654](#)  
   set\_to\_current\_time, [654](#)  
   set\_to\_zero, [655](#)  
   set\_zulu, [655](#)  
   TimeValue, [651, 652](#)  
   to\_msec, [655](#)  
 TIpMReq  
   osclconfig\_io.h, [821](#)  
 TLSStorageOps, [656](#)  
 TLSStorageOps  
   get\_registry, [656](#)  
   save\_registry, [656](#)  
 to\_msec  
   TimeValue, [655](#)  
 to\_system\_time  
   NTPTime, [169](#)  
 TOO\_MANY\_FRAGS  
   BuffFragStatusClass, [123](#)  
 TOO\_MANY\_THREADS\_ERROR  
   OsclProcStatus, [475](#)  
 Top

OsclJump, [421](#)  
 OsclReadyQ, [485](#)  
 OsclTimerQ, [591](#)  
**top**  
 OsclPriorityQueue, [472](#)  
**TOSCL\_StringOp**  
 osclutil, [70](#)  
**TOSCL\_wStringOp**  
 osclutil, [70](#)  
**TOsclBasicLockObject**  
 osclconfig\_unix\_android.h, [842](#)  
 osclconfig\_unix\_common.h, [846](#)  
**TOsclConditionObject**  
 osclconfig\_proc\_unix\_android.h, [834](#)  
 osclconfig\_proc\_unix\_common.h, [836](#)  
**TOsclFileHandle**  
 osclio, [97](#)  
**TOsclFileOffset**  
 osclconfig\_io.h, [821](#)  
**TOsclFileOffsetInt32**  
 osclio, [97](#)  
**TOsclFileOp**  
 osclio, [98](#)  
**TOsclHostent**  
 osclconfig\_io.h, [821](#)  
**TOsclMutexObject**  
 osclconfig\_proc\_unix\_android.h, [834](#)  
 osclconfig\_proc\_unix\_common.h, [836](#)  
**TOsclReady**  
 osclproc, [106](#)  
**TOsclSemaphoreObject**  
 osclconfig\_proc\_unix\_android.h, [834](#)  
 osclconfig\_proc\_unix\_common.h, [836](#)  
**TOsclSockAddr**  
 osclconfig\_io.h, [821](#)  
**TOsclSockAddrLen**  
 osclconfig\_io.h, [821](#)  
**TOsclSocket**  
 osclconfig\_io.h, [821](#)  
**TOsclSocketServStatEvent**  
 oscl\_socket\_stats.h, [770](#)  
**TOsclSocketStatEvent**  
 oscl\_socket\_stats.h, [770](#)  
**TOsclThreadFuncArg**  
 osclconfig\_proc\_unix\_android.h, [834](#)  
 osclconfig\_proc\_unix\_common.h, [836](#)  
**TOsclThreadFuncPtr**  
 oscl\_thread.h, [788](#)  
**TOsclThreadFuncRet**  
 osclconfig\_proc\_unix\_android.h, [834](#)  
 osclconfig\_proc\_unix\_common.h, [836](#)  
**TOsclThreadId**  
 osclconfig\_proc\_unix\_android.h, [834](#)  
 osclconfig\_proc\_unix\_common.h, [836](#)  
**TOsclThreadObject**  
 osclconfig\_proc\_unix\_android.h, [834](#)  
 osclconfig\_proc\_unix\_common.h, [836](#)  
**TOsclThreadTerminate**  
 oscl\_thread.h, [789](#)  
**TOsclTlsKey**  
 osclbase, [35](#)  
 osclconfig\_unix\_android.h, [842](#)  
 osclconfig\_unix\_common.h, [846](#)  
**totalbytes**  
 oscl\_fsstat, [195](#)  
**totalNumAllocs**  
 MM\_Stats\_t, [165](#)  
**totalNumBytes**  
 MM\_Stats\_t, [165](#)  
**TOtherExecStats**  
 OsclExecSchedulerCommonBase, [394](#)  
**TPVDNSEvent**  
 osclio, [99](#)  
**TPVDNSFxn**  
 osclio, [99](#)  
**TPVServicePrecedence**  
 OsclSocketTOS, [563](#)  
**TPVServicePriority**  
 OsclSocketTOS, [563](#)  
**TPVSocketEvent**  
 oscl\_socket\_types.h, [774](#)  
**TPVSocketFxn**  
 oscl\_socket\_types.h, [775](#)  
**TPVSocketOptionLevel**  
 oscl\_socket\_types.h, [775](#)  
**TPVSocketOptionName**  
 oscl\_socket\_types.h, [775](#)  
**TPVSocketShutdown**  
 oscl\_socket\_types.h, [775](#)  
**TPVThreadContext**  
 osclproc, [106](#)  
**Trap**  
 OsclErrorTrapImp, [377](#)  
**TrapNoTls**  
 OsclErrorTrapImp, [377](#)  
**TReadyQueLink**, [657](#)  
 TReadyQueLink, [657](#)  
**TReadyQueLink**  
 iAOPriority, [657](#)  
 iIsIn, [657](#)  
 iSeqNum, [657](#)  
 iTimeQueuedTicks, [657](#)  
 iTimeToRunTicks, [657](#)  
 TReadyQueLink, [657](#)  
**trim**  
 OsclMemPoolResizableAllocator, [452](#)  
**TryLock**  
 OsclMutex, [460](#)

TryWait  
     OsclSemaphore, 522

TSocketServState  
     OsclSocketServIBase, 558

TSymbianAccessMode  
     Oscl\_File, 180

uint  
     osclbase, 35

UINT64  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846

uint64  
     osclbase, 35

UINT64\_HILO  
     osclconfig\_unix\_android.h, 842  
     osclconfig\_unix\_common.h, 846

Unbind  
     OsclSharedPtr, 529

UninstallScheduler  
     OsclExecSchedulerCommonBase, 397

unix\_ntp\_offset  
     osclbase, 46

Unlock  
     OsclLockBase, 424  
     OsclMutex, 460  
     OsclNullLock, 467  
     OsclThreadLock, 579

UnRegister  
     OsclRegistryClient, 510  
     OsclRegistryClientImpl, 512  
     OsclRegistryServTlsImpl, 515

Unregister  
     OsclComponentRegistry, 344

UnTrap  
     OsclErrorTrapImp, 377

update  
     MM\_Stats\_t, 165

UpdateData  
     OsclAsyncFileBuffer, 320

updateEnd  
     OsclFileCacheBuffer, 404

updateStart  
     OsclFileCacheBuffer, 404

updateStatsNode  
     MM\_Audit\_Imp, 159

updateStatsNodeInFailure  
     MM\_Audit\_Imp, 159

UpdateTimers  
     OsclExecSchedulerCommonBase, 397

UpdateTimersMsec  
     OsclExecSchedulerCommonBase, 397

upper\_bound  
     Oscl\_Map, 221, 222

Oscl\_Rb\_Tree, 244

usableSize  
     OsclFileCacheBuffer, 404

USEC\_PER\_SEC  
     osclbase, 46

validate  
     MM\_Audit\_Imp, 159  
     OsclPriorityQueue, 473

validate\_all\_heap  
     MM\_Audit\_Imp, 159

validateblock  
     OsclMemPoolResizableAllocator, 452

Value  
     OsclAOStatus, 315

value  
     Oscl\_Rb\_Tree\_Node, 253  
     Oscl\_TagTree::Node, 280

value\_comp  
     Oscl\_Map, 222

value\_compare  
     Oscl\_Map::value\_compare, 223

value\_type  
     Oscl\_Map, 218  
     Oscl\_Queue, 237  
     Oscl\_Rb\_Tree, 244  
     Oscl\_Rb\_Tree\_Const\_Iterator, 248  
     Oscl\_Rb\_Tree\_Iterator, 251  
     Oscl\_Rb\_Tree\_Node, 253  
     Oscl\_TagTree, 270  
     Oscl\_TAlloc, 282  
     Oscl\_Vector, 286  
     OsclPriorityQueue, 471

vec  
     OsclPriorityQueue, 473

Wait  
     OsclSemaphore, 522

WAIT\_ABANDONED\_ERROR  
     OsclProcStatus, 476

WAIT\_TIMEOUT\_ERROR  
     OsclProcStatus, 476

WaitAndPopTop  
     OsclReadyQ, 485

WaitForReadyAO  
     OsclExecSchedulerCommonBase, 397

WaitForRequestComplete  
     OsclReadyQ, 485

WaitOnRequests  
     OsclSocketServRequestList, 560

Wakeup  
     OsclSocketServRequestList, 560

writable  
     CFastRep, 128

Write

- Oscl\_File, [186](#)
- OsclAsyncFile, [318](#)
- OsclFileCache, [402](#)
- OsclNativeFile, [464](#)

write

- OSCL\_String, [264](#)
- OSCL\_wString, [307](#)
- OsclBinOStream, [331](#)

WriteUnsignedLong

- OsclBinOStreamBigEndian, [333](#)
- OsclBinOStreamLittleEndian, [335](#)

WriteUnsignedShort

- OsclBinOStreamBigEndian, [333](#)
- OsclBinOStreamLittleEndian, [335](#)

WriteUpdatesToFile

- OsclFileCacheBuffer, [404](#)

WStrPtrLen, [658](#)

- osclutil, [69](#)
- WStrPtrLen, [659](#)

WStrPtrLen

- c\_str, [659](#)
- isCIEquivalentTo, [659](#)
- len, [659](#)
- length, [659](#)
- operator!=, [659](#)
- operator=, [659](#)
- operator==, [659](#)
- ptr, [659](#)
- setPtrLen, [659](#)
- size, [659](#)
- WStrPtrLen, [659](#)

xsubi

- MM\_FailInsertParam, [162](#)

Zero

- OsclPtr, [477](#)
- OsclPtrC, [480](#)