



## OSCL API

Build Version: CORE\_8.509.1.3

April 9, 2010

# Contents

<b>1</b>	<b>oscl Module Index</b>	<b>1</b>
1.1	oscl Modules . . . . .	1
<b>2</b>	<b>oscl Hierarchical Index</b>	<b>2</b>
2.1	oscl Class Hierarchy . . . . .	2
<b>3</b>	<b>oscl Data Structure Index</b>	<b>9</b>
3.1	oscl Data Structures . . . . .	9
<b>4</b>	<b>oscl File Index</b>	<b>15</b>
4.1	oscl File List . . . . .	15
<b>5</b>	<b>oscl Page Index</b>	<b>20</b>
5.1	oscl Related Pages . . . . .	20
<b>6</b>	<b>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</b>	<b>oscl Data Structure Documentation</b>	<b>108</b>
7.1	_OscIBasicAllocator Class Reference . . . . .	108
7.2	_OscIHeapBase 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 BufFragGroup< ChainClass, max_frags > Class Template Reference . . . . .	120
7.11 BufFragStatusClass 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 CHeapRep 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 NTPTIME Class Reference . . . . .	166
7.38 Osci_Alloc Class Reference . . . . .	170
7.39 Osci_Dealloc Class Reference . . . . .	171
7.40 Osci_DefAlloc Class Reference . . . . .	172

7.41	<a href="#">OscL_DefAllocWithRefCount&lt; DefAlloc &gt; Class Template Reference</a>	173
7.42	<a href="#">OSCL_FastString Class Reference</a>	175
7.43	<a href="#">OscL_File Class Reference</a>	179
7.44	<a href="#">OscL_File::OscLCacheObserver Class Reference</a>	187
7.45	<a href="#">OscL_File::OscLFixedCacheParam Class Reference</a>	188
7.46	<a href="#">OscL_FileFind Class Reference</a>	189
7.47	<a href="#">OscL_FileServer Class Reference</a>	193
7.48	<a href="#">oscl_fsstat Struct Reference</a>	195
7.49	<a href="#">OSCL_HeapString&lt; Alloc &gt; Class Template Reference</a>	196
7.50	<a href="#">OSCL_HeapStringA Class Reference</a>	198
7.51	<a href="#">OscL_Int64_Utils Class Reference</a>	203
7.52	<a href="#">OscL_Less&lt; T &gt; Struct Template Reference</a>	205
7.53	<a href="#">OscL_Linked_List&lt; LLClass, Alloc &gt; Class Template Reference</a>	206
7.54	<a href="#">OscL_Linked_List_Base Class Reference</a>	211
7.55	<a href="#">OscL_Map&lt; Key, T, Alloc, Compare &gt; Class Template Reference</a>	216
7.56	<a href="#">OscL_Map&lt; Key, T, Alloc, Compare &gt;::value_compare Class Reference</a>	223
7.57	<a href="#">OscL_MTLinked_List&lt; LLClass, Alloc, TheLock &gt; Class Template Reference</a>	225
7.58	<a href="#">OscL_Opaque_Type_Alloc Class Reference</a>	229
7.59	<a href="#">OscL_Opaque_Type_Alloc_LL Class Reference</a>	231
7.60	<a href="#">OscL_Opaque_Type_Compare Class Reference</a>	233
7.61	<a href="#">OscL_Pair&lt; T1, T2 &gt; Struct Template Reference</a>	235
7.62	<a href="#">OscL_Queue&lt; T, Alloc &gt; Class Template Reference</a>	236
7.63	<a href="#">OscL_Queue_Base Class Reference</a>	239
7.64	<a href="#">OscL_Rb_Tree&lt; Key, Value, KeyOfValue, Compare, Alloc &gt; Class Template Reference</a>	242
7.65	<a href="#">OscL_Rb_Tree_Base Class Reference</a>	246
7.66	<a href="#">OscL_Rb_Tree_Const_Iterator&lt; Value &gt; Struct Template Reference</a>	247
7.67	<a href="#">OscL_Rb_Tree_Iterator&lt; Value &gt; Struct Template Reference</a>	250
7.68	<a href="#">OscL_Rb_Tree_Node&lt; Value &gt; Struct Template Reference</a>	253
7.69	<a href="#">OscL_Rb_Tree_Node_Base Struct Reference</a>	254
7.70	<a href="#">OscL_SelectIst&lt; V, U &gt; Struct Template Reference</a>	256
7.71	<a href="#">OSCL_StackString&lt; MaxBufSize &gt; Class Template Reference</a>	257
7.72	<a href="#">oscl_stat_buf Struct Reference</a>	259
7.73	<a href="#">OSCL_String Class Reference</a>	260
7.74	<a href="#">OscL_Tag&lt; Alloc &gt; Struct Template Reference</a>	265
7.75	<a href="#">OscL_Tag_Base Struct Reference</a>	267
7.76	<a href="#">OscL_TagTree&lt; T, Alloc &gt; Class Template Reference</a>	269

7.77 Osci_TagTree< T, Alloc >::const_iterator Struct Reference . . . . .	273
7.78 Osci_TagTree< T, Alloc >::iterator Struct Reference . . . . .	276
7.79 Osci_TagTree< T, Alloc >::Node Struct Reference . . . . .	279
7.80 Osci_TAlloc< T, Alloc > Class Template Reference . . . . .	281
7.81 Osci_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference . . . . .	284
7.82 Osci_Vector< T, Alloc > Class Template Reference . . . . .	285
7.83 Osci_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 OsciAcceptMethod Class Reference . . . . .	308
7.90 OsciAcceptRequest Class Reference . . . . .	309
7.91 OsciActiveObject Class Reference . . . . .	310
7.92 OsciAllocDestructDealloc Class Reference . . . . .	314
7.93 OsciAOStatus Class Reference . . . . .	315
7.94 OsciAsyncFile Class Reference . . . . .	316
7.95 OsciAsyncFileBuffer Class Reference . . . . .	319
7.96 OsciAuditCB Class Reference . . . . .	321
7.97 OsciBindMethod Class Reference . . . . .	322
7.98 OsciBindRequest Class Reference . . . . .	323
7.99 OsciBinIStream Class Reference . . . . .	324
7.100OsciBinIStreamBigEndian Class Reference . . . . .	326
7.101OsciBinIStreamLittleEndian Class Reference . . . . .	329
7.102OsciBinOStream Class Reference . . . . .	331
7.103OsciBinOStreamBigEndian Class Reference . . . . .	332
7.104OsciBinOStreamLittleEndian Class Reference . . . . .	334
7.105OsciBinStream Class Reference . . . . .	336
7.106OsciBuf Class Reference . . . . .	340
7.107OsciCompareLess< T > Class Template Reference . . . . .	342
7.108OsciComponentRegistry Class Reference . . . . .	343
7.109OsciComponentRegistryData Class Reference . . . . .	345
7.110OsciComponentRegistryElement Class Reference . . . . .	346
7.111OsciConnectMethod Class Reference . . . . .	348
7.112OsciConnectRequest Class Reference . . . . .	349

7.113OsciDestructDealloc Class Reference . . . . .	350
7.114OsciDNS Class Reference . . . . .	351
7.115OsciDNSI Class Reference . . . . .	353
7.116OsciDNSIBase Class Reference . . . . .	355
7.117OsciDNSMethod Class Reference . . . . .	358
7.118OsciDNSObserver Class Reference . . . . .	361
7.119OsciDNSRequest Class Reference . . . . .	362
7.120OsciDNSRequestAO Class Reference . . . . .	363
7.121OsciDoubleLink Class Reference . . . . .	366
7.122OsciDoubleList< T > Class Template Reference . . . . .	367
7.123OsciDoubleListBase Class Reference . . . . .	368
7.124OsciDoubleRunner< T > Class Template Reference . . . . .	370
7.125OsciError Class Reference . . . . .	372
7.126OsciErrorAllocator Class Reference . . . . .	374
7.127OsciErrorTrap Class Reference . . . . .	376
7.128OsciErrorTrapImp Class Reference . . . . .	377
7.129OsciException< LeaveCode > Class Template Reference . . . . .	379
7.130OsciExclusiveArrayPtr< T > Class Template Reference . . . . .	380
7.131OsciExclusivePtr< T > Class Template Reference . . . . .	383
7.132OsciExclusivePtrA< T, Alloc > Class Template Reference . . . . .	386
7.133OsciExecScheduler Class Reference . . . . .	389
7.134OsciExecSchedulerBase Class Reference . . . . .	391
7.135OsciExecSchedulerCommonBase Class Reference . . . . .	392
7.136OsciFileCache Class Reference . . . . .	401
7.137OsciFileCacheBuffer Class Reference . . . . .	403
7.138OsciFileHandle Class Reference . . . . .	405
7.139OsciFileManager Class Reference . . . . .	406
7.140OsciFileStats Class Reference . . . . .	411
7.141OsciFileStatsItem Class Reference . . . . .	412
7.142OsciGetHostByNameMethod Class Reference . . . . .	413
7.143OsciGetHostByNameRequest Class Reference . . . . .	414
7.144OsciInit Class Reference . . . . .	415
7.145OsciInteger64Transport Struct Reference . . . . .	416
7.146OsciIpMReq Class Reference . . . . .	417
7.147OsciIPSocketI Class Reference . . . . .	418
7.148OsciJump Class Reference . . . . .	421

7.149OsciListenMethod Class Reference . . . . .	422
7.150OsciListenRequest Class Reference . . . . .	423
7.151OsciLockBase Class Reference . . . . .	424
7.152OsciMem Class Reference . . . . .	425
7.153OsciMemAllocator Class Reference . . . . .	426
7.154OsciMemAllocDestructDealloc< T > Class Template Reference . . . . .	427
7.155OsciMemAudit Class Reference . . . . .	429
7.156OSCLMemAutoPtr< T, _Allocator > Class Template Reference . . . . .	435
7.157OsciMemBasicAllocator Class Reference . . . . .	439
7.158OsciMemBasicAllocDestructDealloc< T > Class Template Reference . . . . .	440
7.159OsciMemGlobalAuditObject Class Reference . . . . .	441
7.160OsciMemoryFragment Struct Reference . . . . .	442
7.161OsciMemPoolFixedChunkAllocator Class Reference . . . . .	443
7.162OsciMemPoolFixedChunkAllocatorObserver Class Reference . . . . .	447
7.163OsciMemPoolResizableAllocator Class Reference . . . . .	448
7.164OsciMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference . . . . .	454
7.165OsciMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference . . . . .	455
7.166OsciMemPoolResizableAllocatorMemoryObserver Class Reference . . . . .	456
7.167OsciMemPoolResizableAllocatorObserver Class Reference . . . . .	457
7.168OsciMemStatsNode Class Reference . . . . .	458
7.169OsciMutex Class Reference . . . . .	459
7.170OsciNameString< __len > Class Template Reference . . . . .	461
7.171OsciNativeFile Class Reference . . . . .	462
7.172OsciNativeFileParams Class Reference . . . . .	465
7.173OsciNetworkAddress Class Reference . . . . .	466
7.174OsciNullLock Class Reference . . . . .	467
7.175OsciPriorityLink Class Reference . . . . .	468
7.176OsciPriorityList< T > Class Template Reference . . . . .	469
7.177OsciPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference . . . . .	470
7.178OsciPriorityQueueBase Class Reference . . . . .	474
7.179OsciProcStatus Class Reference . . . . .	475
7.180OsciPtr Class Reference . . . . .	477
7.181OsciPtrC Class Reference . . . . .	479
7.182OsciRand Class Reference . . . . .	481
7.183OsciReadyAlloc Class Reference . . . . .	482
7.184OsciReadyCompare Class Reference . . . . .	483

7.185OsciReadyQ Class Reference . . . . .	484
7.186OsciRecvFromMethod Class Reference . . . . .	486
7.187OsciRecvFromRequest Class Reference . . . . .	488
7.188OsciRecvMethod Class Reference . . . . .	490
7.189OsciRecvRequest Class Reference . . . . .	491
7.190OsciRefCountCounter Class Reference . . . . .	492
7.191OsciRefCountCounterDA Class Reference . . . . .	494
7.192OsciRefCountCounterMemFrag Class Reference . . . . .	496
7.193OsciRefCountCounterMTDA< LockType > Class Template Reference . . . . .	498
7.194OsciRefCountCounterMTSA< DeallocType, LockType > Class Template Reference . . . . .	500
7.195OsciRefCountCounterSA< DeallocType > Class Template Reference . . . . .	502
7.196OsciRegistryAccessClient Class Reference . . . . .	504
7.197OsciRegistryAccessClientImpl Class Reference . . . . .	506
7.198OsciRegistryAccessClientTlsImpl Class Reference . . . . .	507
7.199OsciRegistryAccessElement Class Reference . . . . .	508
7.200OsciRegistryClient Class Reference . . . . .	509
7.201OsciRegistryClientImpl Class Reference . . . . .	511
7.202OsciRegistryClientTlsImpl Class Reference . . . . .	513
7.203OsciRegistryServTlsImpl Class Reference . . . . .	514
7.204OsciScheduler Class Reference . . . . .	516
7.205OsciSchedulerObserver Class Reference . . . . .	517
7.206OsciScopedLock< LockClass > Class Template Reference . . . . .	518
7.207OsciSelect Class Reference . . . . .	519
7.208OsciSemaphore Class Reference . . . . .	521
7.209OsciSendMethod Class Reference . . . . .	523
7.210OsciSendRequest Class Reference . . . . .	524
7.211OsciSendToMethod Class Reference . . . . .	525
7.212OsciSendToRequest Class Reference . . . . .	526
7.213OsciSharedPtr< TheClass > Class Template Reference . . . . .	527
7.214OsciShutdownMethod Class Reference . . . . .	530
7.215OsciShutdownRequest Class Reference . . . . .	531
7.216OsciSingleton< T, ID, Registry > Class Template Reference . . . . .	532
7.217OsciSingletonRegistry Class Reference . . . . .	534
7.218OsciSocketI Class Reference . . . . .	535
7.219OsciSocketIBase Class Reference . . . . .	540
7.220OsciSocketMethod Class Reference . . . . .	545



7.221OscISocketObserver Class Reference . . . . .	548
7.222OscISocketRequest Class Reference . . . . .	549
7.223OscISocketRequestAO Class Reference . . . . .	550
7.224OscISocketServ Class Reference . . . . .	554
7.225OscISocketServI Class Reference . . . . .	556
7.226OscISocketServIBase Class Reference . . . . .	558
7.227OscISocketServRequestList Class Reference . . . . .	560
7.228OscISocketServRequestQElem Class Reference . . . . .	562
7.229OscISocketTOS Class Reference . . . . .	563
7.230OscITCPSocket Class Reference . . . . .	565
7.231OscITCPSocketI Class Reference . . . . .	572
7.232OscIThread Class Reference . . . . .	575
7.233OscIThreadLock Class Reference . . . . .	579
7.234OscITickCount Class Reference . . . . .	580
7.235OscITimer< Alloc > Class Template Reference . . . . .	582
7.236OscITimerCompare Class Reference . . . . .	585
7.237OscITimerObject Class Reference . . . . .	586
7.238OscITimerObserver Class Reference . . . . .	590
7.239OscITimerQ Class Reference . . . . .	591
7.240OscITLS< T, ID, Registry > Class Template Reference . . . . .	592
7.241OscITLSEx< T, ID, Registry > Class Template Reference . . . . .	594
7.242OscITLSRegistry Class Reference . . . . .	596
7.243OscITLSRegistryEx Class Reference . . . . .	597
7.244OscITrapItem Class Reference . . . . .	598
7.245OscITrapStack Class Reference . . . . .	599
7.246OscITrapStackItem Class Reference . . . . .	600
7.247OscIUDPSocket Class Reference . . . . .	601
7.248OscIUDPSocketI Class Reference . . . . .	607
7.249OscIUuid Struct Reference . . . . .	610
7.250PVActiveBase Class Reference . . . . .	612
7.251PVActiveStats Class Reference . . . . .	616
7.252PVLogger Class Reference . . . . .	617
7.253PVLoggerAppender Class Reference . . . . .	623
7.254PVLoggerFilter Class Reference . . . . .	624
7.255PVLoggerLayout Class Reference . . . . .	626
7.256PVLoggerRegistry 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_gethostbyname.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	<a href="#">oscl_errno.h File Reference</a>	679
8.21	<a href="#">oscl_error.h File Reference</a>	680
8.22	<a href="#">oscl_error_allocator.h File Reference</a>	681
8.23	<a href="#">oscl_error_codes.h File Reference</a>	682
8.24	<a href="#">oscl_error_imp.h File Reference</a>	683
8.25	<a href="#">oscl_error_imp_cppexceptions.h File Reference</a>	684
8.26	<a href="#">oscl_error_imp_fatalerror.h File Reference</a>	685
8.27	<a href="#">oscl_error_imp_jumps.h File Reference</a>	686
8.28	<a href="#">oscl_error_trapcleanup.h File Reference</a>	688
8.29	<a href="#">oscl_exception.h File Reference</a>	689
8.30	<a href="#">oscl_exclusive_ptr.h File Reference</a>	690
8.31	<a href="#">oscl_file_async_read.h File Reference</a>	691
8.32	<a href="#">oscl_file_cache.h File Reference</a>	692
8.33	<a href="#">oscl_file_dir_utils.h File Reference</a>	693
8.34	<a href="#">oscl_file_find.h File Reference</a>	695
8.35	<a href="#">oscl_file_handle.h File Reference</a>	696
8.36	<a href="#">oscl_file_io.h File Reference</a>	697
8.37	<a href="#">oscl_file_manager.h File Reference</a>	698
8.38	<a href="#">oscl_file_native.h File Reference</a>	699
8.39	<a href="#">oscl_file_server.h File Reference</a>	700
8.40	<a href="#">oscl_file_stats.h File Reference</a>	701
8.41	<a href="#">oscl_file_types.h File Reference</a>	702
8.42	<a href="#">oscl_heapbase.h File Reference</a>	703
8.43	<a href="#">oscl_init.h File Reference</a>	704
8.44	<a href="#">oscl_int64_utils.h File Reference</a>	705
8.45	<a href="#">oscl_ip_socket.h File Reference</a>	706
8.46	<a href="#">oscl_linked_list.h File Reference</a>	707
8.47	<a href="#">oscl_lock_base.h File Reference</a>	708
8.48	<a href="#">oscl_map.h File Reference</a>	709
8.49	<a href="#">oscl_math.h File Reference</a>	710
8.50	<a href="#">oscl_media_data.h File Reference</a>	711
8.51	<a href="#">oscl_media_status.h File Reference</a>	712
8.52	<a href="#">oscl_mem.h File Reference</a>	713
8.53	<a href="#">oscl_mem_audit.h File Reference</a>	716
8.54	<a href="#">oscl_mem_audit_internals.h File Reference</a>	718
8.55	<a href="#">oscl_mem_auto_ptr.h File Reference</a>	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
8.100 oscl_socket_serv_imp.h File Reference . . . . .	765
8.101 oscl_socket_serv_imp_base.h File Reference . . . . .	766
8.102 oscl_socket_serv_imp_pv.h File Reference . . . . .	767
8.103 oscl_socket_serv_imp_reqlist.h File Reference . . . . .	768
8.104 oscl_socket_shutdown.h File Reference . . . . .	769
8.105 oscl_socket_stats.h File Reference . . . . .	770
8.106 oscl_socket_tuneables.h File Reference . . . . .	772
8.107 oscl_socket_types.h File Reference . . . . .	774
8.108 oscl_stdstring.h File Reference . . . . .	776
8.109 oscl_str_ptr_len.h File Reference . . . . .	778
8.110 oscl_string.h File Reference . . . . .	779
8.111 oscl_string_containers.h File Reference . . . . .	780
8.112 oscl_string_rep.h File Reference . . . . .	781
8.113 oscl_string_uri.h File Reference . . . . .	782
8.114 oscl_string_utf8.h File Reference . . . . .	783
8.115 oscl_string_utils.h File Reference . . . . .	784
8.116 oscl_string_xml.h File Reference . . . . .	785
8.117 oscl_tagtree.h File Reference . . . . .	786
8.118 oscl_tcp_socket.h File Reference . . . . .	787
8.119 oscl_thread.h File Reference . . . . .	788
8.120 oscl_tickcount.h File Reference . . . . .	790
8.121 oscl_time.h File Reference . . . . .	791
8.122 oscl_timer.h File Reference . . . . .	793
8.123 oscl_tls.h File Reference . . . . .	794
8.124 oscl_tree.h File Reference . . . . .	795
8.125 oscl_types.h File Reference . . . . .	796
8.126 oscl_udp_socket.h File Reference . . . . .	797
8.127 oscl_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:

_OscHeapBase	110
HeapBase	137
Osc_File	179
OSCL_String	260
OSCL_FastString	175
OSCL_HeapString< Alloc >	196
OSCL_HeapStringA	198
OSCL_StackString< MaxBufSize >	257
OscActiveObject	310
OscAsyncFile	316
OscDNSRequestAO	363
OscGetHostByNameRequest	414
OscSocketRequestAO	550
OscAcceptRequest	309
OscBindRequest	323
OscConnectRequest	349
OscListenRequest	423
OscRecvFromRequest	488
OscRecvRequest	491
OscSendRequest	524
OscSendToRequest	526
OscShutdownRequest	531
PVSchedulerStopper	631
OscAsyncFileBuffer	319
OscBuf	340
OscDNS	351
OscFileCache	401
OscNativeFile	462
OscPtr	477
OscPtrC	479
OscRegistryClient	509
OscSocketServ	554
OscTCPSocket	565



OscTimerObject	586
CallbackTimer< Alloc >	124
OscDNSMethod	358
OscGetHostByNameMethod	413
OscSocketMethod	545
OscAcceptMethod	308
OscBindMethod	322
OscConnectMethod	348
OscListenMethod	422
OscRecvFromMethod	486
OscRecvMethod	490
OscSendMethod	523
OscSendToMethod	525
OscShutdownMethod	530
OscSocketServI	556
OscUDPSocket	601
OscExecSchedulerBase	391
OscExecScheduler	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
OscTimer< Alloc >	582
CFastRep	127
CHearRep	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
Osc_Alloc	170
Osc_DefAlloc	172
_OscBasicAllocator	108
OscAllocDestructDealloc	314
OscMemAllocDestructDealloc< T >	427
OscMemBasicAllocDestructDealloc< T >	440

OscMemAllocator	426
OscMemBasicAllocator	439
OscMemPoolFixedChunkAllocator	443
OscMemPoolResizableAllocator	448
OscReadyAlloc	482
Osc_Dealloc	171
Osc_DefAlloc	172
Osc_File::OscCacheObserver	187
Osc_File::OscFixedCacheParam	188
Osc_FileFind	189
Osc_FileServer	193
oscl_fsstat	195
Osc_Int64_Utils	203
Osc_Less< T >	205
Osc_Linked_List_Base	211
Osc_Linked_List< LLClass, Alloc >	206
Osc_Map< Key, T, Alloc, Compare >	216
Osc_Map< Key, T, Alloc, Compare >::value_compare	223
Osc_MTLlinked_List< LLClass, Alloc, TheLock >	225
Osc_Opaque_Type_Alloc	229
Osc_Queue< T, Alloc >	236
Osc_Vector< T, Alloc >	285
Osc_Vector< TOscReady, OscReadyAlloc >	285
Osc_Opaque_Type_Alloc_LL	231
Osc_Linked_List< LLClass, Alloc >	206
Osc_Opaque_Type_Compare	233
OscPriorityQueue< Qelem, Alloc, Container, Compare >	470
OscPriorityQueue< TOscReady, OscReadyAlloc, Osc_Vector< TOscReady, OscReady-Alloc >, OscReadyCompare >	470
OscReadyQ	484
OscPriorityQueue< TOscReady, OscReadyAlloc, Osc_Vector< TOscReady, OscReady-Alloc >, OscTimerCompare >	470
OscTimerQ	591
Osc_Pair< T1, T2 >	235
Osc_Queue_Base	239
Osc_Queue< T, Alloc >	236
Osc_Rb_Tree_Base	246
Osc_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >	242
Osc_Rb_Tree_Const_Iterator< Value >	247
Osc_Rb_Tree_Iterator< Value >	250
Osc_Rb_Tree_Node_Base	254
Osc_Rb_Tree_Node< Value >	253
Osc_Select1st< V, U >	256
oscl_stat_buf	259
Osc_Tag_Base	267
Osc_Tag< Alloc >	265
Osc_TagTree< T, Alloc >	269
Osc_TagTree< T, Alloc >::const_iterator	273
Osc_TagTree< T, Alloc >::iterator	276
Osc_TagTree< T, Alloc >::Node	279
Osc_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

OscIpMReq . . . . .	417
OscIPSocketI . . . . .	418
OscTCPSocketI . . . . .	572
OscUDPSocketI . . . . .	607
OscJump . . . . .	421
OscLockBase . . . . .	424
OscMutex . . . . .	459
OscNullLock . . . . .	467
OscThreadLock . . . . .	579
OscMem . . . . .	425
OscMemAudit . . . . .	429
OSCLMemAutoPtr< T, _Allocator > . . . . .	435
OscMemGlobalAuditObject . . . . .	441
OscMemoryFragment . . . . .	442
BufferFragment . . . . .	117
OscMemPoolFixedChunkAllocatorObserver . . . . .	447
OscMemPoolResizableAllocator::MemPoolBlockInfo . . . . .	454
OscMemPoolResizableAllocator::MemPoolBufferInfo . . . . .	455
OscMemPoolResizableAllocatorMemoryObserver . . . . .	456
OscMemPoolResizableAllocatorObserver . . . . .	457
OscMemStatsNode . . . . .	458
OscNameString< __len > . . . . .	461
OscNativeFileParams . . . . .	465
OscNetworkAddress . . . . .	466
OscPriorityQueueBase . . . . .	474
OscPriorityQueue< Qelem, Alloc, Container, Compare > . . . . .	470
OscPriorityQueue< TOscReady, OscReadyAlloc, Osc_Vector< TOscReady, OscReady-Alloc >, OscReadyCompare > . . . . .	470
OscPriorityQueue< TOscReady, OscReadyAlloc, Osc_Vector< TOscReady, OscReady-Alloc >, OscTimerCompare > . . . . .	470
OscProcStatus . . . . .	475
OscRand . . . . .	481
OscReadyCompare . . . . .	483
OscRefCount . . . . .	492
Osc_DefAllocWithRefCount< DefAlloc > . . . . .	173
OscRefCountDA . . . . .	494
OscRefCountMTDA< LockType > . . . . .	498
OscRefCountMTSA< DeallocType, LockType > . . . . .	500
OscRefCountSA< DeallocType > . . . . .	502
OscRefCountMemFrag . . . . .	496
OscRegistryAccessClient . . . . .	504
OscRegistryAccessElement . . . . .	508
OscRegistryClientImpl . . . . .	511
OscRegistryAccessClientImpl . . . . .	506
OscRegistryServTlsImpl . . . . .	514
OscRegistryAccessClientTlsImpl . . . . .	507
OscRegistryClientTlsImpl . . . . .	513
OscScheduler . . . . .	516
OscSchedulerObserver . . . . .	517
OscScopedLock< LockClass > . . . . .	518
OscSelect . . . . .	519
OscSemaphore . . . . .	521

OscSharedPtr< TheClass > . . . . .	527
OscSingleton< T, ID, Registry > . . . . .	532
OscSingletonRegistry . . . . .	534
OscSocketIBase . . . . .	540
OscSocketI . . . . .	535
OscSocketObserver . . . . .	548
OscSocketRequest . . . . .	549
OscSocketServIBase . . . . .	558
OscSocketServI . . . . .	556
OscSocketServRequestList . . . . .	560
OscSocketServRequestQElem . . . . .	562
OscSocketTOS . . . . .	563
OscThread . . . . .	575
OscTickCount . . . . .	580
OscTimerCompare . . . . .	585
OscTimerObserver . . . . .	590
OscTLS< T, ID, Registry > . . . . .	592
OscTLSEx< T, ID, Registry > . . . . .	594
OscTLSRegistry . . . . .	596
OscTLSRegistryEx . . . . .	597
OscTrapItem . . . . .	598
OscTrapStack . . . . .	599
OscTrapStackItem . . . . .	600
OscUuid . . . . .	610
PVActiveBase . . . . .	612
OscActiveObject . . . . .	310
OscTimerObject . . . . .	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 . . . . .	<a href="#">658</a>
----------------------	---------------------

## Chapter 3

# oscl Data Structure Index

### 3.1 oscl Data Structures

Here are the data structures with brief descriptions:

<a href="#">_OscBasicAllocator</a>	108
<a href="#">_OscHeapBase</a>	110
<a href="#">AcceptParam</a>	112
<a href="#">allocator</a>	113
<a href="#">AllPassFilter</a>	114
<a href="#">BindParam</a>	116
<a href="#">BufferFragment</a>	117
<a href="#">BufferMgr</a>	118
<a href="#">BufferState</a>	119
<a href="#">BufFragGroup&lt; ChainClass, max_frags &gt;</a>	120
<a href="#">BufFragStatusClass</a>	123
<a href="#">CallbackTimer&lt; Alloc &gt;</a>	124
<a href="#">CallbackTimerObserver</a>	126
<a href="#">CFastRep</a>	127
<a href="#">CHepRep</a>	129
<a href="#">ConnectParam</a>	131
<a href="#">CStackRep</a>	132
<a href="#">DNSRequestParam</a>	133
<a href="#">GetHostByNameParam</a>	135
<a href="#">HeapBase</a>	137
<a href="#">internalLeave</a>	139
<a href="#">LinkedListElement&lt; LLClass &gt;</a>	140
<a href="#">ListenParam</a>	141
<a href="#">MediaData&lt; ChainClass, max_frags, local_bufsize &gt;</a>	142
<a href="#">MediaStatusClass</a>	145
<a href="#">MemAllocator&lt; T &gt;</a>	146
<a href="#">MM_AllocBlockFence</a>	147
<a href="#">MM_AllocBlockHdr</a>	148
<a href="#">MM_AllocInfo</a>	149
<a href="#">MM_AllocNode</a>	151
<a href="#">MM_AllocQueryInfo</a>	152
<a href="#">MM_Audit_Imp</a>	153
<a href="#">MM_AuditOverheadStats</a>	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::OscI_CacheObserver	187
OscI_File::OscI_FixedCacheParam	188
OscI_FileFind	189
OscI_FileServer	193
oscl_fsstat	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_MTLLinked_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



<a href="#">OSCL_wString</a>	304
<a href="#">OsciAcceptMethod</a>	308
<a href="#">OsciAcceptRequest</a>	309
<a href="#">OsciActiveObject</a>	310
<a href="#">OsciAllocDestructDealloc</a>	314
<a href="#">OsciAOStatus</a>	315
<a href="#">OsciAsyncFile</a>	316
<a href="#">OsciAsyncFileBuffer</a>	319
<a href="#">OsciAuditCB</a>	321
<a href="#">OsciBindMethod</a>	322
<a href="#">OsciBindRequest</a>	323
<a href="#">OsciBinIStream</a>	324
<a href="#">OsciBinIStreamBigEndian</a>	326
<a href="#">OsciBinIStreamLittleEndian</a>	329
<a href="#">OsciBinOStream</a> (Class <a href="#">OsciBinOStream</a> implements the basic stream functions for an output stream)	331
<a href="#">OsciBinOStreamBigEndian</a> (Class <a href="#">OsciBinOStreamBigEndian</a> implements a binary output stream using big endian byte ordering)	332
<a href="#">OsciBinOStreamLittleEndian</a> (Class <a href="#">OsciBinOStreamLittleEndian</a> implements a binary output stream using little endian byte ordering)	334
<a href="#">OsciBinStream</a>	336
<a href="#">OsciBuf</a>	340
<a href="#">OsciCompareLess&lt; T &gt;</a>	342
<a href="#">OsciComponentRegistry</a>	343
<a href="#">OsciComponentRegistryData</a>	345
<a href="#">OsciComponentRegistryElement</a>	346
<a href="#">OsciConnectMethod</a>	348
<a href="#">OsciConnectRequest</a>	349
<a href="#">OsciDestructDealloc</a>	350
<a href="#">OsciDNS</a>	351
<a href="#">OsciDNSI</a>	353
<a href="#">OsciDNSIBase</a>	355
<a href="#">OsciDNSMethod</a>	358
<a href="#">OsciDNSObserver</a>	361
<a href="#">OsciDNSRequest</a>	362
<a href="#">OsciDNSRequestAO</a>	363
<a href="#">OsciDoubleLink</a>	366
<a href="#">OsciDoubleList&lt; T &gt;</a>	367
<a href="#">OsciDoubleListBase</a>	368
<a href="#">OsciDoubleRunner&lt; T &gt;</a>	370
<a href="#">OsciError</a>	372
<a href="#">OsciErrorAllocator</a> (This class provides static methods to invoke the user defined memory allocation routines)	374
<a href="#">OsciErrorTrap</a>	376
<a href="#">OsciErrorTrapImp</a>	377
<a href="#">OsciException&lt; LeaveCode &gt;</a> ( <a href="#">Osci_exception.h</a> contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from)	379
<a href="#">OsciExclusiveArrayPtr&lt; T &gt;</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 <a href="#">OsciExclusiveArrayPtr</a> expires, its destructor uses delete to free the memory)	380
<a href="#">OsciExclusivePtr&lt; T &gt;</a> (Template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the <a href="#">OsciExclusivePtr</a> expires, its destructor uses delete to free the memory)	383

<a href="#">OsciExclusivePtrA&lt; T, Alloc &gt;</a> (Template class that defines any pointer like object intended to be assigned an address obtained (directly or or indirectly) through Alloc. When the OsciExclusivePtrA expires, Alloc is used to free the memory) . . . . .	386
<a href="#">OsciExecScheduler</a> . . . . .	389
<a href="#">OsciExecSchedulerBase</a> . . . . .	391
<a href="#">OsciExecSchedulerCommonBase</a> . . . . .	392
<a href="#">OsciFileCache</a> . . . . .	401
<a href="#">OsciFileCacheBuffer</a> . . . . .	403
<a href="#">OsciFileHandle</a> . . . . .	405
<a href="#">OsciFileManager</a> . . . . .	406
<a href="#">OsciFileStats</a> . . . . .	411
<a href="#">OsciFileStatsItem</a> . . . . .	412
<a href="#">OsciGetHostByNameMethod</a> . . . . .	413
<a href="#">OsciGetHostByNameRequest</a> . . . . .	414
<a href="#">OsciInit</a> . . . . .	415
<a href="#">OsciInteger64Transport</a> . . . . .	416
<a href="#">OsciIpMReq</a> . . . . .	417
<a href="#">OsciIPSocketI</a> . . . . .	418
<a href="#">OsciJump</a> . . . . .	421
<a href="#">OsciListenMethod</a> . . . . .	422
<a href="#">OsciListenRequest</a> . . . . .	423
<a href="#">OsciLockBase</a> . . . . .	424
<a href="#">OsciMem</a> . . . . .	425
<a href="#">OsciMemAllocator</a> . . . . .	426
<a href="#">OsciMemAllocDestructDealloc&lt; T &gt;</a> . . . . .	427
<a href="#">OsciMemAudit</a> . . . . .	429
<a href="#">OSCLMemAutoPtr&lt; T, _Allocator &gt;</a> (The osci_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 osci_auto_ptr expires, its destructor uses delete to free the memory) . . . . .	435
<a href="#">OsciMemBasicAllocator</a> . . . . .	439
<a href="#">OsciMemBasicAllocDestructDealloc&lt; T &gt;</a> . . . . .	440
<a href="#">OsciMemGlobalAuditObject</a> . . . . .	441
<a href="#">OsciMemoryFragment</a> . . . . .	442
<a href="#">OsciMemPoolFixedChunkAllocator</a> . . . . .	443
<a href="#">OsciMemPoolFixedChunkAllocatorObserver</a> . . . . .	447
<a href="#">OsciMemPoolResizableAllocator</a> . . . . .	448
<a href="#">OsciMemPoolResizableAllocator::MemPoolBlockInfo</a> . . . . .	454
<a href="#">OsciMemPoolResizableAllocator::MemPoolBufferInfo</a> . . . . .	455
<a href="#">OsciMemPoolResizableAllocatorMemoryObserver</a> . . . . .	456
<a href="#">OsciMemPoolResizableAllocatorObserver</a> . . . . .	457
<a href="#">OsciMemStatsNode</a> . . . . .	458
<a href="#">OsciMutex</a> . . . . .	459
<a href="#">OsciNameString&lt; __len &gt;</a> . . . . .	461
<a href="#">OsciNativeFile</a> . . . . .	462
<a href="#">OsciNativeFileParams</a> . . . . .	465
<a href="#">OsciNetworkAddress</a> . . . . .	466
<a href="#">OsciNullLock</a> . . . . .	467
<a href="#">OsciPriorityLink</a> . . . . .	468
<a href="#">OsciPriorityList&lt; T &gt;</a> . . . . .	469
<a href="#">OsciPriorityQueue&lt; Qelem, Alloc, Container, Compare &gt;</a> . . . . .	470
<a href="#">OsciPriorityQueueBase</a> . . . . .	474
<a href="#">OsciProcStatus</a> . . . . .	475
<a href="#">OsciPtr</a> . . . . .	477

OsciPtrC	479
OsciRand	481
OsciReadyAlloc	482
OsciReadyCompare	483
OsciReadyQ	484
OsciRecvFromMethod	486
OsciRecvFromRequest	488
OsciRecvMethod	490
OsciRecvRequest	491
OsciRefCount	492
OsciRefCountDA	494
OsciRefCountMemFrag	496
OsciRefCountMTDA< LockType >	498
OsciRefCountMTSA< DeallocType, LockType >	500
OsciRefCountSA< DeallocType >	502
OsciRegistryAccessClient	504
OsciRegistryAccessClientImpl	506
OsciRegistryAccessClientTlsImpl	507
OsciRegistryAccessElement	508
OsciRegistryClient	509
OsciRegistryClientImpl	511
OsciRegistryClientTlsImpl	513
OsciRegistryServTlsImpl	514
OsciScheduler	516
OsciSchedulerObserver	517
OsciScopedLock< LockClass > (Template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the Osci-ScopedLock goes out of scope)	518
OsciSelect	519
OsciSemaphore	521
OsciSendMethod	523
OsciSendRequest	524
OsciSendToMethod	525
OsciSendToRequest	526
OsciSharedPtr< TheClass > (A parameterized smart pointer class)	527
OsciShutdownMethod	530
OsciShutdownRequest	531
OsciSingleton< T, ID, Registry >	532
OsciSingletonRegistry	534
OsciSocketI	535
OsciSocketIBase	540
OsciSocketMethod	545
OsciSocketObserver	548
OsciSocketRequest	549
OsciSocketRequestAO	550
OsciSocketServ	554
OsciSocketServI	556
OsciSocketServIBase	558
OsciSocketServRequestList	560
OsciSocketServRequestQElem	562
OsciSocketTOS	563
OsciTCPSocket	565
OsciTCPSocketI	572
OsciThread	575

OscThreadLock	579
OscTickCount	580
OscTimer< Alloc >	582
OscTimerCompare	585
OscTimerObject	586
OscTimerObserver	590
OscTimerQ	591
OscTLS< T, ID, Registry >	592
OscTLSEx< T, ID, Registry >	594
OscTLSRegistry	596
OscTLSRegistryEx	597
OscTrapItem	598
OscTrapStack	599
OscTrapStackItem	600
OscUDPSocket	601
OscUDPSocketI	607
OscUuid	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 <a href="#">StrPtrLen</a> , 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:

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

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

<a href="#">oscl_priqueue.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 <a href="#">OscQueue</a> . It is similar to the STL::queue class, with some differences: - less complete - based on array rather than a deque - some interfaces modeled on <a href="#">oscl_vector</a> , 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 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 OscRegistry) . . . . .	733
<a href="#">oscl_registry_client_impl.h</a> (Client-side implementation of OscRegistryInterface) . . . . .	734
<a href="#">oscl_registry_serv_impl.h</a> (Server-side implementation of OscRegistry 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 Osc registry interfaces) . . . . .	738
<a href="#">oscl_scheduler.h</a> . . . . .	739
<a href="#">oscl_scheduler_ao.h</a> (Osc Scheduler user execution object classes) . . . . .	740
<a href="#">oscl_scheduler_aobase.h</a> (Osc Scheduler internal active object classes) . . . . .	741
<a href="#">oscl_scheduler_readyq.h</a> (Ready q types for osc scheduler) . . . . .	742
<a href="#">oscl_scheduler_threadcontext.h</a> (Thread context functions needed by osc scheduler) . . . . .	743
<a href="#">oscl_scheduler_tuneables.h</a> (Tuneable settings for Osc 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 <a href="#">OscSharedPtr</a> which is a "smart pointer" to the parameterized type) . . . . .	747
<a href="#">oscl_singleton.h</a> (This file defines the <a href="#">OscSingleton</a> 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, 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)	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 to 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 <a href="#">OscLuid32</a> )	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 <a href="#">osclerror</a> )	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



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

## Chapter 5

# oscl Page Index

### 5.1 oscl Related Pages

Here is a list of all related documentation pages:

Todo List . . . . .	<a href="#">861</a>
---------------------	---------------------

## 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 [OscVector](#) and [OscMap](#) class, etc.*
- file [oscl\\_dll.h](#)  
*Defines a DLL entry point.*
- file [oscl\\_exclusive\\_ptr.h](#)  
*This file defines the [OscExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.*
- file [oscl\\_linked\\_list.h](#)  
*The file [oscl\\_linked\\_list.h](#) defines the template class [OscLinkedList](#) 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, [OscLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OscNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OscScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.*
- file [oscl\\_map.h](#)  
*The file [oscl\\_map.h](#) defines the template class [OscMap](#) which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.*
- file [oscl\\_mem\\_inst.h](#)  
*The file defines default memory instrumentation level.*

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

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

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

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

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

A general purpose reference counter to object lifetimes.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## Data Structures

- class [\\_OscLBasicAllocator](#)
- class [LinkedListElement](#)
- class [NTPTIME](#)

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



- class [OscI\\_Alloc](#)
- class [OscI\\_Dealloc](#)
- class [OscI\\_DefAlloc](#)
- class [OscI\\_DefAllocWithRefCount](#)
- struct [OscI\\_Less](#)
- class [OscI\\_Linked\\_List](#)
- class [OscI\\_Linked\\_List\\_Base](#)
- class [OscI\\_Map](#)
- class [OscI\\_MTLlinked\\_List](#)
- class [OscI\\_Opaque\\_Type\\_Alloc](#)
- class [OscI\\_Opaque\\_Type\\_Alloc\\_LL](#)
- class [OscI\\_Opaque\\_Type\\_Compare](#)
- struct [OscI\\_Pair](#)
- class [OscI\\_Queue](#)
- class [OscI\\_Queue\\_Base](#)
- class [OscI\\_Rb\\_Tree](#)
- class [OscI\\_Rb\\_Tree\\_Base](#)
- struct [OscI\\_Rb\\_Tree\\_Const\\_Iterator](#)
- struct [OscI\\_Rb\\_Tree\\_Iterator](#)
- struct [OscI\\_Rb\\_Tree\\_Node](#)
- struct [OscI\\_Rb\\_Tree\\_Node\\_Base](#)
- struct [OscI\\_Select1st](#)
- struct [OscI\\_Tag](#)
- struct [OscI\\_Tag\\_Base](#)
- class [OscI\\_TagTree](#)
- class [OscI\\_TAlloc](#)
- class [OscI\\_Vector](#)
- class [OscI\\_Vector\\_Base](#)
- class [OscIAllocDestructDealloc](#)
- class [OscIDestructDealloc](#)
- class [OscIExclusiveArrayPtr](#)

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

- class [OscIExclusivePtr](#)

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

- class [OscIExclusivePtrA](#)

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

- class [OscILockBase](#)
- struct [OscIMemoryFragment](#)
- class [OscINullOrLock](#)
- class [OscIRefCount](#)
- class [OscIRefCountDA](#)

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

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

- class [OscSharedPtr](#)

*A parameterized smart pointer class.*

- class [OscTLS](#)
- class [OscTLSRegistry](#)
- class [TimeValue](#)

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

- class [TLSSStorageOps](#)

## 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 [OscAny](#) [TOscTlsKey](#)
- typedef int [c\\_bool](#)

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

- typedef void [OscAny](#)

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

- typedef char [mbchar](#)

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

- typedef unsigned int [uint](#)

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

- typedef uint8 [octet](#)

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

- typedef float [OscFloat](#)

*The Float type defined as OscFloat.*

- typedef [OSCL\\_NATIVE\\_INT64\\_TYPE](#) [int64](#)
- typedef [OSCL\\_NATIVE\\_UINT64\\_TYPE](#) [uint64](#)
- typedef [OSCL\\_NATIVE\\_WCHAR\\_TYPE](#) [oscl\\_wchar](#)
- typedef [oscl\\_wchar](#) [OSCL\\_TCHAR](#)

*define OSCL\_TCHAR*

## Enumerations

- enum [TimeUnits](#) { [SECONDS](#) = 0, [MILLISECONDS](#) = 1, [MICROSECONDS](#) = 2 }

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

## Functions

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

- OSCL\_IMPORT\_REF int32 `oscl_CIstrncmp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2, uint32 count)
- OSCL\_IMPORT\_REF char `oscl_tolower` (const char car)
- OSCL\_IMPORT\_REF `oscl_wchar` `oscl_tolower` (const `oscl_wchar` car)
- OSCL\_IMPORT\_REF bool `oscl_isLetter` (const char car)
- OSCL\_IMPORT\_REF const char \* `oscl_strstr` (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF char \* `oscl_strstr` (char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF const `oscl_wchar` \* `oscl_strstr` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strstr` (`oscl_wchar` \*str1, const `oscl_wchar` \*str2)
- OSCL\_IMPORT\_REF char \* `oscl_strcat` (char \*dest, const char \*src)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strcat` (`oscl_wchar` \*dest, const `oscl_wchar` \*src)
- OSCL\_IMPORT\_REF void `PV8601ToRFC822` (`PV8601timeStrBuf` pv8601\_buffer, `CtimeStrBuf` ctime\_buffer)
- OSCL\_IMPORT\_REF void `ISO8601ToRFC822` (`ISO8601timeStrBuf` iso8601\_buffer, `CtimeStrBuf` ctime\_buffer)
- OSCL\_IMPORT\_REF void `RFC822ToPV8601` (`CtimeStrBuf` ctime\_buffer, `PV8601timeStrBuf`)
- OSCL\_COND\_IMPORT\_REF `TimeValue` operator- (const `TimeValue` &a, const `TimeValue` &b)
- OSCL\_COND\_IMPORT\_REF `TimeValue` operator+ (const `TimeValue` &a, const int32 bSeconds)
- OSCL\_COND\_IMPORT\_REF `TimeValue` operator+ (const int32 aSeconds, const `TimeValue` &b)
- OSCL\_COND\_IMPORT\_REF `TimeValue` operator- (const `TimeValue` &a, const int32 bSeconds)
- OSCL\_COND\_IMPORT\_REF `TimeValue` operator- (const int32 aSeconds, const `TimeValue` &b)
- bool `operator==` (const `OscSharedPtr` &b) const  
*Test for equality to see if two PVHandles wrap the same object.*
- void `Bind` (const `OscSharedPtr` &inHandle)  
*Use this function to bind an existing `OscSharedPtr` to a already-wrapped object.*
- void `Bind` (`TheClass` \*ptr, `OscRefCount` \*in\_refcnt)  
*Use this function to bind an existing `OscSharedPtr` 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):OSCL\_Assert(#\_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 OslSharedPtr< TheClass >::Bind (TheClass \* *ptr*, OslRefCounter \* *in\_refcnt*) [inline, inherited]

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

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

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

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

Convert host to big endian format.

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

#### Parameters:

*data* A pointer to the input/output buffer

*size* The number of bytes in the buffer.

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

Convert host to little endian format.

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

#### Parameters:

*data* A pointer to the input/output buffer

*size* The number of bytes in the buffer.

**6.2.5.7 OSCL\_IMPORT\_REF void ISO8601ToRFC822 (ISO8601timeStrBuf iso8601\_buffer, CtimeStrBuf ctime\_buffer)**

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

Convert little endian to host format.

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

**Parameters:**

*data* A pointer to the input/output buffer

*size* The number of bytes in the buffer.

**6.2.5.9 OSCL\_COND\_IMPORT\_REF TimeValue operator+ (const int32 aSeconds, const TimeValue & b)**

**6.2.5.10 OSCL\_COND\_IMPORT\_REF TimeValue operator+ (const TimeValue & a, const int32 bSeconds)**

**6.2.5.11 OSCL\_COND\_IMPORT\_REF TimeValue operator- (const int32 aSeconds, const TimeValue & b)**

**6.2.5.12 OSCL\_COND\_IMPORT\_REF TimeValue operator- (const TimeValue & a, const int32 bSeconds)**

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

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

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

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

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

**Parameters:**

*expr* is the expression to be evaluated

*filename* is the name of the current source file

*line\_number* is the line number in the current source file

**6.2.5.16 OSCL\_IMPORT\_REF int32 osci\_CIstrcmp (const osci\_wchar \* str1, const osci\_wchar \* str2)**

Case in-sensitive string comparison.

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

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

#### Parameters:

*dest* null terminated destination string

*src* source string

*count* number of characters to append.

#### Returns:

*dest*

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

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

#### Parameters:

*dest* null terminated destination string

*src* source string

#### Returns:

*dest*

### 6.2.5.23 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strchr (oscl\_wchar \* *str*, int32 *c*)

### 6.2.5.24 OSCL\_IMPORT\_REF const oscl\_wchar\* oscl\_strchr (const oscl\_wchar \* *str*, int32 *c*)

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

#### Parameters:

*str* null terminated source string

*c* character to search for

#### Returns:

**6.2.5.25 OSCL\_IMPORT\_REF** `char* oscl_strchr (char * str, int32 c)`

**6.2.5.26 OSCL\_IMPORT\_REF** `const char* oscl_strchr (const char * str, int32 c)`

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

**Parameters:**

*str* null terminated source string  
*c* character to search for

**Returns:**

**6.2.5.27 OSCL\_IMPORT\_REF** `int32 oscl_strcmp (const oscl\_wchar * str1, const oscl\_wchar * str2)`

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

**Parameters:**

*str1* String to compare  
*str2* String to compare

**Returns:**

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

**6.2.5.28 OSCL\_IMPORT\_REF** `int32 oscl_strcmp (const char * str1, const char * str2)`

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

**Parameters:**

*str1* String to compare  
*str2* String to compare

**Returns:**

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

**6.2.5.29 OSCL\_IMPORT\_REF** `uint32 oscl_strlen (const oscl\_wchar * str)`

Gets the length of a wide char string

**Parameters:**

*str* NULL terminated string.

**Returns:**

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

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

Gets the length of a string

**Parameters:**

*str* NULL terminated string.

**Returns:**

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

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

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

**Parameters:**

*dest* null terminated destination string

*src* source string

*count* number of characters to append.

**Returns:**

dest

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

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

**Parameters:**

*dest* null terminated destination string

*src* source string

*count* number of characters to append.

**Returns:**

dest

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

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

**Parameters:**

*str1* String to compare

*str2* String to compare

*count* Number of characters to compare

#### Returns:

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

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

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

#### Parameters:

*str1* String to compare

*str2* String to compare

*count* Number of characters to compare

#### Returns:

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

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

Copies the chars of one string to another.

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

#### Parameters:

*dest* Destination string

*src* NULL terminated source string

*count* Number of chars to copy

#### Returns:

Returns *dest*.

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

Copies the chars of one string to another.

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

#### Parameters:

*dest* Destination string



*src* NULL terminated source string

*count* Number of chars to copy

#### Returns:

Returns dest.

**6.2.5.37** OSCL\_IMPORT\_REF [oscl\\_wchar\\*](#) oscl\_strchr ([oscl\\_wchar](#) \* *str*, int32 *c*)

**6.2.5.38** OSCL\_IMPORT\_REF const [oscl\\_wchar\\*](#) oscl\_strchr (const [oscl\\_wchar](#) \* *str*, int32 *c*)

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

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

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

#### Parameters:

*str* null terminated source string

*c* character to search for

#### Returns:

**6.2.5.41** OSCL\_IMPORT\_REF [oscl\\_wchar\\*](#) oscl\_strset ([oscl\\_wchar](#) \* *dest*, [oscl\\_wchar](#) *val*, uint32 *count*)

Sets the characters of a string to a specified character

#### Parameters:

*dest* buffer to modify

*val* character to set

*count* number of chars to set

#### Returns:

the value of dest

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

Sets the characters of a string to a specified character

#### Parameters:

*dest* buffer to modify

*val* character to set

*count* number of chars to set

#### Returns:

the value of dest

**6.2.5.43** OSCL\_IMPORT\_REF `oscl_wchar*` `oscl_strstr` (`oscl_wchar * str1`, `const oscl_wchar * str2`)

**6.2.5.44** OSCL\_IMPORT\_REF `const oscl_wchar*` `oscl_strstr` (`const oscl_wchar * str1`, `const oscl_wchar * str2`)

find the occurrence of sub-string in a string.

**Parameters:**

*str1* string.

*str2* sub-string

**Returns:**

pointer to the 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\_PERRORTRAP = 5

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

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

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

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

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

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

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

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

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

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

## 6.3 OSCL Memory

### Files

- file [oscl\\_mem.h](#)  
*This file contains basic memory definitions for common use across platforms.*
- file [oscl\\_mem\\_audit.h](#)  
*This file contains the definition and partial implementation of MM\_Audit class.*
- file [oscl\\_mem\\_audit\\_internals.h](#)  
*This file contains the internal definitions for the mem audit library.*
- file [oscl\\_mem\\_auto\\_ptr.h](#)  
*This file defines the oscl\_mem\_auto\_ptr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.*
- file [oscl\\_mem\\_basic\\_functions.h](#)  
*This file contains prototypes for the basic memory functions.*
- file [oscl\\_mem\\_mempool.h](#)  
*This file contains the definition of memory pool allocators.*

### Data Structures

- class [allocator](#)
- class [allocator](#)
- class [HeapBase](#)
- struct [MM\\_AllocBlockFence](#)
- struct [MM\\_AllocBlockHdr](#)
- struct [MM\\_AllocInfo](#)
- struct [MM\\_AllocNode](#)
- struct [MM\\_AllocQueryInfo](#)
- class [MM\\_Audit\\_Imp](#)
- struct [MM\\_AuditOverheadStats](#)
- struct [MM\\_FailInsertParam](#)
- struct [MM\\_Stats\\_CB](#)
- struct [MM\\_Stats\\_t](#)
- class [OscIAuditCB](#)
- class [OscIMem](#)
- class [OscIMemAllocator](#)
- class [OscIMemAllocator](#)
- class [OscIMemAllocDestructDealloc](#)
- class [OscIMemAllocDestructDealloc](#)
- class [OscIMemAudit](#)
- class [OSCLMemAutoPtr](#)

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

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

### Defines

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

- #define `_OSCL_CLEANUP_BASE_CLASS(T) this → T::~~T()`
- #define `MM_ALLOC_MAX_QUERY_FILENAME_LEN` 128
- #define `MM_ALLOC_MAX_QUERY_TAG_LEN` 64
- #define `MM_AUDIT_VALIDATE_BLOCK` 1
- #define `MM_AUDIT_PREFILL_FLAG` 0x1
- #define `MM_AUDIT_POSTFILL_FLAG` 0x2
- #define `MM_AUDIT_VALIDATE_ALL_HEAP_FLAG` 0x4
- #define `MM_AUDIT_VALIDATE_ON_FREE_FLAG` 0x8
- #define `MM_AUDIT_ALLOC_NODE_ENABLE_FLAG` 0x10
- #define `MM_AUDIT_SUPPRESS_FILENAME_FLAG` 0x20
- #define `DEFAULT_MM_AUDIT_MODE` 0
- #define `MM_AUDIT_ALLOC_NODE_SUPPORT` 1
- #define `MM_AUDIT_FENCE_SUPPORT` 0
- #define `MM_AUDIT_INCLUDE_ALL_HEAP_VALIDATION` 1
- #define `MM_AUDIT_FILL_SUPPORT` 0
- #define `MM_AUDIT_FAILURE_SIMULATION_SUPPORT` 1
- #define `FENCE_PATTERN` 0xAA
- #define `MIN_FENCE_SIZE` 4
- #define `MEM_ALIGN_SIZE` 8
- #define `COMPUTE_MEM_ALIGN_SIZE(x, y, z) (y+(((x+y)%z) ? (z - (x+y)%z) : 0))`
- #define `DEFAULT_PREFILL_PATTERN` 0x96
- #define `DEFAULT_POSTFILL_PATTERN` 0x5A
- #define `OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT`

## Typedefs

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

## Functions

- OSCL\_COND\_IMPORT\_REF void \* `_oscl_malloc` (int32 count)
- OSCL\_COND\_IMPORT\_REF void \* `_oscl_calloc` (int32 nelems, int32 size)
- OSCL\_COND\_IMPORT\_REF void \* `_oscl_realloc` (void \*src, int32 count)
- OSCL\_COND\_IMPORT\_REF void `_oscl_free` (void \*src)
- OSCL\_COND\_IMPORT\_REF void \* `oscl_memcpy` (void \*dest, const void \*src, uint32 count)
- OSCL\_COND\_IMPORT\_REF void \* `oscl_memmove` (void \*dest, const void \*src, uint32 count)
- OSCL\_COND\_IMPORT\_REF void \* `oscl_memmove32` (void \*dest, const void \*src, uint32 count)
- OSCL\_COND\_IMPORT\_REF void \* `oscl_memset` (void \*dest, uint8 val, uint32 count)

- OSCL\_COND\_IMPORT\_REF int `oscl_memcmp` (const void \*buf1, const void \*buf2, uint32 count)
- OSCL\_COND\_IMPORT\_REF uint `oscl_mem_aligned_size` (uint size)
- OSCL\_IMPORT\_REF void `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;\
    OsclAny* __ptr=exp;\
    OSCL_TRY(__err,T_ptr=new(__ptr) T params);\
    if(__err){\
        freeFunc(__ptr);\
        T_ptr=NULL;\
        OsclError::Leave(__err);\
    }\
}
```



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

**Parameters:**

*exp*: expression to allocate memory.

*Tptr:variable* to hold result.

*T*: type

*params*: constructor arg list

*freeFunc*: delete or free function.

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

**6.3.1.4** `#define DEFAULT_MM_AUDIT_MODE 0`

**6.3.1.5** `#define DEFAULT_POSTFILL_PATTERN 0x5A`

**6.3.1.6** `#define DEFAULT_PREFILL_PATTERN 0x96`

**6.3.1.7** `#define FENCE_PATTERN 0xAA`

**6.3.1.8** `#define MEM_ALIGN_SIZE 8`

**6.3.1.9** `#define MIN_FENCE_SIZE 4`

**6.3.1.10** `#define MM_ALLOC_MAX_QUERY_FILENAME_LEN 128`

**6.3.1.11** `#define MM_ALLOC_MAX_QUERY_TAG_LEN 64`

**6.3.1.12** `#define MM_AUDIT_ALLOC_NODE_ENABLE_FLAG 0x10`

**6.3.1.13** `#define MM_AUDIT_ALLOC_NODE_SUPPORT 1`

**6.3.1.14** `#define MM_AUDIT_FAILURE_SIMULATION_SUPPORT 1`

**6.3.1.15** `#define MM_AUDIT_FENCE_SUPPORT 0`

**6.3.1.16** `#define MM_AUDIT_FILL_SUPPORT 0`

**6.3.1.17** `#define MM_AUDIT_INCLUDE_ALL_HEAP_VALIDATION 1`

**6.3.1.18** `#define MM_AUDIT_POSTFILL_FLAG 0x2`

**6.3.1.19** `#define MM_AUDIT_PREFILL_FLAG 0x1`

**6.3.1.20** `#define MM_AUDIT_SUPPRESS_FILENAME_FLAG 0x20`

**6.3.1.21** `#define MM_AUDIT_VALIDATE_ALL_HEAP_FLAG 0x4`

**6.3.1.22** `#define MM_AUDIT_VALIDATE_BLOCK 1`

**6.3.1.23** `#define MM_AUDIT_VALIDATE_ON_FREE_FLAG 0x8`

**6.3.1.24** `#define OSCL_ALLOC_DELETE(ptr, T_allocator, T)`

**Value:**

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

Deletes the object of type T using the given allocator

**Parameters:**

*T\_allocator* allocator for objects of type T

*T* type of object to delete

*ptr* pointer to previously created object

**Exceptions:**

*none* , unless thrown by the given allocator

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

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

**Parameters:**

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

*T* type of object to create

*params* object initialization parameters

**Returns:**

pointer to created object

**Exceptions:**

*none* , unless thrown by the given allocator

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

OscL array delete operator..

**Parameters:**

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

**Returns:**

void

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

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

**Parameters:**

*T* data type for 'new' operation

*count* number of elements to create

**Returns:**

pointer to the newly created object array of type T

**Exceptions:**

*may* leave with code = bad alloc

**6.3.1.28** `#define OSCL_AUDIT_ARRAY_NEW(auditCB, T, count)`  
`new(_oscl_audit_new(sizeof(T)*(count),auditCB)) T`

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

**Parameters:**

*auditCB* input memory management audit object

*T* data type for 'new' operation

*count* number of elements to create

**Returns:**

pointer to the newly created object array of type T

**Exceptions:**

*may* leave with code = bad alloc

**6.3.1.29** `#define OSCL_AUDIT_CALLOC(auditCB, num, size) _oscl_audit_calloc(num,size,`  
`auditCB)`

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

**Parameters:**

*auditCB* input memory management audit object

*num* number of elements

*size* number of bytes to allocate for each element

**Returns:**

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

**Exceptions:**

*none*

**6.3.1.30** `#define OSCL_AUDIT_MALLOC(auditCB, count) _oscl_audit_malloc(count, auditCB)`

Allocates a memory block using the given audit object.

**Parameters:**

*auditCB* input memory management audit object

*count* number of bytes to allocate

**Returns:**

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

**Exceptions:**

*none*

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

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_malloc(a, b) OSCL_CALLOC(a,b)`

**6.3.1.34** `#define OSCL_CALLOC(num, size) _oscl_default_audit_malloc(num,size)`

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

**Parameters:**

*num* number of elements

*size* number of bytes to allocate for each element

**Returns:**

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

**Exceptions:**

*none*

### 6.3.1.35 #define OSCL\_CLEANUP\_BASE\_CLASS(T) \_OSCL\_CLEANUP\_BASE\_CLASS(T)

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

#### Parameters:

*T*: name of the base class.

### 6.3.1.36 #define OSCL\_DEFAULT\_FREE(x) OSCL\_FREE(x)

Another back-compatibility definition.

### 6.3.1.37 #define OSCL\_DEFAULT\_MALLOC(x) OSCL\_MALLOC(x)

Another back-compatibility definition.

### 6.3.1.38 #define OSCL\_DELETE(ptr)

#### Value:

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

Oscl "delete" operator.

#### Parameters:

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

#### Returns:

void

### 6.3.1.39 #define OSCL\_DISABLE\_WARNING\_RETURN\_TYPE\_NOT\_UDT

### 6.3.1.40 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE

Previously this was in oscl\_mem\_imp.h

### 6.3.1.41 #define oscl\_free(x) OSCL\_FREE(x)

### 6.3.1.42 #define OSCL\_FREE(ptr) \_oscl\_audit\_free(ptr)

Deallocates or frees a memory block.

#### Parameters:

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

**6.3.1.43** `#define OSCL_HAS_GLOBAL_NEW_DELETE 1`

**6.3.1.44** `#define oscl_malloc(a) OSCL_MALLOC(a)`

**6.3.1.45** `#define OSCL_MALLOC(count) _oscl_default_audit_malloc(count)`

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

**Parameters:**

*count* number of bytes to allocate

**Returns:**

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

**Exceptions:**

*none*

**6.3.1.46** `#define OSCL_NEW(T, params) new T params`

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

**Parameters:**

*T* data type for 'new' operation

*params* object initialization parameters

**Returns:**

pointer to the newly created object of type T

**Exceptions:**

*may* leave with code = bad alloc

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

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

**6.3.1.49** `#define OSCL_REALLOC(ptr, new_size) _oscl_default_audit_realloc(ptr,new_size)`

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

**Parameters:**

*ptr* original memory block

*new\_size* New size of the block

**Returns:**

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

**Exceptions:**

*none*

**6.3.1.50** `#define OSCL_TRAP_ALLOC_NEW(T_ptr, T_allocator, T, params) _OSCL_TRAP_NEW(T_allocator.allocate(1),T_allocator.deallocate,T_ptr,T,params)`

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

**Parameters:**

*T\_ptr* variable to hold return value– pointer to new object of type T.  
*T\_allocator* allocator for objects of type T, must be an `Oscl_TAlloc<T, Allocator>`, where Allocator is an `Oscl_DefAlloc`  
*T* type of object to create  
*params* object initialization parameters

**Returns:**

pointer to created object

**Exceptions:**

*none* , unless thrown by the given allocator

**6.3.1.51** `#define OSCL_TRAP_AUDIT_NEW(T_ptr, auditCB, T, params) _OSCL_TRAP_NEW(_oscl_audit_new(sizeof(T),auditCB),_oscl_audit_free,T_ptr,T,params)`

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

**Parameters:**

*T\_ptr* variable to hold return value– pointer to new object of type T.  
*auditCB* input memory management audit object  
*T* data type for 'new' operation  
*params* object initialization parameters

**Returns:**

pointer to the newly created object of type T

**Exceptions:**

*may* leave with code = bad alloc

**6.3.1.52** `#define OSCL_TRAP_NEW(T_ptr, T, params) _OSCL_TRAP_NEW(_oscl_default_audit_new(sizeof(T)),_oscl_audit_free,T_ptr,T,params)`

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

**Parameters:**

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



*T* data type for 'new' operation  
*params* object initialization parameters

**Returns:**

pointer to the newly created object of type *T*

**Exceptions:**

*may* leave with code = bad alloc

## 6.3.2 Typedef Documentation

6.3.2.1 typedef **OSCLMemAutoPtr**<**MM\_AllocNode**, **OscL\_TAlloc**<**MM\_AllocNode**,  
**OscMemBasicAllocator**> > **MM\_AllocNodeAutoPtr**

6.3.2.2 typedef **OSCLMemAutoPtr**<**OscMemStatsNode**, **OscL\_TAlloc**<**OscMemStatsNode**,  
**OscMemBasicAllocator**> > **MM\_StatsNodeTagTreeType**

6.3.2.3 typedef **OSCLMemAutoPtr**<char, **OscL\_TAlloc**<char, **OscMemBasicAllocator**> >  
**MMAuditCharAutoPtr**

6.3.2.4 typedef **OSCLMemAutoPtr**<uint8, **OscL\_TAlloc**<uint8, **\_OscBasicAllocator**> >  
**MMAuditUint8AutoPtr**

6.3.2.5 typedef **OSCLMemAutoPtr**<**OscMemStatsNode**, **OscL\_TAlloc**<**OscMemStatsNode**,  
**OscMemBasicAllocator**> > **OscMemStatsNodeAutoPtr**

6.3.2.6 typedef **OscL\_TagTree**<**MM\_StatsNodeTagTreeType**, **TagTree\_Allocator**>  
**OscTagTreeType**

6.3.2.7 typedef **OscL\_TAlloc**<**MM\_StatsNodeTagTreeType**, **OscMemBasicAllocator**>  
**TagTree\_Allocator**

## 6.3.3 Function Documentation

6.3.3.1 **OSCL\_IMPORT\_REF** void\* **\_oscl\_audit\_calloc** (size\_t, size\_t, **OscAuditCB** &, 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, **OscAuditCB** &, 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, [OscAuditCB](#) &, const char \*f = NULL, const int l = 0)
- 6.3.3.5** OSCL\_IMPORT\_REF void\* \_oscl\_audit\_realloc (void \*, size\_t, [OscAuditCB](#) &, const char \*f = NULL, const int l = 0)
- 6.3.3.6** OSCL\_COND\_IMPORT\_REF void\* \_oscl\_calloc (int32 *nelems*, int32 *size*)
- 6.3.3.7** OSCL\_IMPORT\_REF void\* \_oscl\_default\_audit\_calloc (size\_t, size\_t, const char \*f = NULL, const int l = 0)
- 6.3.3.8** OSCL\_IMPORT\_REF void\* \_oscl\_default\_audit\_malloc (size\_t, const char \*f = NULL, const int l = 0)
- 6.3.3.9** OSCL\_IMPORT\_REF void\* \_oscl\_default\_audit\_new (size\_t, const char \*f = NULL, const int l = 0)
- 6.3.3.10** OSCL\_IMPORT\_REF void\* \_oscl\_default\_audit\_realloc (void \*, size\_t, const char \*f = NULL, const int l = 0)
- 6.3.3.11** OSCL\_COND\_IMPORT\_REF void \_oscl\_free (void \*src)
- 6.3.3.12** OSCL\_COND\_IMPORT\_REF void\* \_oscl\_malloc (int32 *count*)
- 6.3.3.13** OSCL\_COND\_IMPORT\_REF void\* \_oscl\_realloc (void \*src, int32 *count*)
- 6.3.3.14** void operator delete (void \*) [inline]
- 6.3.3.15** ]
- void operator delete[] (void \*aPtr) [inline]
- 6.3.3.16** void\* operator new (size\_t) [inline]
- 6.3.3.17** void\* operator new (size\_t aSize, const char \*aFile, int aLine) [inline]
- 6.3.3.18** ]
- void\* operator new[] (size\_t aSize) [inline]
- 6.3.3.19** ]
- void\* operator new[] (size\_t aSize, const char \*aFile, int aLine) [inline]
- 6.3.3.20** OSCL\_COND\_IMPORT\_REF [uint](#) oscl\_mem\_aligned\_size ([uint](#) size)

Get memory-aligned size of an object.

**Parameters:**

*size* size of object

**Returns:**

memory-aligned size

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

Compare characters in two buffers

**Parameters:***buf1* first buffer*buf2* second buffer*count* number of bytes to compare**Returns:**<0 *buf1* less than *buf2* 0 *buf1* equal to *buf2* >0 *buf1* greater than *buf2***6.3.3.22 OSCL\_COND\_IMPORT\_REF void\* oscl\_memcpy (void \* *dest*, const void \* *src*, uint32 *count*)**

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

**Parameters:***dest* new buffer*src* buffer to copy*count* number of bytes to copy**Returns:**the value of `dest`**6.3.3.23 OSCL\_COND\_IMPORT\_REF void\* oscl\_memmove (void \* *dest*, const void \* *src*, uint32 *count*)**

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

**Parameters:***dest* new buffer*src* buffer to copy*count* number of bytes to copy**Returns:**the value of `dest`

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

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

##### Parameters:

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

##### Returns:

the value of dest

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

Sets the bytes of a buffer to a specified character

##### Parameters:

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

##### Returns:

the value of dest

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

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

##### Parameters:

*auditCB* memory management audit object

### 6.3.4 Variable Documentation

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

## 6.4 OSCL Util

### Files

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

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

## Data Structures

- class [BufferFragment](#)
- class [BufferMgr](#)
- class [BufferState](#)
- class [BufFragGroup](#)
- class [BufFragStatusClass](#)
- class [CFastRep](#)
- class [CHepRep](#)
- 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 [OscBinIStream](#)
- class [OscBinIStreamBigEndian](#)
- class [OscBinIStreamLittleEndian](#)

- class [OscBinOStream](#)  
*Class OscBinOStream implements the basic stream functions for an output stream.*
- class [OscBinOStreamBigEndian](#)  
*Class OscBinOStreamBigEndian implements a binary output stream using big endian byte ordering.*
- class [OscBinOStreamLittleEndian](#)  
*Class OscBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.*
- class [OscBinStream](#)
- class [OscCompareLess](#)
- class [OscComponentRegistry](#)
- class [OscComponentRegistryData](#)
- class [OscComponentRegistryElement](#)
- class [OscPriorityQueue](#)
- class [OscPriorityQueueBase](#)
- class [OscRand](#)
- class [OscRegistryAccessClient](#)
- class [OscRegistryAccessClientImpl](#)
- class [OscRegistryAccessClientTlsImpl](#)
- class [OscRegistryAccessElement](#)
- class [OscRegistryClient](#)
- class [OscRegistryClientImpl](#)
- class [OscRegistryClientTlsImpl](#)
- class [OscRegistryServTlsImpl](#)
- class [OscTickCount](#)
- struct [StrCSumPtrLen](#)  
*same as [StrPtrLen](#), but includes checksum field and method to speed up querying*
- struct [StrPtrLen](#)  
*This data structure encapsulates a set of functions used to perform.*
- struct [WStrPtrLen](#)  
*This data structure encapsulates a set of functions used to perform.*

## Defines

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

## Typedefs

- typedef [OscAny](#) \* [OscComponentFactory](#)
- typedef void(\* [BufferFreeFuncPtr](#) )(void \*)
- typedef uint32 [MediaTimestamp](#)
- typedef StrPtrLen [StrPtrLen](#)  
*This data structure encapsulates a set of functions used to perform.*

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

## Enumerations

- enum [TOSCL\\_StringOp](#) { [EOSCL\\_StringOp\\_CompressASCII](#), [EOSCL\\_StringOp\\_UTF16ToUTF8](#) }
- enum [TOSCL\\_wStringOp](#) { [EOSCL\\_wStringOp\\_ExpandASCII](#), [EOSCL\\_wStringOp\\_UTF8ToUTF16](#) }

## Functions

- [OSCL\\_IMPORT\\_REF](#) const char \* [skip\\_whitespace](#) (const char \*ptr)
- [OSCL\\_IMPORT\\_REF](#) char \* [skip\\_whitespace](#) (char \*ptr)
- [OSCL\\_IMPORT\\_REF](#) const char \* [skip\\_whitespace](#) (const char \*start, const char \*end)
- [OSCL\\_IMPORT\\_REF](#) const char \* [skip\\_to\\_whitespace](#) (const char \*start, const char \*end)
- [OSCL\\_IMPORT\\_REF](#) const char \* [skip\\_to\\_line\\_term](#) (const char \*start\_ptr, const char \*end\_ptr)
- [OSCL\\_IMPORT\\_REF](#) const char \* [skip\\_whitespace\\_and\\_line\\_term](#) (const char \*start, const char \*end)
- [OSCL\\_IMPORT\\_REF](#) int [extract\\_string](#) (const char \*in\_ptr, char \*outstring, int maxsize)
- [OSCL\\_IMPORT\\_REF](#) int [extract\\_string](#) (const char \*start, const char \*end, char \*outstring, int max-size)
- [OSCL\\_IMPORT\\_REF](#) bool [PV\\_atoi](#) (const char \*buf, const char new\_format, uint32 &value)
- [OSCL\\_IMPORT\\_REF](#) bool [PV\\_atoi](#) (const char \*buf, const char new\_format, int length, uint32 &value)
- [OSCL\\_IMPORT\\_REF](#) bool [PV\\_atoi](#) (const char \*buf, const char new\_format, int length, [uint64](#) &value)
- [OSCL\\_IMPORT\\_REF](#) bool [PV\\_atof](#) (const char \*buf, [OscFloat](#) &value)
- [OSCL\\_IMPORT\\_REF](#) bool [PV\\_atof](#) (const char \*buf, int length, [OscFloat](#) &value)
- [OSCL\\_IMPORT\\_REF](#) int [oscl\\_abs](#) (int aVal)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_log](#) (double value)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_log10](#) (double value)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_sqrt](#) (double value)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_pow](#) (double x, double y)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_exp](#) (double value)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_sin](#) (double value)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_cos](#) (double value)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_tan](#) (double value)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_asin](#) (double value)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_atan](#) (double value)
- [OSCL\\_COND\\_IMPORT\\_REF](#) double [oscl\\_floor](#) (double value)
- [OSCL\\_IMPORT\\_REF](#) int32 [oscl\\_snprintf](#) (char \*str, uint32 count, const char \*fmt,...)
- [OSCL\\_IMPORT\\_REF](#) int32 [oscl\\_snprintf](#) ([oscl\\_wchar](#) \*str, uint32 count, const [oscl\\_wchar](#) \*fmt,...)
- [OSCL\\_IMPORT\\_REF](#) int32 [oscl\\_vsnprintf](#) (char \*str, uint32 count, const char \*fmt, va\_list args)



- OSCL\_IMPORT\_REF int32 [oscl\\_vsnprintf](#) ([oscl\\_wchar](#) \*str, uint32 count, const [oscl\\_wchar](#) \*fmt, va\_list args)
- OSCL\_IMPORT\_REF bool [oscl\\_str\\_unescape\\_uri](#) (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes, uint32 &out\_buf\_len)  
*unescape any of the special escape sequence in the uri string*
- OSCL\_IMPORT\_REF bool [oscl\\_str\\_unescape\\_uri](#) (const [OSCL\\_String](#) &oscl\_str\_in, [OSCL\\_String](#) &oscl\_str\_out, uint32 &out\_buf\_len)  
*unescape any of the special escape sequence in the uri string*
- OSCL\_IMPORT\_REF bool [oscl\\_str\\_is\\_valid\\_utf8](#) (const uint8 \*str\_buf, uint32 &num\_valid\_characters, uint32 max\_bytes=0, uint32 max\_char\_2\_valid=0, uint32 \*num\_byte\_4\_char=NULL)  
*Check if the input string contains any illegal UTF-8 character. The function scans the string and validate that each character is a valid utf-8. It stops at the first NULL character, invalid character or the max\_byte value. The string is valid if and only if every character is a valid utf-8 character and the scanning stopped on a character boundary.*
- OSCL\_IMPORT\_REF int32 [oscl\\_str\\_truncate\\_utf8](#) (uint8 \*str\_buf, uint32 max\_char, uint32 max\_bytes=0)  
*Truncates the UTF-8 string upto the required size.*
- OSCL\_IMPORT\_REF bool [oscl\\_str\\_need\\_escape\\_xml](#) (const char \*str\_buf, uint32 &num\_escape\_bytes, uint32 max\_bytes=0)  
*Check if the input string contains any special ASCII character like &, <, >, ', ". The function scans the string and check if each character is a special character. It stops at the first NULL character (if max\_bytes = 0), or the max\_byte value.*
- OSCL\_IMPORT\_REF int32 [oscl\\_str\\_escape\\_xml](#) (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes=0, uint32 \*num\_bytes\_written=NULL)  
*Escape any of the following special characters in the string Special ASCII characters: &, <, >, ', "*
- OSCL\_IMPORT\_REF int32 [oscl\\_UTF8ToUnicode](#) (const char \*input, int32 inLength, [oscl\\_wchar](#) \*output, int32 outLength)  
*Convert UTF8 byte sequence to Unicode string.*
- OSCL\_IMPORT\_REF int32 [oscl\\_UnicodeToUTF8](#) (const [oscl\\_wchar](#) \*input, int32 inLength, char \*output, int32 outLength)  
*Convert Unicode string to UTF8 byte sequence.*
- [BufferFragment](#) \* [GetFragment](#) (const int32 idx)
- [BufferState](#) \* [GetBufferState](#) (const int32 idx)
- uint32 [get\\_size](#) () const
- uint32 [get\\_size](#) () const
- uint32 [get\\_maxsize](#) () const
- uint32 [get\\_maxsize](#) () const
- const chartype \* [get\\_cstr](#) () const
- const chartype \* [get\\_cstr](#) () const
- chartype \* [get\\_str](#) () const
- chartype \* [get\\_str](#) () const
- [OSCL\\_HeapString](#) ()
- [OSCL\\_wHeapString](#) ()

- [OSCL\\_HeapString](#) (const chartype \*cstr)
- [OSCL\\_wHeapString](#) (const chartype \*cstr)
- void [set](#) (const chartype \*buf, uint32 length)
- void [set](#) (const chartype \*buf, uint32 length)
- void [set](#) (const other\_chartype \*buf, optype op)
- void [set](#) (const other\_chartype \*buf, optype op)
- void [set](#) (const other\_chartype \*buf, uint32 length, optype op)
- void [set](#) (const other\_chartype \*buf, uint32 length, optype op)
- [OSCL\\_HeapString](#) (const chartype \*buf, uint32 length)
- [OSCL\\_wHeapString](#) (const chartype \*buf, uint32 length)
- [OSCL\\_HeapString](#) (const OSCL\_HeapString &src)
- [OSCL\\_wHeapString](#) (const OSCL\_wHeapString &src)
- [OSCL\\_HeapString](#) (const [OSCL\\_String](#) &src)
- [OSCL\\_wHeapString](#) (const [OSCL\\_wString](#) &src)
- [~OSCL\\_HeapString](#) ()
- [~OSCL\\_wHeapString](#) ()
- [OSCL\\_HeapString](#) & [operator=](#) (const [OSCL\\_HeapString](#) &src)
- [OSCL\\_wHeapString](#) & [operator=](#) (const [OSCL\\_wHeapString](#) &src)
- [OSCL\\_HeapString](#) & [operator=](#) (const [OSCL\\_String](#) &src)
- [OSCL\\_wHeapString](#) & [operator=](#) (const [OSCL\\_wString](#) &src)
- [OSCL\\_HeapString](#) & [operator=](#) (const chartype \*cstr)
- [OSCL\\_wHeapString](#) & [operator=](#) (const chartype \*cstr)
- uint32 [get\\_size](#) () const
- uint32 [get\\_size](#) () const
- uint32 [get\\_maxsize](#) () const
- uint32 [get\\_maxsize](#) () const
- const chartype \* [get\\_cstr](#) () const
- const chartype \* [get\\_cstr](#) () const
- chartype \* [get\\_str](#) () const
- chartype \* [get\\_str](#) () const
- [OSCL\\_StackString](#) ()
- [OSCL\\_wStackString](#) ()
- [OSCL\\_StackString](#) (const chartype \*cstr)
- [OSCL\\_wStackString](#) (const chartype \*cstr)
- void [set](#) (const chartype \*buf, uint32 length)
- void [set](#) (const chartype \*buf, uint32 length)
- void [set](#) (const other\_chartype \*buf, optype op)
- void [set](#) (const other\_chartype \*buf, optype op)
- void [set](#) (const other\_chartype \*buf, uint32 length, optype op)
- void [set](#) (const other\_chartype \*buf, uint32 length, optype op)
- [OSCL\\_StackString](#) (const chartype \*buf, uint32 length)
- [OSCL\\_wStackString](#) (const chartype \*buf, uint32 length)
- [OSCL\\_StackString](#) (const OSCL\_StackString &src)
- [OSCL\\_wStackString](#) (const OSCL\_wStackString &src)
- [OSCL\\_StackString](#) (const [OSCL\\_String](#) &src)
- [OSCL\\_wStackString](#) (const [OSCL\\_wString](#) &src)
- [~OSCL\\_StackString](#) ()
- [~OSCL\\_wStackString](#) ()
- [OSCL\\_StackString](#) & [operator=](#) (const [OSCL\\_StackString](#) &src)
- [OSCL\\_wStackString](#) & [operator=](#) (const [OSCL\\_wStackString](#) &src)

- `OSCL_StackString & operator=` (const `OSCL_String` &src)
- `OSCL_wStackString & operator=` (const `OSCL_wString` &src)
- `OSCL_StackString & operator=` (const `chartype *cstr`)
- `OSCL_wStackString & operator=` (const `chartype *cstr`)

## Variables

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

### 6.4.1 Define Documentation

#### 6.4.1.1 `#define MAX_NUMBER_OF_BYTE_PER_UTF8 3`

Define the maximum UTF8 representation in bytes.

#### Todo:

Handle 4-byte surrogate pair representation

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

#### 6.4.1.3 `#define OSCLTICKCOUNT_MAX_TICKS 0xffffffff`

### 6.4.2 Typedef Documentation

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

#### 6.4.2.2 `typedef uint32 MediaTimestamp`

#### 6.4.2.3 `typedef WStrPtrLen OSCL_TStrPtrLen`

#### 6.4.2.4 `typedef OsclAny* OsclComponentFactory`

`OsclComponentFactory` is an opaque pointer.

#### 6.4.2.5 `typedef StrCsumPtrLen StrCsumPtrLen`

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

#### 6.4.2.6 `typedef struct StrPtrLen StrPtrLen`

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

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

### 6.4.2.7 typedef struct WStrPtrLen WStrPtrLen

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

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

## 6.4.3 Enumeration Type Documentation

### 6.4.3.1 enum TOSCL\_StringOp

Conversion operations for [OSCL\\_String](#) classes

Enumeration values:

**EOSCL\_StringOp\_CompressASCII**

**EOSCL\_StringOp\_UTF16ToUTF8**

### 6.4.3.2 enum TOSCL\_wStringOp

Conversion operations for [OSCL\\_wString](#) classes

Enumeration values:

**EOSCL\_wStringOp\_ExpandASCII**

**EOSCL\_wStringOp\_UTF8ToUTF16**

## 6.4.4 Function Documentation

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

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

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

Implements [OSCL\\_wString](#).

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

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

Implements [OSCL\\_String](#).

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

Implements [OSCL\\_wString](#).

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

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

Implements [OSCL\\_String](#).

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

Implements [OSCL\\_wString](#).

**6.4.4.8** `template<uint32 MaxBufSize> uint32 OSCL_StackString< MaxBufSize >::get_maxsize  
() [virtual, inherited]`

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

Implements [OSCL\\_String](#).

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

Implements [OSCL\\_wString](#).

**6.4.4.10** `template<class Alloc> uint32 OSCL_HeapString< Alloc >::get_maxsize ()  
[virtual, inherited]`

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

Implements [OSCL\\_String](#).

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

Implements [OSCL\\_wString](#).

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

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

Implements [OSCL\\_String](#).

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

Implements [OSCL\\_wString](#).

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

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

Implements [OSCL\\_String](#).

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

Implements [OSCL\\_wString](#).

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

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

Implements [OSCL\\_String](#).

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

Implements [OSCL\\_wString](#).

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

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

Implements [OSCL\\_String](#).

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

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

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

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

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

Assignment operator

**am:** null-terminated string

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

**6.4.4.23** `template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize > & OSCL_wStackString< MaxBufSize >::operator= (const OSCL\_wString & src)`  
[inherited]

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

**6.4.4.24** `template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize > & OSCL_StackString< MaxBufSize >::operator= (const OSCL\_String & src)`  
[inherited]

Assignment operator

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

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

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

Assignment operators

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

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

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

Assignment operator

**am:** null-terminated string

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

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

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

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

Assignment operator

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

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

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

Assignment operators

**6.4.4.33** `OSCL_IMPORT_REF int oscl_abs (int aVal)`

**6.4.4.34** `OSCL_COND_IMPORT_REF double oscl_asin (double value)`

Calculates the arc since of a number

**Parameters:**

*value* source value

**6.4.4.35** `OSCL_COND_IMPORT_REF double oscl_atan (double value)`

Calculates the arc tangent of a number

**Parameters:**

*value* source value

**6.4.4.36** `OSCL_COND_IMPORT_REF double oscl_cos (double value)`

Calculates the cosine of a number

**Parameters:**

*value* source value

**6.4.4.37** `OSCL_COND_IMPORT_REF double oscl_exp (double value)`

Calculates the exponential of e for a number

**Parameters:**

*value* source value

**6.4.4.38** `OSCL_COND_IMPORT_REF double oscl_floor (double value)`

Calculates the floor of a number

**Parameters:**

*value* source value



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

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

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

**Parameters:**

*src*: input string.

**6.4.4.41** `template<class Alloc> OSCL_HeapString< Alloc >::OSCL_HeapString (const chartype * buf, uint32 length) [inherited]`

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

**Parameters:**

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

*length*: number of characters to copy.

**6.4.4.42** `template<class Alloc> OSCL_HeapString< Alloc >::OSCL_HeapString (const chartype * cstr) [inherited]`

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

**Parameters:**

*cp*: null-terminated string.

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

The default constructor creates an empty string.

**6.4.4.44** `OSCL_COND_IMPORT_REF double oscl_log (double value)`

Calculates the natural log of a number

**Parameters:**

*value* source value

**6.4.4.45** `OSCL_COND_IMPORT_REF double oscl_log10 (double value)`

Calculates the logarithm to base 10 of a number

**Parameters:**

*value* source value

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

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

##### Parameters:

*x* base value  
*y* power

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

Calculates the sine of a number

##### Parameters:

*value* source value

#### 6.4.4.48 OSCL\_IMPORT\_REF int32 oscl\_snprintf (oscl\_wchar \* *str*, uint32 *count*, const oscl\_wchar \* *fmt*, ...)

#### 6.4.4.49 OSCL\_IMPORT\_REF int32 oscl\_snprintf (char \* *str*, uint32 *count*, const char \* *fmt*, ...)

#### 6.4.4.50 OSCL\_COND\_IMPORT\_REF double oscl\_sqrt (double *value*)

Calculates the square root of a number

##### Parameters:

*value* source value

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

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

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

##### Parameters:

*src*: input string.

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

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

##### Parameters:

*src*: a character array, not necessarily null-terminated.  
*length*: the number of characters to copy.

**6.4.4.54** `template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize >::OSCL_StackString (const chartype * cstr) [inherited]`

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

**Parameters:**

*cp*: a null-terminated string.

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

Creates an OSCL\_StackString initialized with an empty string.

**6.4.4.56** `OSCL_IMPORT_REF int32 oscl_str_escape_xml (const char * str_buf_in, char * str_buf_out, uint32 max_out_buf_bytes, uint32 max_bytes = 0, uint32 * num_bytes_written = NULL)`

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

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

**Parameters:**

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

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

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

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

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

**Returns:**

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

**6.4.4.57** `OSCL_IMPORT_REF bool oscl_str_is_valid_utf8 (const uint8 * str_buf, uint32 & num_valid_characters, uint32 max_bytes = 0, uint32 max_char_2_valid = 0, uint32 * num_byte_4_char = NULL)`

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

**Parameters:**

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

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

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

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

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

#### Returns:

True if the string is valid and false otherwise.

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

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

#### Parameters:

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

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

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

#### Returns:

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

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

Truncates the UTF-8 string upto the required size.

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

#### Parameters:

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

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

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

#### Returns:

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

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

unescape any of the special escape sequence in the uri string

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

##### Parameters:

*oscl\_str\_in* Ptr to an input OSCL\_String  
*oscl\_str\_out* Ptr to an output OSCL\_String which stores the modified string  
*out\_buf\_len* The length of the result string (not including the null character)

##### Returns:

It returns true if succeeds, otherwise false.

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

unescape any of the special escape sequence in the uri string

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

##### Parameters:

*str\_buf\_in* Ptr to an input string  
*str\_buf\_out* Ptr to an output buffer which stores the modified string  
*max\_out\_buf\_bytes* The size of str\_buf\_out.  
*max\_bytes* The maximum number of bytes to read. It is the length of str\_buf\_in.  
*out\_buf\_len* The length of the result string (not including the null character)

##### Returns:

It returns true if succeeds, otherwise false.

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

Calculates the tangential of a number

##### Parameters:

*value* source value

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

Convert Unicode string to UTF8 byte sequence.

The function converts Unicode string to UTF8 byte sequence. The length of input Unicode string is specified. It stops at two conditions: (A) Whole input Unicode string is successfully converted. (B) Destination bufer 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 bufer 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 **char**type \* *buf*, uint32 *length*) [inherited]
- 6.4.4.70 template<class Alloc> OSCL\_wHeapString< Alloc >::OSCL\_wHeapString (const **char**type \* *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 **char**type \* *buf*, uint32 *length*) [inherited]
- 6.4.4.75 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString (const **char**type \* *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\_char**type \* *buf*, uint32 *length*, **optype** *op*) [inherited]
- 6.4.4.83 template<uint32 MaxBufSize> void OSCL\_StackString< MaxBufSize >::set (const **other\_char**type \* *buf*, uint32 *length*, **optype** *op*) [inherited]



**Parameters:**

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

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

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

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

**Parameters:**

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

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

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

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

**Parameters:**

*buf*: string or character array.  
*length*: number of characters to copy.

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

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

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

**Parameters:**

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

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

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

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

**Parameters:**

*buf*: NULL-terminated wide string.

*op*: conversion operation to apply

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

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

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

**Parameters:**

*buf*: string or character array.

*length*: number of characters to copy.

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

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

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

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

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

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

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

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

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

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

## 6.4.5 Variable Documentation

**6.4.5.1** `const int32 APPEND_MEDIA_AT_END = -1`

**6.4.5.2** `const uint8 OSCL_ASCII_CASE_MAGIC_BIT = 0x20`

## 6.5 OSCL Error

### Files

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

### Data Structures

- class [\\_OscHeapBase](#)
- class [internalLeave](#)
- class [OscError](#)
- class [OscErrorAllocator](#)  
*This class provides static methods to invoke the user defined memory allocation routines.*

- class `OscErrorTrap`
- class `OscErrorTrapImp`
- class `OscException`

*`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 `OscJump`
- class `OscNameString`
- class `OscTLSEx`
- class `OscTLSRegistryEx`
- class `OscTrapItem`
- class `OscTrapStack`
- class `OscTrapStackItem`

## Defines

- #define `OSCL_TRAPSTACK_PUSH(a)` `OscError::PushL(a)`
- #define `OSCL_TRAPSTACK_POP()` `OscError::Pop()`
- #define `OSCL_TRAPSTACK_POPDEALLOC()` `OscError::PopDealloc()`
- #define `OscErrNone` 0
- #define `OscErrGeneral` 100
- #define `OscErrNoMemory` 101
- #define `OscErrCancelled` 102
- #define `OscErrNotSupported` 103
- #define `OscErrArgument` 104
- #define `OscErrBadHandle` 105
- #define `OscErrAlreadyExists` 106
- #define `OscErrBusy` 107
- #define `OscErrNotReady` 108
- #define `OscErrCorrupt` 109
- #define `OscErrTimeout` 110
- #define `OscErrOverflow` 111
- #define `OscErrUnderflow` 112
- #define `OscErrInvalidState` 113
- #define `OscErrNoResources` 114
- #define `OscErrNotInstalled` 115
- #define `OscErrAlreadyInstalled` 116
- #define `OscErrSystemCallFailed` 117
- #define `OscErrNoHandler` 118
- #define `OscErrThreadContextIncorrect` 119
- #define `OSCL_ERR_NONE` `OscErrNone`
- #define `OSCL_BAD_ALLOC_EXCEPTION_CODE` `OscErrNoMemory`
- #define `OscSuccess` 0
- #define `OscPending` 1
- #define `OscFailure` -1
- #define `PERROR_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 [PERRORTRAP\\_REGISTRY\\_ID](#) [OSCL\\_TLS\\_ID\\_PERRORTRAP](#)
- #define [PERRORTRAP\\_REGISTRY](#) [OscTLSRegistry](#)
- #define [OSCL\\_LEAVE](#)([\\_leave\\_status](#)) [OscError::Leave\(\\_leave\\_status\)](#)  
*Use this macro to cause a Leave. It terminates the execution of the current active function.*
- #define [OSCL\\_TRY](#)([\\_leave\\_status](#), [\\_statements](#)) [\\_PV\\_TRAP\(\\_leave\\_status, \\_statements\)](#)  
*This macro will be used to set up a try block.*
- #define [OSCL\\_TRY\\_NO\\_TLS](#)([\\_\\_trapimp](#), [\\_leave\\_status](#), [\\_statements](#)) [\\_PV\\_TRAP\\_NO\\_TLS\(\\_\\_trapimp, \\_leave\\_status, \\_statements\)](#)
- #define [OSCL\\_FIRST\\_CATCH\\_ANY](#)([\\_leave\\_status](#), [\\_statements](#)) if ([\\_leave\\_status](#)!=[OscErrNone](#)) { [\\_statements](#); }  
*This section defines the macros to be used in the catch block following the try block Use this macro to call a function that handles all exception types thrown in the preceding try block.*
- #define [OSCL\\_FIRST\\_CATCH](#)([\\_leave\\_status](#), [\\_catch\\_value](#), [\\_statements](#)) if ([\\_leave\\_status](#)!=[OscErrNone](#) && [\\_leave\\_status](#) == [\\_catch\\_value](#)){[\\_statements](#);}  
*Use this macro to define a block of code that catches the first exception type thrown in the preceding try block.*
- #define [OSCL\\_CATCH](#)([\\_leave\\_status](#), [\\_catch\\_value](#), [\\_statements](#)) else if ([\\_leave\\_status](#)!=[OscErrNone](#) && [\\_leave\\_status](#) == [\\_catch\\_value](#)){[\\_statements](#);}  
*Use this macro to define a block of code for catching additional exception types.*
- #define [OSCL\\_CATCH\\_ANY](#)([\\_leave\\_status](#), [\\_statements](#)) else if ([\\_leave\\_status](#)!=[OscErrNone](#)){ [\\_statements](#);}  
*Use this macro to call a function that will catch all remaining exception types.*
- #define [OSCL\\_LAST\\_CATCH](#)([\\_leave\\_status](#)) else if ([\\_leave\\_status](#)!=[OscErrNone](#)){[OSCL\\_LEAVE\(\\_leave\\_status\)](#);}  
*Use this macro if [OSCL\\_CATCH\\_ANY](#) has not been used. It will mark the end of the catch block.*

## Typedefs

- typedef int32 [OscLeaveCode](#)
- typedef int32 [OscReturnCode](#)
- typedef void(\* [OscTrapOperation](#) )(OscAny \*)

## Functions

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

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

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

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

## 6.5.1 Define Documentation

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

**Value:**

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

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

**Value:**

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

### 6.5.1.3 #define internalLeave (-1)

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

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

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

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

**Parameters:**

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

*exceptiontype* is the exception handled by this catch block

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

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

**Parameters:**

*\_leave\_status*

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

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

For backward compatibility with old definitions

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

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

**Parameters:**

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

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

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

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

**Parameters:**

*\_leave\_status*

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

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

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

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

**Parameters:**

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

#### 6.5.1.12 #define OSCL\_LEAVE(\_leave\_status) OslError::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() OslError::Pop()

#### 6.5.1.15 #define OSCL\_TRAPSTACK\_POPDEALLOC() OslError::PopDealloc()

#### 6.5.1.16 #define OSCL\_TRAPSTACK\_PUSH(a) OslError::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 OsciErrAlreadyExists 106`

**6.5.1.20** `#define OsciErrAlreadyInstalled 116`

**6.5.1.21** `#define OsciErrArgument 104`

**6.5.1.22** `#define OsciErrBadHandle 105`

**6.5.1.23** `#define OsciErrBusy 107`

**6.5.1.24** `#define OsciErrCancelled 102`

**6.5.1.25** `#define OsciErrCorrupt 109`

**6.5.1.26** `#define OsciErrGeneral 100`

**6.5.1.27** `#define OsciErrInvalidState 113`

**6.5.1.28** `#define OsciErrNoHandler 118`

**6.5.1.29** `#define OsciErrNoMemory 101`

**6.5.1.30** `#define OsciErrNone 0`

**6.5.1.31** `#define OsciErrNoResources 114`

**6.5.1.32** `#define OsciErrNotInstalled 115`

**6.5.1.33** `#define OsciErrNotReady 108`

**6.5.1.34** `#define OsciErrNotSupported 103`

**6.5.1.35** `#define OsciErrOverflow 111`

**6.5.1.36** `#define OsciErrSystemCallFailed 117`

**6.5.1.37** `#define OsciErrThreadContextIncorrect 119`

**6.5.1.38** `#define OsciErrTimeout 110`

**6.5.1.39** `#define OsciErrUnderflow 112`

**6.5.1.40** `#define OsciFailure -1`

**6.5.1.41** `#define OsciPending 1`

**6.5.1.42** `#define OsciSuccess 0`

**6.5.1.43** `#define PVErrror_DoLeave() internalLeave __ilv;__ilv.a=0;throw(__ilv)`

**6.5.1.44** `#define PVErrror_IMP_JUMPS`

**6.5.1.45** `#define PERRORTRAP_REGISTRY` [OscTLSRegistry](#)

**6.5.1.46** `#define PERRORTRAP_REGISTRY_ID` [OSCL\\_TLS\\_ID\\_PERRORTRAP](#)

## 6.5.2 Typedef Documentation

**6.5.2.1** `typedef int32` [OscLeaveCode](#)

Leave Codes

**6.5.2.2** `typedef int32` [OscReturnCode](#)

Return Codes

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

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

## 6.5.3 Function Documentation

**6.5.3.1** `OSCL_IMPORT_REF int OSCL_GetLastError ()`

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

### Returns:

Returns 0 for system's that do not have this functionality The value of the error number variable does not change until the user calls `SetLastError` or if another system call occurs that changes the value  
Supported Platforms: Win32/wince, Unix  
Unsupported Platforms : Symbian

**6.5.3.2** `OSCL_IMPORT_REF bool OSCL_IsErrnoSupported ()`

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

### Returns:

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

**6.5.3.3** `OSCL_IMPORT_REF bool OSCL_SetLastError (int newVal)`

This function sets the last error code for the system.

### Parameters:

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

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

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

**Parameters:**

*errnum* This value represents the error number to map

**Returns:**

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

## 6.6 OSCL IO

### Files

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

### Data Structures

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

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

## Defines

- #define `TOscFileOffsetInt32` `int32`
- #define `OSCL_FILE_STATS_LOGGER_NODE` `"OscFileStats"`
- #define `OSCL_IO_FILENAME_MAXLEN` `512`
- #define `OSCL_IO_EXTENSION_MAXLEN` `512`
- #define `OSCL_FILE_WCHAR_PATH_DELIMITER` `_STRLIT("/")`
- #define `OSCL_FILE_CHAR_PATH_DELIMITER` `_STRLIT_CHAR("/")`

## Typedefs

- typedef `oscl_fsstat` `OSCL_FSSTAT`
- typedef `oscl_stat_buf` `OSCL_STAT_BUF`
- typedef `FILE *` `TOscFileHandle`

## Enumerations

- enum `TPVDNSFxn` { `EPVDNSGetHostByName` }
- enum `TPVDNSEvent` { `EPVDNSSuccess`, `EPVDNSPending`, `EPVDNSTimeout`, `EPVDNSFailure`, `EPVDNSCancel` }
- enum `OSCL_FILEMGMT_PERMS` { `OSCL_FILEMGMT_PERMS_READ` = 0x1, `OSCL_FILEMGMT_PERMS_WRITE` = 0x2, `OSCL_FILEMGMT_PERMS_EXECUTE` = 0x4 }
- enum `OSCL_FILEMGMT_MODES` { `OSCL_FILEMGMT_MODE_DIR` = 0x1 }
- enum `OSCL_FILEMGMT_ERR_TYPE` { `OSCL_FILEMGMT_E_OK` = 0, `OSCL_FILEMGMT_E_PATH_TOO_LONG`, `OSCL_FILEMGMT_E_PATH_NOT_FOUND`, `OSCL_FILEMGMT_E_ALREADY_EXISTS`, `OSCL_FILEMGMT_E_NOT_EMPTY`, `OSCL_FILEMGMT_E_PERMISSION_DENIED`, `OSCL_FILEMGMT_E_NO_MATCH`, `OSCL_FILEMGMT_E_UNKNOWN`, `OSCL_FILEMGMT_E_SYS_SPECIFIC`, `OSCL_FILEMGMT_E_NOT_IMPLEMENTED` }
- enum `TOscFileOp` { `EOscFileOp_Open`, `EOscFileOp_Close`, `EOscFileOp_Read`, `EOscFileOp_Write`, `EOscFileOp_Seek`, `EOscFileOp_Tell`, `EOscFileOp_Size`, `EOscFileOp_Flush`, `EOscFileOp_EndOfFile`, `EOscFileOp_SetSize`, `EOscFileOp_NativeOpen`, `EOscFileOp_NativeClose`, `EOscFileOp_NativeRead`, `EOscFileOp_NativeWrite`, `EOscFileOp_NativeSeek`, `EOscFileOp_NativeTell`, `EOscFileOp_NativeSize`, `EOscFileOp_NativeFlush`, `EOscFileOp_NativeEndOfFile`, `EOscFileOp_NativeSetSize`, `EOscFileOp_Last` }

## Functions

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

### 6.6.1 Define Documentation

6.6.1.1 #define OSCL\_FILE\_CHAR\_PATH\_DELIMITER \_STRLIT\_CHAR("/")

6.6.1.2 #define OSCL\_FILE\_STATS\_LOGGER\_NODE "OscFileStats"

6.6.1.3 #define OSCL\_FILE\_WCHAR\_PATH\_DELIMITER \_STRLIT("/")

6.6.1.4 #define OSCL\_IO\_EXTENSION\_MAXLEN 512

6.6.1.5 #define OSCL\_IO\_FILENAME\_MAXLEN 512

6.6.1.6 #define TOscFileOffsetInt32 int32

### 6.6.2 Typedef Documentation

6.6.2.1 typedef struct oscl\_fsstat OSCL\_FSSTAT

6.6.2.2 typedef struct oscl\_stat\_buf OSCL\_STAT\_BUF

6.6.2.3 typedef FILE\* TOscFileHandle

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

### 6.6.3 Enumeration Type Documentation

#### 6.6.3.1 enum OSCL\_FILEMGMT\_ERR\_TYPE

Enumeration values:

- OSCL\_FILEMGMT\_E\_OK
- OSCL\_FILEMGMT\_E\_PATH\_TOO\_LONG
- OSCL\_FILEMGMT\_E\_PATH\_NOT\_FOUND
- OSCL\_FILEMGMT\_E\_ALREADY\_EXISTS
- OSCL\_FILEMGMT\_E\_NOT\_EMPTY
- OSCL\_FILEMGMT\_E\_PERMISSION\_DENIED
- OSCL\_FILEMGMT\_E\_NO\_MATCH
- OSCL\_FILEMGMT\_E\_UNKNOWN
- OSCL\_FILEMGMT\_E\_SYS\_SPECIFIC
- OSCL\_FILEMGMT\_E\_NOT\_IMPLEMENTED

#### 6.6.3.2 enum OSCL\_FILEMGMT\_MODES

Enumeration values:

- OSCL\_FILEMGMT\_MODE\_DIR

#### 6.6.3.3 enum OSCL\_FILEMGMT\_PERMS

Enumeration values:

- OSCL\_FILEMGMT\_PERMS\_READ
- OSCL\_FILEMGMT\_PERMS\_WRITE
- OSCL\_FILEMGMT\_PERMS\_EXECUTE

#### 6.6.3.4 enum TOsclFileOp

Enumeration values:

- EOsclFileOp\_Open
- EOsclFileOp\_Close
- EOsclFileOp\_Read
- EOsclFileOp\_Write
- EOsclFileOp\_Seek
- EOsclFileOp\_Tell
- EOsclFileOp\_Size
- EOsclFileOp\_Flush
- EOsclFileOp\_EndOfFile
- EOsclFileOp\_SetSize
- EOsclFileOp\_NativeOpen
- EOsclFileOp\_NativeClose



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

#### 6.6.3.5 enum TPVDNSEvent

Enumeration values:

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

#### 6.6.3.6 enum TPVDNSFxn

Enumeration values:

**EPVDNSGetHostByName**

### 6.6.4 Function Documentation

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

oscl\_chdir changes the current directory to the path given

**Parameters:**

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

**Returns:**

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

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

oscl\_chdir changes the current directory to the path given

**Parameters:**

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

**Returns:**

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

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

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

#### Parameters:

*pointer* to character buffer to receive the current directory

*size* size of buffer in characters

#### Returns:

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

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

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

#### Parameters:

*pointer* to wide character buffer to receive the current directory

*size* size of buffer in wide characters

#### Returns:

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

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

oscl\_mkdir function creates a directory in the path given

#### Parameters:

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

#### Returns:

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

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

oscl\_mkdir function creates a directory in the path given

#### Parameters:

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

#### Returns:

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

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

oscl\_rmdir removes an empty directory in the path given

##### Parameters:

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

##### Returns:

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

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

oscl\_rename function renames a file or directory

##### Parameters:

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

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

##### Returns:

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

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

oscl\_rmdir removes an empty directory in the path given

##### Parameters:

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

##### Returns:

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

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

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

##### Parameters:

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

##### Returns:

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

#### 6.6.4.11 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\_statfs (OSCL\_FSSTAT \* stats, const oscl\_wchar \* path)

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

##### Parameters:

*stats* pointer to structure to hold information

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

##### Returns:

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

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

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

##### Parameters:

*stats* pointer to structure to hold information

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

##### Returns:

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

## 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 [OscTimerQ](#)
- class [PVActiveBase](#)
- class [PVActiveStats](#)
- class [PVSchedulerStopper](#)
- class [PVThreadContext](#)
- class [TReadyQueLink](#)

## Defines

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

## Typedefs

- typedef [PVActiveBase](#) \* [TOscReady](#)

## Enumerations

- enum [TPVThreadContext](#) { [EPVThreadContext\\_InThread](#), [EPVThreadContext\\_OscThread](#), [EPVThreadContext\\_NonOscThread](#), [EPVThreadContext\\_Undetermined](#) }

## Functions

- template<class T, class S> T \* [OscPtrAdd](#) (T \*aPtr, S aVal)
- template<class T, class S> T \* [OscPtrSub](#) (T \*aPtr, S aVal)

## Variables

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

## 6.7.1 Define Documentation

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

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

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

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

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

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

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

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

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

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

**6.7.1.10 #define PVEXECNAMELEN 30**

**6.7.1.11 #define PVSCHEDNAMELEN 30**

PV Scheduler class

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

**Value:**

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

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

**Value:**

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

## 6.7.2 Typedef Documentation

6.7.2.1 typedef **PVActiveBase\*** TOsclReady

## 6.7.3 Enumeration Type Documentation

6.7.3.1 enum TPVThreadContext

Thread context type

Enumeration values:

EPVThreadContext\_InThread

EPVThreadContext\_OsclThread

EPVThreadContext\_NonOsclThread

EPVThreadContext\_Undetermined

## 6.7.4 Function Documentation

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

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

## 6.7.5 Variable Documentation

6.7.5.1 const int32 OSCL\_REQUEST\_ERR\_CANCEL = (-1)

6.7.5.2 const int32 OSCL\_REQUEST\_ERR\_GENERAL = (-2)

6.7.5.3 const int32 OSCL\_REQUEST\_ERR\_NONE = 0

6.7.5.4 const int32 OSCL\_REQUEST\_PENDING = (-0x7fffffff)



## 6.8 OSCL Init

### Files

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

### Data Structures

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

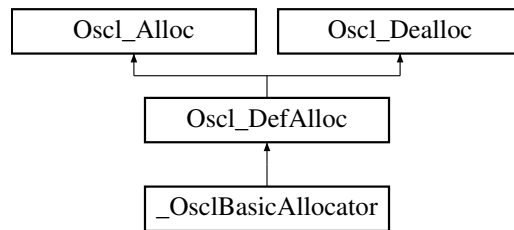
## Chapter 7

# oscl Data Structure Documentation

### 7.1 `_OscBasicAllocator` Class Reference

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

Inheritance diagram for `_OscBasicAllocator`:



#### Public Methods

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

#### 7.1.1 Detailed Description

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

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

## 7.1.2 Constructor & Destructor Documentation

**7.1.2.1** `virtual _OscBasicAllocator::~~_OscBasicAllocator ()` [inline, virtual]

## 7.1.3 Member Function Documentation

**7.1.3.1** `OscAny* _OscBasicAllocator::allocate (const uint32 size)` [inline, virtual]

Implements [Osc\\_DefAlloc](#).

**7.1.3.2** `void _OscBasicAllocator::deallocate (OscAny *p)` [inline, virtual]

Implements [Osc\\_DefAlloc](#).

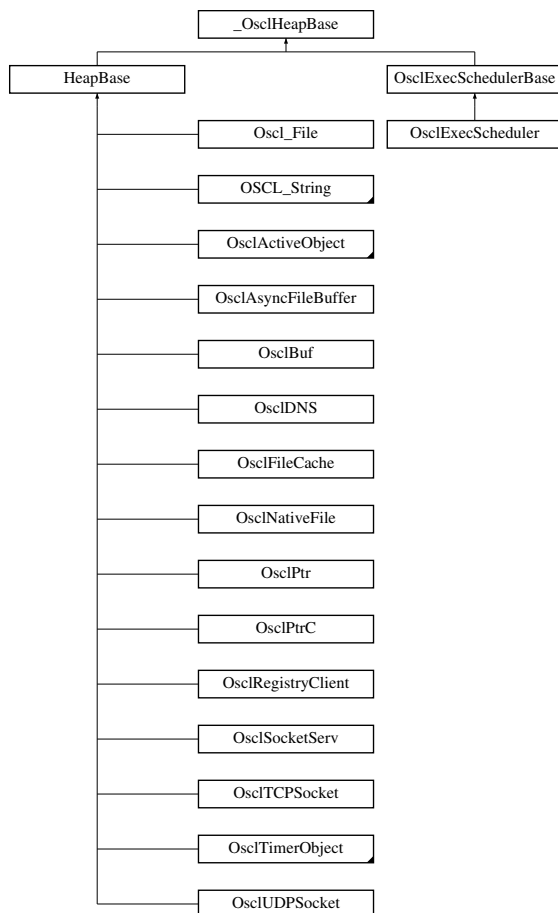
The documentation for this class was generated from the following file:

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

## 7.2 \_OscHeapBase Class Reference

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

Inheritance diagram for \_OscHeapBase::



### Public Methods

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

### Protected Methods

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

### Friends

- class [PVCleanupStack](#)

### 7.2.1 Detailed Description

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

### 7.2.2 Constructor & Destructor Documentation

**7.2.2.1** `virtual _OscHeapBase::~~_OscHeapBase ()` [inline, virtual]

**7.2.2.2** `_OscHeapBase::_OscHeapBase ()` [inline, protected]

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

### 7.2.3 Friends And Related Function Documentation

**7.2.3.1** `friend class PVCleanupStack` [friend]

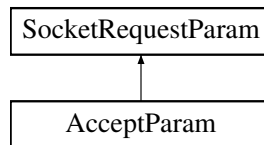
The documentation for this class was generated from the following file:

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

## 7.3 AcceptParam Class Reference

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

Inheritance diagram for AcceptParam::



### Public Methods

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

### Data Fields

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

### 7.3.1 Constructor & Destructor Documentation

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

### 7.3.2 Field Documentation

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

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

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

## 7.4 allocator Class Reference

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

### 7.4.1 Detailed Description

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

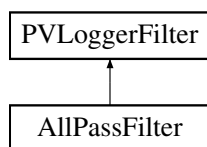
The documentation for this class was generated from the following file:

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

## 7.5 AllPassFilter Class Reference

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

Inheritance diagram for AllPassFilter::



### Public Types

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

### Public Methods

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

### 7.5.1 Detailed Description

Example filter that allows all messages to be logged.

### 7.5.2 Member Typedef Documentation

#### 7.5.2.1 typedef PVLoggerFilter::filter\_status\_type AllPassFilter::filter\_status\_type

Reimplemented from [PVLoggerFilter](#).

#### 7.5.2.2 typedef PVLoggerFilter::log\_level\_type AllPassFilter::log\_level\_type

Reimplemented from [PVLoggerFilter](#).

#### 7.5.2.3 typedef PVLoggerFilter::message\_id\_type AllPassFilter::message\_id\_type

Reimplemented from [PVLoggerFilter](#).



### 7.5.3 Constructor & Destructor Documentation

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

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

### 7.5.4 Member Function Documentation

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

Implements [PVLoggerFilter](#).

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

Implements [PVLoggerFilter](#).

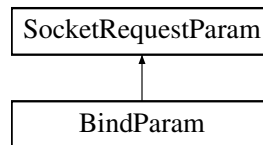
The documentation for this class was generated from the following file:

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

## 7.6 BindParam Class Reference

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

Inheritance diagram for BindParam::



### Public Methods

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

### Data Fields

- [OscNetworkAddress](#) iAddr

### 7.6.1 Constructor & Destructor Documentation

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

### 7.6.2 Field Documentation

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

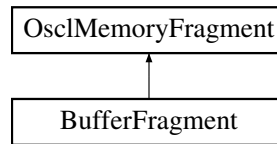
The documentation for this class was generated from the following file:

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

## 7.7 BufferFragment Class Reference

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

Inheritance diagram for BufferFragment::



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

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

## 7.8 BufferMgr Class Reference

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

### Public Methods

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

### 7.8.1 Constructor & Destructor Documentation

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

### 7.8.2 Member Function Documentation

**7.8.2.1** virtual void [BufferMgr::BufferReleased](#) (void \* *ptr*, [BufferState](#) \* *state* = NULL) [pure virtual]

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

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

## 7.9 BufferState Class Reference

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

### Public Methods

- [BufferState](#) ([BufferFreeFuncPtr](#) the\_free\_function, void \*bufptr=0)
- [BufferState](#) ([BufferMgr](#) \*the\_buf\_mgr=0, void \*bufptr=0)
- void [increment\\_refcnt](#) ()
- void [decrement\\_refcnt](#) ()
- void [bind](#) (void \*in\_ptr, [BufferFreeFuncPtr](#) in\_free\_function)
- void [bind](#) (void \*in\_ptr, [BufferMgr](#) \*in\_buf\_mgr)
- void \* [get\\_ptr](#) ()
- int32 [get\\_refcount](#) ()
- [BufferFreeFuncPtr](#) [get\\_free\\_function](#) ()
- [BufferMgr](#) \* [get\\_buf\\_mgr](#) ()
- void [reset](#) ()

### 7.9.1 Constructor & Destructor Documentation

**7.9.1.1** [BufferState::BufferState](#) ([BufferFreeFuncPtr](#) the\_free\_function, void \* bufptr = 0)  
[inline]

**7.9.1.2** [BufferState::BufferState](#) ([BufferMgr](#) \* the\_buf\_mgr = 0, void \* bufptr = 0) [inline]

### 7.9.2 Member Function Documentation

**7.9.2.1** void [BufferState::bind](#) (void \* in\_ptr, [BufferMgr](#) \* in\_buf\_mgr) [inline]

**7.9.2.2** void [BufferState::bind](#) (void \* in\_ptr, [BufferFreeFuncPtr](#) in\_free\_function) [inline]

**7.9.2.3** void [BufferState::decrement\\_refcnt](#) () [inline]

**7.9.2.4** [BufferMgr](#)\* [BufferState::get\\_buf\\_mgr](#) () [inline]

**7.9.2.5** [BufferFreeFuncPtr](#) [BufferState::get\\_free\\_function](#) () [inline]

**7.9.2.6** void\* [BufferState::get\\_ptr](#) () [inline]

**7.9.2.7** int32 [BufferState::get\\_refcount](#) () [inline]

**7.9.2.8** void [BufferState::increment\\_refcnt](#) () [inline]

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

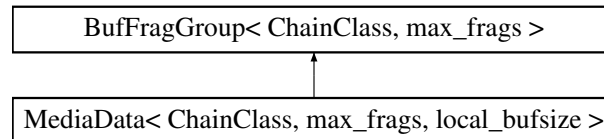
The documentation for this class was generated from the following file:

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

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

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

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



### Public Methods

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

### Protected Methods

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

### Protected Attributes

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

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

## 7.10.1 Constructor & Destructor Documentation

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

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

## 7.10.2 Member Function Documentation

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

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

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

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

**7.10.2.4** `template<class ChainClass, uint32 max_frgs> uint32 BufFragGroup< ChainClass, max_frgs >::GetLength () const [inline]`

**7.10.2.5** `template<class ChainClass, uint32 max_frgs> int32 BufFragGroup< ChainClass, max_frgs >::GetMaxFrgs () const [inline]`

**7.10.2.6** `template<class ChainClass, uint32 max_frgs> ChainClass* BufFragGroup< ChainClass, max_frgs >::GetNext () const [inline]`

**7.10.2.7** `template<class ChainClass, uint32 max_frgs> int32 BufFragGroup< ChainClass, max_frgs >::GetNumFrgs () const [inline]`

### 7.10.3 Field Documentation

**7.10.3.1** `template<class ChainClass, uint32 max_frgs> BufferState* BufFragGroup< ChainClass, max_frgs >::buffer_states[max_frgs] [protected]`

**7.10.3.2** `template<class ChainClass, uint32 max_frgs> BufferFragment BufFragGroup< ChainClass, max_frgs >::fragments[max_frgs] [protected]`

**7.10.3.3** `template<class ChainClass, uint32 max_frgs> uint32 BufFragGroup< ChainClass, max_frgs >::length [protected]`

**7.10.3.4** `template<class ChainClass, uint32 max_frgs> ChainClass* BufFragGroup< ChainClass, max_frgs >::next [protected]`

**7.10.3.5** `template<class ChainClass, uint32 max_frgs> uint32 BufFragGroup< ChainClass, max_frgs >::num_fragments [protected]`

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

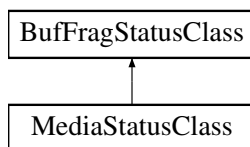
- [oscl\\_media\\_data.h](#)



## 7.11 BufFragStatusClass Class Reference

```
#include <oscl_media_status.h>
```

Inheritance diagram for BufFragStatusClass::



### Public Types

- enum `status_t` { `BFG_SUCCESS` = 0, `TOO_MANY_FRAGS` = 1, `NOT_ENOUGH_SPACE` = 2, `EMPTY_FRAGMENT` = 3, `NULL_INPUT` = 4, `FIXED_FRAG_LOC_FULL` = 5, `INTERNAL_ERROR`, `INVALID_ID` }

### 7.11.1 Member Enumeration Documentation

#### 7.11.1.1 enum BufFragStatusClass::status\_t

Enumeration values:

**BFG\_SUCCESS**  
**TOO\_MANY\_FRAGS**  
**NOT\_ENOUGH\_SPACE**  
**EMPTY\_FRAGMENT**  
**NULL\_INPUT**  
**FIXED\_FRAG\_LOC\_FULL**  
**INTERNAL\_ERROR**  
**INVALID\_ID**

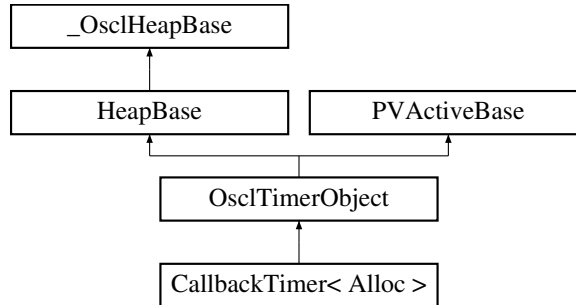
The documentation for this class was generated from the following file:

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

## 7.12 CallbackTimer< Alloc > Class Template Reference

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

Inheritance diagram for CallbackTimer< Alloc >::



### Public Methods

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

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

### 7.12.2 Member Function Documentation

**7.12.2.1** `template<class Alloc> void CallbackTimer< Alloc >::Run ()` [inline, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's [Run\(\)](#) function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

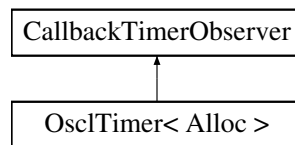
The documentation for this class was generated from the following file:

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

## 7.13 CallbackTimerObserver Class Reference

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

Inheritance diagram for CallbackTimerObserver::



### Public Methods

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

### 7.13.1 Constructor & Destructor Documentation

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

### 7.13.2 Member Function Documentation

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

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

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

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

## 7.14 CFastRep Class Reference

```
#include <oscl_string_rep.h>
```

### Public Methods

- [CFastRep](#) ()
- OSCL\_IMPORT\_REF void [set\\_w](#) (char \*cp, uint32 len, uint32 maxlen)
- OSCL\_IMPORT\_REF void [set\\_w](#) ([oscl\\_wchar](#) \*cp, uint32 len, uint32 maxlen)
- OSCL\_IMPORT\_REF void [set\\_r](#) (const char \*cp, uint32 len)
- OSCL\_IMPORT\_REF void [set\\_r](#) (const [oscl\\_wchar](#) \*cp, uint32 len)
- OSCL\_IMPORT\_REF void [append](#) (const char \*cp, uint32 len)
- OSCL\_IMPORT\_REF void [append](#) (const [oscl\\_wchar](#) \*cp, uint32 len)

### Data Fields

- uint32 [maxsize](#)
- uint32 [size](#)
- [OsclAny](#) \* [buffer](#)
- bool [writable](#)
- bool [overwrite](#)

### 7.14.1 Detailed Description

For internal use only– fast string representation

## 7.14.2 Constructor & Destructor Documentation

7.14.2.1 CFastRep::CFastRep() [inline]

## 7.14.3 Member Function Documentation

7.14.3.1 OSCL\_IMPORT\_REF void CFastRep::append (const [oscl\\_wchar](#) \* *cp*, *uint32 len*)

7.14.3.2 OSCL\_IMPORT\_REF void CFastRep::append (const char \* *cp*, *uint32 len*)

7.14.3.3 OSCL\_IMPORT\_REF void CFastRep::set\_r (const [oscl\\_wchar](#) \* *cp*, *uint32 len*)

7.14.3.4 OSCL\_IMPORT\_REF void CFastRep::set\_r (const char \* *cp*, *uint32 len*)

7.14.3.5 OSCL\_IMPORT\_REF void CFastRep::set\_w ([oscl\\_wchar](#) \* *cp*, *uint32 len*, *uint32 maxlen*)

7.14.3.6 OSCL\_IMPORT\_REF void CFastRep::set\_w (char \* *cp*, *uint32 len*, *uint32 maxlen*)

## 7.14.4 Field Documentation

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

7.14.4.2 *uint32* CFastRep::maxsize

7.14.4.3 *bool* CFastRep::overwrite

7.14.4.4 *uint32* CFastRep::size

7.14.4.5 *bool* CFastRep::writable

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

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

## 7.15 CHeapRep Class Reference

```
#include <oscl_string_rep.h>
```

### Public Methods

- [CHeapRep](#) ()
- OSCL\_IMPORT\_REF bool [set](#) (uint32, const char \*, [OscDefAlloc](#) &)
- OSCL\_IMPORT\_REF bool [set](#) (uint32, const [oscl\\_wchar](#) \*, [OscDefAlloc](#) &)
- OSCL\_IMPORT\_REF bool [append](#) (uint32, const char \*, uint32, const char \*, [OscDefAlloc](#) &)
- OSCL\_IMPORT\_REF bool [append](#) (uint32, const [oscl\\_wchar](#) \*, uint32, const [oscl\\_wchar](#) \*, [OscDefAlloc](#) &)
- OSCL\_IMPORT\_REF void [add\\_ref](#) ()
- OSCL\_IMPORT\_REF void [remove\\_ref](#) ([OscDefAlloc](#) &)

### Static Public Methods

- OSCL\_IMPORT\_REF void [set\\_rep](#) (CHeapRep \*&, [OscDefAlloc](#) &, const char \*, uint32)
- OSCL\_IMPORT\_REF void [set\\_rep](#) (CHeapRep \*&, [OscDefAlloc](#) &, const [oscl\\_wchar](#) \*, uint32)
- OSCL\_IMPORT\_REF void [append\\_rep](#) (CHeapRep \*&, [OscDefAlloc](#) &, const char \*, uint32)
- OSCL\_IMPORT\_REF void [append\\_rep](#) (CHeapRep \*&, [OscDefAlloc](#) &, const [oscl\\_wchar](#) \*, uint32)
- OSCL\_IMPORT\_REF void [assign](#) (CHeapRep \*&, CHeapRep \*, [OscDefAlloc](#) &)

### Data Fields

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

### 7.15.1 Detailed Description

For internal use only– heap string representation

## 7.15.2 Constructor & Destructor Documentation

7.15.2.1 CHeapRep::CHeapRep () [inline]

## 7.15.3 Member Function Documentation

7.15.3.1 OSCL\_IMPORT\_REF void CHeapRep::add\_ref ()

7.15.3.2 OSCL\_IMPORT\_REF bool CHeapRep::append (uint32, const [oscl\\_wchar](#) \*, uint32, const [oscl\\_wchar](#) \*, [OscDefAlloc](#) &)

7.15.3.3 OSCL\_IMPORT\_REF bool CHeapRep::append (uint32, const char \*, uint32, const char \*, [OscDefAlloc](#) &)

7.15.3.4 OSCL\_IMPORT\_REF void CHeapRep::append\_rep (CHeapRep \* &, [OscDefAlloc](#) &, const [oscl\\_wchar](#) \*, uint32) [static]

7.15.3.5 OSCL\_IMPORT\_REF void CHeapRep::append\_rep (CHeapRep \* &, [OscDefAlloc](#) &, const char \*, uint32) [static]

7.15.3.6 OSCL\_IMPORT\_REF void CHeapRep::assign (CHeapRep \* &, CHeapRep \*, [OscDefAlloc](#) &) [static]

7.15.3.7 OSCL\_IMPORT\_REF void CHeapRep::remove\_ref ([OscDefAlloc](#) &)

7.15.3.8 OSCL\_IMPORT\_REF bool CHeapRep::set (uint32, const [oscl\\_wchar](#) \*, [OscDefAlloc](#) &)

7.15.3.9 OSCL\_IMPORT\_REF bool CHeapRep::set (uint32, const char \*, [OscDefAlloc](#) &)

7.15.3.10 OSCL\_IMPORT\_REF void CHeapRep::set\_rep (CHeapRep \* &, [OscDefAlloc](#) &, const [oscl\\_wchar](#) \*, uint32) [static]

7.15.3.11 OSCL\_IMPORT\_REF void CHeapRep::set\_rep (CHeapRep \* &, [OscDefAlloc](#) &, const char \*, uint32) [static]

## 7.15.4 Field Documentation

7.15.4.1 [OscAny](#)\* CHeapRep::buffer

7.15.4.2 uint32 CHeapRep::maxsize

7.15.4.3 uint32 CHeapRep::refcount

7.15.4.4 uint32 CHeapRep::size

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

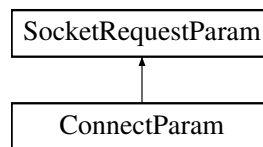
- [oscl\\_string\\_rep.h](#)



## 7.16 ConnectParam Class Reference

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

Inheritance diagram for ConnectParam::



### Public Methods

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

### Data Fields

- [OscNetworkAddress](#) iAddr

### 7.16.1 Constructor & Destructor Documentation

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

### 7.16.2 Field Documentation

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

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

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

## 7.17 CStackRep Class Reference

```
#include <oscl_string_rep.h>
```

### Public Methods

- [CStackRep](#) ()
- OSCL\_IMPORT\_REF void [set](#) (const char \*cp, uint32 len)
- OSCL\_IMPORT\_REF void [set](#) (const [oscl\\_wchar](#) \*cp, uint32 len)
- OSCL\_IMPORT\_REF void [append](#) (const char \*cp, uint32 len)
- OSCL\_IMPORT\_REF void [append](#) (const [oscl\\_wchar](#) \*cp, uint32 len)

### Data Fields

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

### 7.17.1 Detailed Description

For internal use only– stack string representation

### 7.17.2 Constructor & Destructor Documentation

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

### 7.17.3 Member Function Documentation

7.17.3.1 OSCL\_IMPORT\_REF void [CStackRep::append](#) (const [oscl\\_wchar](#) \* cp, uint32 len)

7.17.3.2 OSCL\_IMPORT\_REF void [CStackRep::append](#) (const char \* cp, uint32 len)

7.17.3.3 OSCL\_IMPORT\_REF void [CStackRep::set](#) (const [oscl\\_wchar](#) \* cp, uint32 len)

7.17.3.4 OSCL\_IMPORT\_REF void [CStackRep::set](#) (const char \* cp, uint32 len)

### 7.17.4 Field Documentation

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

7.17.4.2 uint32 [CStackRep::maxsize](#)

7.17.4.3 uint32 [CStackRep::size](#)

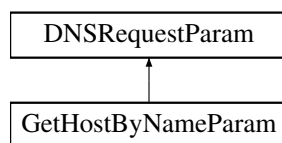
The documentation for this class was generated from the following file:

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

## 7.18 DNSRequestParam Class Reference

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

Inheritance diagram for DNSRequestParam::



### Public Methods

- virtual [~DNSRequestParam](#) ()
- void [RemoveRef](#) ()
- void [InThread](#) ()
- virtual void [Destroy](#) ()=0

### Data Fields

- [TPVDNSFxn](#) iFxn
- [OscDNSRequest](#) \* iDNSRequest

### Protected Methods

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

### Protected Attributes

- uint32 [iRefCount](#)

### 7.18.1 Constructor & Destructor Documentation

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

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

### 7.18.2 Member Function Documentation

**7.18.2.1** virtual void DNSRequestParam::Destroy () [pure virtual]

Implemented in [GetHostByNameParam](#).

**7.18.2.2**   `void DNSRequestParam::InThread ()`

**7.18.2.3**   `void DNSRequestParam::RemoveRef ()`

### **7.18.3   Field Documentation**

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

**7.18.3.2**   [TPVDNSFxn](#) `DNSRequestParam::iFxn`

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

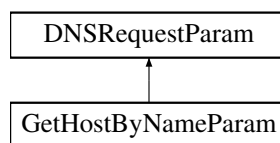
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_param.h](#)

## 7.19 GetHostByNameParam Class Reference

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

Inheritance diagram for GetHostByNameParam::



### Public Types

- enum { [addressListCapacity](#) = 10 }

### Public Methods

- void [Destroy](#) ()
- [~GetHostByNameParam](#) ()
- void [PersistHostAddress](#) (const [OscNetworkAddress](#) &addr)
- bool [canPersistMoreHostAddresses](#) ()

### Static Public Methods

- [GetHostByNameParam](#) \* [Create](#) (const char \*name, [OscNetworkAddress](#) \*&addr, [Osc\\_Vector](#)<[OscNetworkAddress](#), [OscMemAllocator](#)> \*aAddressList)

### Data Fields

- char \* [iName](#)
- [OscNetworkAddress](#) \* [iAddr](#)
- [Osc\\_Vector](#)<[OscNetworkAddress](#), [OscMemAllocator](#)> \* [iAddressList](#)

### 7.19.1 Member Enumeration Documentation

#### 7.19.1.1 anonymous enum

Enumeration values:

[addressListCapacity](#)

## 7.19.2 Constructor & Destructor Documentation

7.19.2.1 `GetHostByNameParam::~GetHostByNameParam ()`

## 7.19.3 Member Function Documentation

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

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

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

Implements [DNSRequestParam](#).

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

## 7.19.4 Field Documentation

7.19.4.1 `OsciNetworkAddress* GetHostByNameParam::iAddr`

7.19.4.2 `Osci\_Vector<OsciNetworkAddress, OsciMemAllocator>* GetHostByNameParam::i-AddressList`

7.19.4.3 `char* GetHostByNameParam::iName`

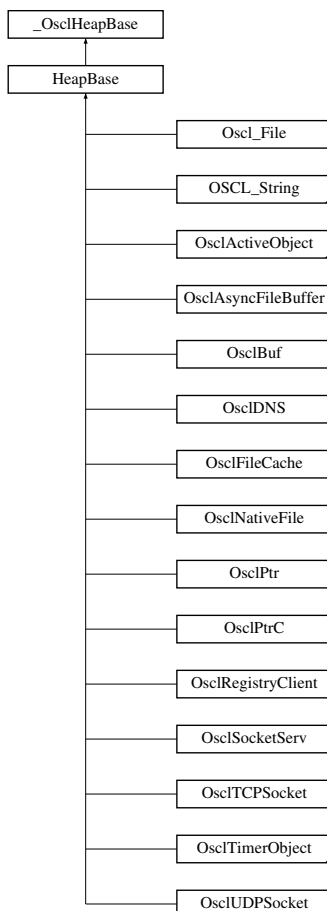
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_param.h](#)

## 7.20 HeapBase Class Reference

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

Inheritance diagram for HeapBase::



### Public Methods

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

### 7.20.1 Detailed Description

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

HeapBase has overloaded new and delete operators.

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

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

## 7.20.2 Constructor & Destructor Documentation

**7.20.2.1 HeapBase::HeapBase ()** [inline]

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

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

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



## 7.21 internalLeave Class Reference

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

### Data Fields

- int [a](#)

### 7.21.1 Field Documentation

#### 7.21.1.1 int internalLeave::a

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

- [oscl\\_error\\_imp\\_cppexceptions.h](#)

## 7.22 LinkedListElement< LLClass > Class Template Reference

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

### Public Methods

- [LinkedListElement](#) (LLClass in\_data)

### Data Fields

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

#### 7.22.1 Detailed Description

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

Linked List Element Class

#### 7.22.2 Constructor & Destructor Documentation

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

#### 7.22.3 Field Documentation

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

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

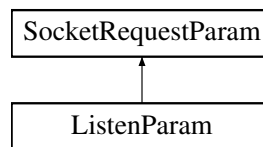
The documentation for this class was generated from the following file:

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

## 7.23 ListenParam Class Reference

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

Inheritance diagram for ListenParam::



### Public Methods

- [ListenParam](#) (uint32 aSize)

### Data Fields

- uint32 [iQSize](#)

### 7.23.1 Constructor & Destructor Documentation

**7.23.1.1** `ListenParam::ListenParam (uint32 aSize) [inline]`

### 7.23.2 Field Documentation

**7.23.2.1** `uint32 ListenParam::iQSize`

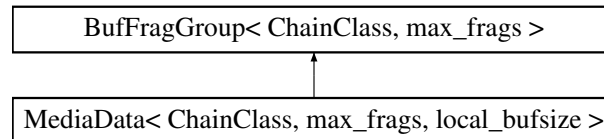
The documentation for this class was generated from the following file:

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

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

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

Inheritance diagram for `MediaData< ChainClass, max_frgs, local_bufsize >::`



### Public Methods

- `MediaData ()`
- `virtual ~MediaData ()`
- `uint32 GetLocalBufsize () const`
- `MediaTimestamp GetTimestamp () const`
- `void SetTimestamp (MediaTimestamp in_timestamp)`
- `uint32 GetAvailableBufferSize () const`
- `MediaStatusClass::status_t GetLocalFragment (BufferFragment &fragment)`
- `virtual void Clear ()`
- `bool IsLocalData (const OsciMemoryFragment &frag) const`
- `int GetMediaSize () const`
- `BufferFragment * GetMediaFragment (const uint32 idx)`
- `uint32 GetNumMediaFrgs (const uint32 idx) const`

### Protected Methods

- `MediaStatusClass::status_t AddLocalFragment (const BufferFragment &frag, int32 location_offset)`

### Protected Attributes

- `MediaTimestamp timestamp`
- `uint8 localbuf [local_bufsize]`
- `uint32 available_localbuf`
- `int num_reserved_fragments`

```
template<class ChainClass, uint32 max_frgs, uint32 local_bufsize> class MediaData< ChainClass,
max_frgs, local_bufsize >
```

### 7.24.1 Constructor & Destructor Documentation

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

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

### 7.24.2 Member Function Documentation

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

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

Reimplemented from `BufFragGroup< ChainClass, max_frgs >`.

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

### 7.24.3 Field Documentation

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

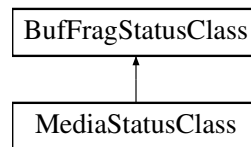
The documentation for this class was generated from the following file:

- [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\_frgs=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_frgs = 1)` [pure virtual]

7.26.3.2 `template<class T> virtual void MemAllocator< T >::deallocate (pointer p)` [pure virtual]

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

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



## 7.27 MM\_AllocBlockFence Struct Reference

```
#include <oscl_mem_audit_internals.h>
```

### Public Methods

- [MM\\_AllocBlockFence](#) ()
- void [fill\\_fence](#) ()
- bool [check\\_fence](#) ()

### Data Fields

- uint8 [pad](#) [COMPUTE\_MEM\_ALIGN\_SIZE(sizeof([MM\\_AllocBlockHdr](#)), MIN\_FENCE\_SIZE, MEM\_ALIGN\_SIZE)]

### 7.27.1 Constructor & Destructor Documentation

7.27.1.1 [MM\\_AllocBlockFence::MM\\_AllocBlockFence](#) () [inline]

### 7.27.2 Member Function Documentation

7.27.2.1 bool [MM\\_AllocBlockFence::check\\_fence](#) () [inline]

7.27.2.2 void [MM\\_AllocBlockFence::fill\\_fence](#) () [inline]

### 7.27.3 Field Documentation

7.27.3.1 uint8 [MM\\_AllocBlockFence::pad](#)[COMPUTE\_MEM\_ALIGN\_SIZE(sizeof([MM\\_AllocBlockHdr](#)), MIN\_FENCE\_SIZE, MEM\_ALIGN\_SIZE)]

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

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

## 7.28 MM\_AllocBlockHdr Struct Reference

```
#include <oscl_mem_audit_internals.h>
```

### Public Methods

- bool [isAllocNodePtr](#) ()
- void [setAllocNodeFlag](#) ()
- [MM\\_AllocBlockHdr](#) ()
- [MM\\_AllocBlockHdr](#) (void \*ptr, uint32 inSize)

### Data Fields

- void \* [pNode](#)
- uint32 [size](#)
- void \* [pRootNode](#)
- uint32 [pad](#)

### Static Public Attributes

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

### 7.28.1 Constructor & Destructor Documentation

7.28.1.1 [MM\\_AllocBlockHdr::MM\\_AllocBlockHdr](#) () [inline]

7.28.1.2 [MM\\_AllocBlockHdr::MM\\_AllocBlockHdr](#) (void \**ptr*, uint32 *inSize*) [inline]

### 7.28.2 Member Function Documentation

7.28.2.1 bool [MM\\_AllocBlockHdr::isAllocNodePtr](#) () [inline]

7.28.2.2 void [MM\\_AllocBlockHdr::setAllocNodeFlag](#) () [inline]

### 7.28.3 Field Documentation

7.28.3.1 uint32 [MM\\_AllocBlockHdr::pad](#)

7.28.3.2 void\* [MM\\_AllocBlockHdr::pNode](#)

7.28.3.3 void\* [MM\\_AllocBlockHdr::pRootNode](#)

7.28.3.4 uint32 [MM\\_AllocBlockHdr::size](#)

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

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

## 7.29 MM\_AllocInfo Struct Reference

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

### Public Methods

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

### Data Fields

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

## 7.29.1 Constructor & Destructor Documentation

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

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

## 7.29.2 Member Function Documentation

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

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

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

## 7.29.3 Field Documentation

7.29.3.1 `uint32 MM_AllocInfo::allocNum`

7.29.3.2 `bool MM_AllocInfo::bSetFailure`

7.29.3.3 `uint32 MM_AllocInfo::lineNo`

7.29.3.4 `char* MM_AllocInfo::pFileName`

7.29.3.5 `void* MM_AllocInfo::pMemBlock`

7.29.3.6 `OsclMemStatsNode* MM_AllocInfo::pStatsNode`

7.29.3.7 `uint32 MM_AllocInfo::size`

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

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

## 7.30 MM\_AllocNode Struct Reference

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

### Public Methods

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

### Data Fields

- [MM\\_AllocInfo](#) \* [pAllocInfo](#)
- MM\_AllocNode \* [pPrev](#)
- MM\_AllocNode \* [pNext](#)

### 7.30.1 Constructor & Destructor Documentation

7.30.1.1 [MM\\_AllocNode::MM\\_AllocNode](#) () [inline]

7.30.1.2 [MM\\_AllocNode::~~MM\\_AllocNode](#) () [inline]

### 7.30.2 Member Function Documentation

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

7.30.2.2 void\* [MM\\_AllocNode::operator new](#) ([oscl\\_memsize\\_t](#) *size*, MM\_AllocNode \**ptr*) [inline]

7.30.2.3 void\* [MM\\_AllocNode::operator new](#) ([oscl\\_memsize\\_t](#) *size*) [inline]

### 7.30.3 Field Documentation

7.30.3.1 [MM\\_AllocInfo](#)\* [MM\\_AllocNode::pAllocInfo](#)

7.30.3.2 MM\_AllocNode\* [MM\\_AllocNode::pNext](#)

7.30.3.3 MM\_AllocNode\* [MM\\_AllocNode::pPrev](#)

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

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

## 7.31 MM\_AllocQueryInfo Struct Reference

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

### Data Fields

- uint32 [allocNum](#)
- char [fileName](#) [MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN]
- uint32 [lineNo](#)
- uint32 [size](#)
- const void \* [pMemBlock](#)
- char [tag](#) [MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN]

### 7.31.1 Field Documentation

**7.31.1.1**    uint32 MM\_AllocQueryInfo::allocNum

**7.31.1.2**    char MM\_AllocQueryInfo::fileName[MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN]

**7.31.1.3**    uint32 MM\_AllocQueryInfo::lineNo

**7.31.1.4**    const void\* MM\_AllocQueryInfo::pMemBlock

**7.31.1.5**    uint32 MM\_AllocQueryInfo::size

**7.31.1.6**    char MM\_AllocQueryInfo::tag[MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN]

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

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

## 7.32 MM\_Audit\_Imp Class Reference

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

### Public Methods

- [MM\\_Audit\\_Imp](#) ()
- [~MM\\_Audit\\_Imp](#) ()
- OSCL\_IMPORT\_REF void \* [MM\\_allocate](#) (const [OscMemStatsNode](#) \*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\\_GetTreeNodees](#) (const char \*tagIn)
- OSCL\_IMPORT\_REF bool [MM\\_AddTag](#) (const char \*tagIn)
- OSCL\_IMPORT\_REF const [OscMemStatsNode](#) \* [MM\\_GetTagNode](#) (const char \*tagIn)
- OSCL\_IMPORT\_REF const [OscMemStatsNode](#) \* [MM\\_GetExistingTag](#) (const char \*tagIn)
- OSCL\_IMPORT\_REF const [OscMemStatsNode](#) \* [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, [OscMemStatsNode](#) \*pStatsNode, const char \*pFileName, uint32 lineNumber)
- [OscMemStatsNode](#) \* [removeAllocNode](#) (void \*pMemBlockIn, uint32 &size)
- void [removeALLAllocNodes](#) ()
- [OscMemStatsNode](#) \* [createStatsNode](#) (const char \*tagIn)
- bool [updateStatsNode](#) ([OscMemStatsNode](#) \*pCurrStatsNode, const [MM\\_Stats\\_t](#) &pDelta, bool b-Add)
- bool [updateStatsNodeInFailure](#) (const char \*tagIn)
- bool [updateStatsNodeInFailure](#) ([OscMemStatsNode](#) \*pStatsNode)
- bool [pruneSubtree](#) ([OscMemStatsNode](#) \*pNode)
- bool [pruneSubtree](#) (const char \*tagIn)
- void [retrieveParentTag](#) (char \*tag)
- int32 [retrieveParentTagLength](#) (const char \*tag, int32 bound)
- void [makeValidTag](#) (const char \*tagIn, [MMAuditCharAutoPtr](#) &autoptr)

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

## Static Public Methods

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

## 7.32.1 Constructor & Destructor Documentation

### 7.32.1.1 MM\_Audit\_Imp::MM\_Audit\_Imp ()

Constructor, create the root node in statistics table

### 7.32.1.2 MM\_Audit\_Imp::~MM\_Audit\_Imp ()

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

## 7.32.2 Member Function Documentation

### 7.32.2.1 [MM\\_AllocNode](#)\* MM\_Audit\_Imp::addAllocNode (void \*pMem, uint32 sizeIn, [OscMemStatsNode](#) \*pStatsNode, const char \*pFileName, uint32 lineNumber)

#### Returns:

true if operation succeeds;

### 7.32.2.2 [OscMemStatsNode](#)\* MM\_Audit\_Imp::createStatsNode (const char \*tagIn)

#### Returns:

true if operation succeeds;

### 7.32.2.3 [OscMemAudit](#)\* MM\_Audit\_Imp::getAuditRoot (void \*ptrIn) [static]

#### Returns:

audit root pointer.

### 7.32.2.4 uint32 MM\_Audit\_Imp::getSize (void \*ptrIn) [static]

#### Returns:

original block size. leaves if bad pointer.



### 7.32.2.5 uint32 MM\_Audit\_Imp::getTagActualSize (const char \* *tagIn*)

#### Returns:

the size of the truncated tag; 0 means NO truncation

### 7.32.2.6 bool MM\_Audit\_Imp::isSetFailure (OscMemStatsNode \* *statsNode*)

### 7.32.2.7 bool MM\_Audit\_Imp::isSetFailure (const char \* *tagIn*)

#### Returns:

true if operation succeeds;

### 7.32.2.8 void MM\_Audit\_Imp::makeValidTag (const char \* *tagIn*, MMAuditCharAutoPtr & *autoPtr*)

#### Returns:

a valid tag; NULL will be converted into root tag

### 7.32.2.9 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 OscMemStatsNode \* *statsNode*, uint32 *sizeIn*, const char \* *pFileName*, uint32 *lineNumber*, bool *allocNodeTracking* = false)

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

#### Returns:

the memory pointer if operation succeeds.

### 7.32.2.11 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 OsciMemStatsNode\* MM\_Audit\_Imp::MM\_GetExisting-Tag (const char \* tagIn)

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

#### Parameters:

*tagIn* input tag

#### Returns:

true if operation succeeds;

### 7.32.2.16 uint32 MM\_Audit\_Imp::MM\_GetMode (void) [inline]

API to get the operating mode of the mm\_audit class.

### 7.32.2.17 uint32 MM\_Audit\_Imp::MM\_GetNumAllocNodes () [inline]

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

### 7.32.2.18 void MM\_Audit\_Imp::MM\_GetOverheadStats (MM\_AuditOverheadStats & stats) [inline]

API to get the overhead statistics for the memory used by the mm\_audit class.

### 7.32.2.19 uint32 MM\_Audit\_Imp::MM\_GetPostfillPattern (void) [inline]

API to get the postfill pattern. The pattern is used to fill the memory before freeing it.

### 7.32.2.20 uint8 MM\_Audit\_Imp::MM\_GetPrefillPattern (void) [inline]

API to get the prefill pattern. The pattern is used to fill the memory before returning it to the caller.

### 7.32.2.21 OSCL\_IMPORT\_REF const OsciMemStatsNode\* MM\_Audit\_Imp::MM\_GetRootNode () [inline]

### 7.32.2.22 OSCL\_IMPORT\_REF MM\_Stats\_t\* MM\_Audit\_Imp::MM\_GetStats (const char \*const tagIn)

API to get memory statistics through context string(tag)

#### Returns:

statistics pointer if operation succeeds

### 7.32.2.23 OSCL\_IMPORT\_REF uint32 MM\_Audit\_Imp::MM\_GetStatsInDepth (const char \*tagIn, MM\_Stats\_CB \*array\_ptr, uint32 max\_nodes)

API to get memory statistics in detail through context string(tag) including its subtree

#### Returns:

statistics pointer array and actual number of nodes if operation succeeds

### 7.32.2.24 OSCL\_IMPORT\_REF const OsciMemStatsNode\* MM\_Audit\_Imp::MM\_GetTagNode (const char \*tagIn)

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

#### Parameters:

*tagIn* input tag

#### Returns:

pointer to OsciMemStatsNode which should be passed to MM\_allocate

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

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

#### Parameters:

*tagIn* input tag

#### Returns:

the number of tree nodes ; 0 means no tag node

**7.32.2.26 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_ReleaseAllocNodeInfo (MM\_AllocQueryInfo \* info)**

**7.32.2.27 OSCL\_IMPORT\_REF bool MM\_Audit\_Imp::MM\_SetFailurePoint (const char \* tagIn, uint32 alloc\_number)**

API to insert allocation failure deterministically according to allocation number associated with tag

**Parameters:**

*tagIn* input tag

*alloc\_number* allocation number associated with tag

**Returns:**

true if operation succeeds;

**7.32.2.28 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_SetMode (uint32 inMode)**

API to set the operating mode of the mm\_audit class.

**7.32.2.29 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_SetPostfillPattern (uint8 pattern)**

API to set the postfill pattern.

**7.32.2.30 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_SetPrefillPattern (uint8 pattern)**

API to set the prefill pattern.

**7.32.2.31 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_SetTagLevel (uint32 level)**

API to set the maximum tag level,i.e. tag level for a.b.c.d = 4

**Parameters:**

*level* input tag level to be set

**7.32.2.32 OSCL\_IMPORT\_REF void MM\_Audit\_Imp::MM\_UnsetFailurePoint (const char \* tagIn)**

API to cancel the allocation failure point associated with tag

**Parameters:**

*tagIn* input tag

**7.32.2.33 OSCL\_IMPORT\_REF bool MM\_Audit\_Imp::MM\_Validate (const void \* ptrIn)**

API to check the input pointer is a valid pointer to a chunk of memory

**Parameters:**

*ptrIn* input pointer to be validated

**Returns:**

true if operation succeeds;

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

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

**Returns:**

true if operation succeeds;

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

**7.32.2.37** `OsciMemStatsNode* 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 (OsciMemStatsNode * pCurrStatsNode, const MM\_Stats\_t & pDelta, bool bAdd)`

**Returns:**

true if operation succeeds;

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

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

**Returns:**

true if operation succeeds;

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

**Returns:**

true if operation succeeds;

**7.32.2.44** `bool MM_Audit_Imp::validate_all_heap ()`**Returns:**

true if operation succeeds;

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

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

## 7.33 MM\_AuditOverheadStats Struct Reference

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

### Data Fields

- uint32 [per\\_allocation\\_overhead](#)
- uint32 [stats\\_overhead](#)

### 7.33.1 Field Documentation

**7.33.1.1** uint32 MM\_AuditOverheadStats::per\_allocation\_overhead

**7.33.1.2** uint32 MM\_AuditOverheadStats::stats\_overhead

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

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

## 7.34 MM\_FailInsertParam Struct Reference

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

### Public Methods

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

### Data Fields

- uint32 [nAllocNum](#)
- uint16 [xsubi](#) [3]

### 7.34.1 Constructor & Destructor Documentation

7.34.1.1 [MM\\_FailInsertParam::MM\\_FailInsertParam \(\)](#) [inline]

### 7.34.2 Member Function Documentation

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

7.34.2.2 [void\\* MM\\_FailInsertParam::operator new \(oscl\\_memsize\\_t size, MM\\_FailInsertParam \\*ptr\)](#) [inline]

7.34.2.3 [void\\* MM\\_FailInsertParam::operator new \(oscl\\_memsize\\_t size\)](#) [inline]

7.34.2.4 [void MM\\_FailInsertParam::reset \(\)](#) [inline]

### 7.34.3 Field Documentation

7.34.3.1 [uint32 MM\\_FailInsertParam::nAllocNum](#)

7.34.3.2 [uint16 MM\\_FailInsertParam::xsubi\[3\]](#)

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

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



## 7.35 MM\_Stats\_CB Struct Reference

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

### Public Methods

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

### Data Fields

- const char \* [tag](#)
- const [MM\\_Stats\\_t](#) \* [pStats](#)
- uint32 [num\\_child\\_nodes](#)

### 7.35.1 Constructor & Destructor Documentation

7.35.1.1 [MM\\_Stats\\_CB::MM\\_Stats\\_CB](#) () [inline]

### 7.35.2 Member Function Documentation

7.35.2.1 void [MM\\_Stats\\_CB::operator delete](#) (void \* *ptr*) throw () [inline]

7.35.2.2 void\* [MM\\_Stats\\_CB::operator new](#) ([oscl\\_memsize\\_t](#) *size*, MM\_Stats\_CB \* *ptr*) [inline]

7.35.2.3 void\* [MM\\_Stats\\_CB::operator new](#) ([oscl\\_memsize\\_t](#) *size*) [inline]

### 7.35.3 Field Documentation

7.35.3.1 uint32 [MM\\_Stats\\_CB::num\\_child\\_nodes](#)

7.35.3.2 const [MM\\_Stats\\_t](#)\* [MM\\_Stats\\_CB::pStats](#)

7.35.3.3 const char\* [MM\\_Stats\\_CB::tag](#)

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

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

## 7.36 MM\_Stats\_t Struct Reference

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

### Public Methods

- [MM\\_Stats\\_t](#) ()
- [MM\\_Stats\\_t](#) (uint32 sizeIn)
- void [reset](#) ()
- void [update](#) (const MM\_Stats\_t &delta, bool add)
- void \* [operator new](#) ([oscl\\_memsize\\_t](#) size)
- void \* [operator new](#) ([oscl\\_memsize\\_t](#) size, MM\_Stats\_t \*ptr)
- void [operator delete](#) (void \*ptr) throw ()

### Data Fields

- uint32 [numBytes](#)
- uint32 [peakNumBytes](#)
- uint32 [numAllocs](#)
- uint32 [peakNumAllocs](#)
- uint32 [numAllocFails](#)
- uint32 [totalNumAllocs](#)
- uint32 [totalNumBytes](#)

### 7.36.1 Constructor & Destructor Documentation

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

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

### 7.36.2 Member Function Documentation

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

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

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

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

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

### 7.36.3 Field Documentation

7.36.3.1 `uint32 MM_Stats_t::numAllocFails`

7.36.3.2 `uint32 MM_Stats_t::numAllocs`

7.36.3.3 `uint32 MM_Stats_t::numBytes`

7.36.3.4 `uint32 MM_Stats_t::peakNumAllocs`

7.36.3.5 `uint32 MM_Stats_t::peakNumBytes`

7.36.3.6 `uint32 MM_Stats_t::totalNumAllocs`

7.36.3.7 `uint32 MM_Stats_t::totalNumBytes`

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

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

## 7.37 NTPTIME Class Reference

The NTPTIME class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900.

```
#include <oscl_time.h>
```

### Public Methods

- OSCL\_COND\_IMPORT\_REF NTPTIME ()  
*The default constructor creates an NTPTIME instance representing the current system time.*
- OSCL\_COND\_IMPORT\_REF NTPTIME (const NTPTIME &src)  
*Copy constructor to create a new NTPTIME from an existing one.*
- OSCL\_COND\_IMPORT\_REF NTPTIME (const uint32 seconds)  
*Construct an NTPTIME from a uint32.*
- OSCL\_COND\_IMPORT\_REF NTPTIME (const int32 seconds)  
*Construct an NTPTIME from a int.*
- OSCL\_COND\_IMPORT\_REF NTPTIME (const TimeValue &t)  
*Construct a NTPTIME instance from a TimeValue instance.*
- OSCL\_COND\_IMPORT\_REF NTPTIME (const uint64 value)  
*Construct a NTPTIME instance from a uint64 value.*
- OSCL\_COND\_IMPORT\_REF NTPTIME & operator= (uint32 newval)  
*The assignment operator for a 32 bit integer.*
- OSCL\_COND\_IMPORT\_REF NTPTIME & operator= (uint64 newval)  
*The assignment operator for a 64 bit integer.*
- OSCL\_COND\_IMPORT\_REF NTPTIME & operator+= (uint64 val)  
*The += operator is used to add a 64 bit 32.32 value to an existing NTPTIME value.*
- OSCL\_COND\_IMPORT\_REF NTPTIME operator- (const NTPTIME &ntpt) const  
*The - operator allows subtraction of one NTPTIME value from another. This is useful to measure an interval.*
- void set\_from\_system\_time (const uint32 systemtime)  
*This method converts a 32-bit system time to NTP time.*
- OSCL\_COND\_IMPORT\_REF uint32 get\_middle32 () const  
*Grab the middle 32 bits of the 64 bit 32.32 representation.*
- OSCL\_COND\_IMPORT\_REF uint32 get\_upper32 () const  
*This method returns the upper 32 bits of the 32.32 representation.*
- OSCL\_COND\_IMPORT\_REF uint32 get\_lower32 () const  
*This method returns the lower 32 bits of the 32.32 representation.*

- `int32 to_system_time () const`  
*This method converts the ntp time value to system time.*
- `OSCL_COND_IMPORT_REF uint64 get_value () const`  
*This method returns the 32.32 ntp representation.*
- `OSCL_IMPORT_REF int set_to_current_time ()`  
*This method sets the 32.32 representation to the current system time value.*

### 7.37.1 Detailed Description

The NTPTIME class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900.

The NTPTIME class: Conversion to/from Unix (epoch at 0h Jan. 1, 1970) amount to addition/subtraction of 2208988800. A single 64 bit value is used to represent the time. This value represents the number of seconds since 0h (UTC) Jan. 1, 1900. There is an implied binary point between the upper 32 bits and lower 32 bits (this is referred to as a 32.32 fractional representation). For example a binary value of 00000000 00000000 00000011 10000000 00000000 00000000 00000000 represents 3.5 seconds since Jan 1, 1900.

### 7.37.2 Constructor & Destructor Documentation

#### 7.37.2.1 OSCL\_COND\_IMPORT\_REF NTPTIME::NTPTIME ()

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

#### 7.37.2.2 OSCL\_COND\_IMPORT\_REF NTPTIME::NTPTIME (const NTPTIME & src)

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

#### 7.37.2.3 OSCL\_COND\_IMPORT\_REF NTPTIME::NTPTIME (const uint32 seconds)

Construct an NTPTIME from a uint32.

##### Parameters:

*seconds* The uint32 input represents the number of seconds since Jan. 1, 1900.

#### 7.37.2.4 OSCL\_COND\_IMPORT\_REF NTPTIME::NTPTIME (const int32 seconds)

Construct an NTPTIME from a int.

##### Parameters:

*seconds* The int input represents the number of seconds since Jan. 1, 1900.

### 7.37.2.5 OSCL\_COND\_IMPORT\_REF NTPTIME::NTPTIME (const TimeValue & t)

Construct a NTPTIME instance from a TimeValue instance.

This constructor creates an NTPTIME value representing the same absolute time as the TimeValue parameter.

#### Parameters:

*t* A reference to a TimeValue object.

### 7.37.2.6 OSCL\_COND\_IMPORT\_REF NTPTIME::NTPTIME (const uint64 value)

Construct a NTPTIME instance from a uint64 value.

#### Parameters:

*value* A 64 bit integer argument which is used as the ntp 32.32 fractional representation.

## 7.37.3 Member Function Documentation

### 7.37.3.1 OSCL\_COND\_IMPORT\_REF uint32 NTPTIME::get\_lower32 ()

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

### 7.37.3.2 OSCL\_COND\_IMPORT\_REF uint32 NTPTIME::get\_middle32 ()

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

### 7.37.3.3 OSCL\_COND\_IMPORT\_REF uint32 NTPTIME::get\_upper32 ()

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

### 7.37.3.4 OSCL\_COND\_IMPORT\_REF uint64 NTPTIME::get\_value ()

This method returns the 32.32 ntp representation.

### 7.37.3.5 OSCL\_COND\_IMPORT\_REF NTPTIME& NTPTIME::operator+= (uint64 val)

The += operator is used to add a 64 bit 32.32 value to an existing NTPTIME value.

#### Parameters:

*val* The 64 bit 32.32 value to add to this object's value.

### 7.37.3.6 OSCL\_COND\_IMPORT\_REF NTPTIME NTPTIME::operator- (const NTPTIME & ntp)

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

#### Parameters:

*ntp* 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.

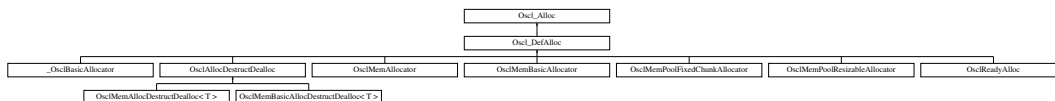
The documentation for this class was generated from the following file:

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

## 7.38 Osl\_Alloc Class Reference

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

Inheritance diagram for Osl\_Alloc::



### Public Methods

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

### 7.38.1 Constructor & Destructor Documentation

**7.38.1.1** virtual `Osl_Alloc::~~Osl_Alloc()` [inline, virtual]

### 7.38.2 Member Function Documentation

**7.38.2.1** virtual `OslAny* Osl_Alloc::allocate(const uint32 size)` [pure virtual]

Implemented in `_OslBasicAllocator`, `Osl_DefAlloc`, `OslMemAllocator`, `OslMemBasicAllocator`, `OslMemAllocDestructDealloc< T >`, `OslMemBasicAllocDestructDealloc< T >`, `OslMemPoolFixedChunkAllocator`, `OslMemPoolResizableAllocator`, and `OslReadyAlloc`.

**7.38.2.2** virtual `OslAny* Osl_Alloc::allocate_fl(const uint32 size, const char *file_name, const int line_num)` [inline, virtual]

Reimplemented in `Osl_DefAlloc`, `OslMemAllocator`, `OslMemAllocDestructDealloc< T >`, and `OslReadyAlloc`.

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

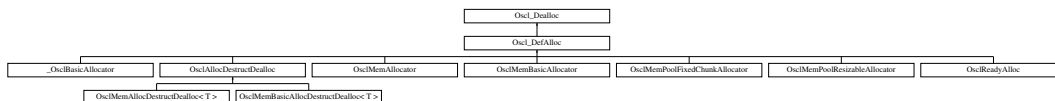
- `osl_defalloc.h`



## 7.39 Osci\_Dealloc Class Reference

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

Inheritance diagram for Osci\_Dealloc::



### Public Methods

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

### 7.39.1 Constructor & Destructor Documentation

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

### 7.39.2 Member Function Documentation

**7.39.2.1** virtual void [Osci\\_Dealloc::deallocate](#) ([OsciAny](#) \*p) [pure virtual]

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

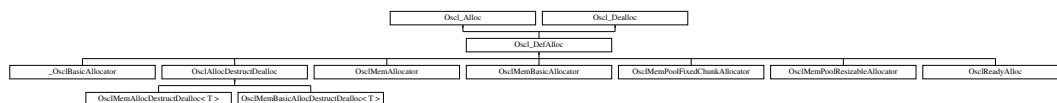
The documentation for this class was generated from the following file:

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

## 7.40 Osci\_DefAlloc Class Reference

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

Inheritance diagram for Osci\_DefAlloc::



### Public Methods

- virtual [OsciAny](#) \* [allocate](#) (const uint32 size)=0
- virtual [OsciAny](#) \* [allocate\\_fl](#) (const uint32 size, const char \*file\_name, const int line\_num)
- virtual void [deallocate](#) ([OsciAny](#) \*p)=0

### 7.40.1 Member Function Documentation

#### 7.40.1.1 virtual [OsciAny](#)\* Osci\_DefAlloc::allocate (const uint32 size) [pure virtual]

Implements [Osci\\_Alloc](#).

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

#### 7.40.1.2 virtual [OsciAny](#)\* Osci\_DefAlloc::allocate\_fl (const uint32 size, const char \*file\_name, const int line\_num) [inline, virtual]

Reimplemented from [Osci\\_Alloc](#).

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

#### 7.40.1.3 virtual void Osci\_DefAlloc::deallocate ([OsciAny](#) \*p) [pure virtual]

Implements [Osci\\_Dealloc](#).

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

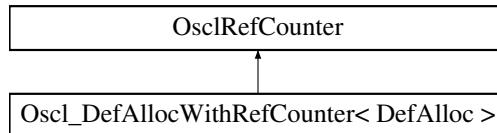
The documentation for this class was generated from the following file:

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

## 7.41 Osci\_DefAllocWithRefCount< DefAlloc > Class Template Reference

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

Inheritance diagram for Osci\_DefAllocWithRefCount< DefAlloc >::



### Public Methods

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

### Static Public Methods

- Osci\_DefAllocWithRefCount \* [New](#) ()

### 7.41.1 Detailed Description

**template<class DefAlloc> class Osci\_DefAllocWithRefCount< DefAlloc >**

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

### 7.41.2 Member Function Documentation

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

Add to the reference count

Implements [OsciRefCount](#).

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

Delete object

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

Gets the current number of references

Implements [OsciRefCount](#).

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

Create object

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

Delete from reference count

Implements [OsciRefCount](#).

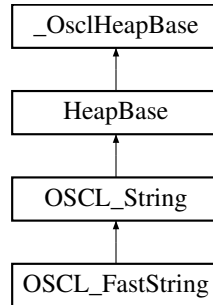
The documentation for this class was generated from the following file:

- [osci\\_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](#) \* *cstring*)

Create the string and initialize it to contain the input string. The string is not writable.

**am:** null-terminated string.

#### 7.42.3.4 OSCL\_IMPORT\_REF OSCL\_FastString::OSCL\_FastString ([chartype](#) \* *buf*, uint32 *maxlen*)

Create the string and initialize it to contain the input string. The string is writable.

#### Parameters:

*cp*: null-terminated string.

*maxlen*: maximum size of storage at *cp*, not incl null terminator. If input string is not null-terminated, the function leaves.

### 7.42.3.5 OSCL\_IMPORT\_REF OSCL\_FastString::~~OSCL\_FastString ()

## 7.42.4 Member Function Documentation

### 7.42.4.1 OSCL\_IMPORT\_REF const [char](#)\* OSCL\_FastString::get\_cstr () [virtual]

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

Implements [OSCL\\_String](#).

### 7.42.4.2 OSCL\_IMPORT\_REF uint32 OSCL\_FastString::get\_maxsize () [virtual]

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

Implements [OSCL\\_String](#).

### 7.42.4.3 OSCL\_IMPORT\_REF uint32 OSCL\_FastString::get\_size () [virtual]

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

Implements [OSCL\\_String](#).

### 7.42.4.4 OSCL\_IMPORT\_REF [char](#)\* OSCL\_FastString::get\_str () [virtual]

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

Implements [OSCL\\_String](#).

### 7.42.4.5 OSCL\_IMPORT\_REF OSCL\_FastString& OSCL\_FastString::operator= (const [char](#)\* cstr)

Assignment operator

**am:** null-terminated string

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

### 7.42.4.6 OSCL\_IMPORT\_REF OSCL\_FastString& OSCL\_FastString::operator= (const OSCL\_FastString & src)

Assignment operators

### 7.42.4.7 OSCL\_IMPORT\_REF void OSCL\_FastString::set (const [other\\_char](#)\* buf, uint32 numofbyte, [optype](#) op)

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

#### Parameters:

**buf:** string or character array.

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

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

#### 7.42.4.8 OSCL\_IMPORT\_REF void OSCL\_FastString::set ([chartype](#) \* *cstr*, uint32 *maxlen*)

This function can be used to reassign the string to a new writable string. If input string is not null-terminated, the function leaves.

#### 7.42.4.9 OSCL\_IMPORT\_REF void OSCL\_FastString::set\_length ()

This function can be used to refresh the string size in case the contents of the string buffer have been modified since the container was created.

### 7.42.5 Friends And Related Function Documentation

#### 7.42.5.1 friend class OSCL\_String [friend]

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

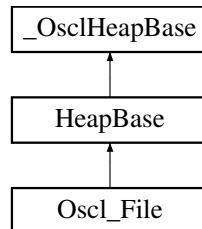
- [oscl\\_string\\_containers.h](#)



## 7.43 Osci\_File Class Reference

```
#include <osci_file_io.h>
```

Inheritance diagram for Osci\_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 `Osci_File` ()
- OSCL\_IMPORT\_REF `Osci_File` (uint32 aCacheSize)
- OSCL\_IMPORT\_REF `Osci_File` (uint32 aCacheSize, `OsciFileHandle` \*aFileHandle)
- OSCL\_IMPORT\_REF `~Osci_File` ()
- OSCL\_IMPORT\_REF void `SetPVCacheSize` (uint32 aSize)
- void `AddFixedCache` (const `OsciFixedCacheParam` &aParam)
- void `RemoveFixedCache` (const `TOsciFileOffset` &aPos)
- void `SetCacheObserver` (`OsciCacheObserver` \*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` (`OsciFileHandle` \*aHandle)
- OSCL\_IMPORT\_REF int32 `Open` (const char \*filename, uint32 mode, `Osci_FileServer` &fileserv)
- OSCL\_IMPORT\_REF int32 `Open` (const `osci_wchar` \*filename, uint32 mode, `Osci_FileServer` &fileserv)
- OSCL\_IMPORT\_REF uint32 `Read` (`OsciAny` \*buffer, uint32 size, uint32 numelements)
- OSCL\_IMPORT\_REF uint32 `Write` (const `OsciAny` \*buffer, uint32 size, uint32 numelements)
- OSCL\_IMPORT\_REF int32 `Seek` (`TOsciFileOffset` offset, `seek_type` origin)
- OSCL\_IMPORT\_REF `TOsciFileOffset` `Tell` ()
- OSCL\_IMPORT\_REF int32 `Close` ()
- OSCL\_IMPORT\_REF int32 `Flush` ()
- OSCL\_IMPORT\_REF int32 `SetSize` (uint32 size)
- OSCL\_IMPORT\_REF int32 `EndOfFile` ()
- OSCL\_IMPORT\_REF int32 `GetError` ()

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

## Friends

- class [OsciFileCache](#)
- class [OsciFileCacheBuffer](#)
- class [asyncfilereadwrite\\_test](#)
- class [largeasyncfilereadwrite\\_test](#)
- class [asyncfilereadcancel\\_test](#)

## 7.43.1 Member Enumeration Documentation

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

#### Enumeration values:

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

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

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

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

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

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

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

#### Enumeration values:

**SEEKSET** Beginning of file

**SEEKCUR** Current position of file pointer

**SEEKEND** End of file

### 7.43.1.3 enum Osci\_File::TSymbianAccessMode

Defines mode options for SetNativeAccessMode on Symbian.

#### Enumeration values:

**ESymbianAccessMode\_Rfile**

**ESymbianAccessMode\_RfileBuf**

## 7.43.2 Constructor & Destructor Documentation

### 7.43.2.1 OSCL\_IMPORT\_REF Osl\_File::Osl\_File ()

Constructor

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

Deprecated Constructor, present for back-compatibility.

#### Parameters:

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

### 7.43.2.3 OSCL\_IMPORT\_REF Osl\_File::Osl\_File (uint32 *aCacheSize*, [OslFileHandle](#) \* *aFileHandle*)

Deprecated Constructor, present for back-compatibility.

#### Parameters:

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

*aFileHandle*: open file handle.

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

Destructor

## 7.43.3 Member Function Documentation

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

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

#### Parameters:

*aParam*: Cache location and size.

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

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

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

#### Returns:

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

#### 7.43.3.3 OSCL\_IMPORT\_REF int32 Osl\_File::EndOfFile ()

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

**Returns:**

#### 7.43.3.4 OSCL\_IMPORT\_REF int32 Osl\_File::Flush ()

The File Flush operation On an output stream Osl\_FileFlush causes any buffered but unwritten data to be written to the file. 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 Osl\_File::GetError ()

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

**Returns:**

#### 7.43.3.6 [OslFileHandle\\*](#) Osl\_File::Handle () [inline]

Retrieve the file handle.

**Returns:**

file handle

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

Opens a file.

Note: when an external file handle is used, Open will attach to the file handle and initialize 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 Osl\_File::Open (const char \* *filename*, uint32 *mode*, Osl\_FileServer & *fileserv*)

Opens a file.

Note: when an external file handle is used, Open will attach to the file handle and initialize 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 Osl\_File::Read (OslAny \* *buffer*, uint32 *size*, uint32 *numelements*)

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

#### Parameters:

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

#### Returns:

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

### 7.43.3.10 void Osl\_File::RemoveFixedCache (const TOslFileOffset & *aPos*) [inline]

RemoveFixedCache removes a fixed cache.

#### Parameters:

*aPos*: Cache location and size.

### 7.43.3.11 OSCL\_IMPORT\_REF int32 Osl\_File::Seek (TOslFileOffset *offset*, seek\_type *origin*)

The File Seek operation Sets the position for file pointer

#### Parameters:

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

#### Returns:

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

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

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

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

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

#### Parameters:

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

### 7.43.3.13 void Osl\_File::SetCacheObserver (OslCacheObserver \* *aObs*) [inline]

### 7.43.3.14 OSCL\_IMPORT\_REF int32 Osl\_File::SetFileHandle (OslFileHandle \* *aHandle*)

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

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

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

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

#### Parameters:

*aHandle*: container for an open file handle.

#### Returns:

returns 0 if successful, non-zero if error.

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

SetLoggingEnable configures the PVLogger output for this file. This will enable full logging of each API entry and exit using the logger object "Osl\_File", plus full logging of native operation entry & exit using logger object "OslNativeFile".

#### Parameters:

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

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

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

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

**Parameters:**

*aMode:* access mode.

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

SetNativeBufferSize configures the native file buffering feature, when available.

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

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

**Parameters:**

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

**7.43.3.18 OSCL\_IMPORT\_REF void Osci\_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 Osci\_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 Osci\_File::SetSummaryStatsLoggingEnable (bool aEnable)**

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

**Parameters:**

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

**7.43.3.21 OSCL\_IMPORT\_REF [TOsciFileOffset](#) Osci\_File::Size ()**

Get the file size in bytes.

**Returns:**

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

**7.43.3.22 OSCL\_IMPORT\_REF [TOsciFileOffset](#) Osci\_File::Tell ()**

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

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

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

**Parameters:**

- buffer* pointer to buffer of type void
- size* element size in bytes
- numelements* number of elements to write

**Returns:**

The number of elements written

**7.43.4 Friends And Related Function Documentation**

**7.43.4.1 friend class asyncfilereadcancel\_test** [friend]

**7.43.4.2 friend class asyncfilereadwrite\_test** [friend]

**7.43.4.3 friend class largeasyncfilereadwrite\_test** [friend]

**7.43.4.4 friend class OsciFileCache** [friend]

**7.43.4.5 friend class OsciFileCacheBuffer** [friend]

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

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



## 7.44 Osl\_File::OslCacheObserver Class Reference

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

### Public Methods

- virtual [~OslCacheObserver](#) ()
- virtual [OslFileCacheBuffer](#) \* [ChooseCurCache](#) ([OslFileCache](#) &aContext, [TOslFileOffset](#) 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 [Osl\\_File::OslCacheObserver::~OslCacheObserver](#) () [inline, virtual]

### 7.44.3 Member Function Documentation

**7.44.3.1** virtual [OslFileCacheBuffer](#)\* [Osl\\_File::OslCacheObserver::ChooseCurCache](#) ([OslFileCache](#) &aContext, [TOslFileOffset](#) aPos) [pure virtual]

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

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

## 7.45 Osl\_File::OslFixedCacheParam Class Reference

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

### Public Methods

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

### Data Fields

- [TOslFileOffset](#) iFilePosition
- uint32 [iSize](#)

### 7.45.1 Detailed Description

Parameters for defining a fixed cache

### 7.45.2 Member Function Documentation

**7.45.2.1** bool Osl\_File::OslFixedCacheParam::Contains ([TOslFileOffset](#) *pos*) const  
[inline]

### 7.45.3 Field Documentation

**7.45.3.1** [TOslFileOffset](#) Osl\_File::OslFixedCacheParam::iFilePosition

**7.45.3.2** uint32 Osl\_File::OslFixedCacheParam::iSize

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

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

## 7.46 Osl\_FileFind Class Reference

```
#include <osl_file_find.h>
```

### Public Types

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

### Public Methods

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

### 7.46.1 Detailed Description

Osl\_FileFind class defines the generic way of finding filesystem elements that match a pattern within a directory

### 7.46.2 Member Enumeration Documentation

#### 7.46.2.1 enum Osl\_FileFind::element\_type

Enumeration values:

`FILE_TYPE`

`DIR_TYPE`

`INVALID_TYPE`

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

Enumeration values:

`E_OK`

`E_INVALID_STATE`

`E_INVALID_ARG`

`E_PATH_TOO_LONG`

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

### 7.46.3 Constructor & Destructor Documentation

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

constructor.

**Returns:**  
none

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

destructor. will deallocate open handles if necessary

**Returns:**  
none

### 7.46.4 Member Function Documentation

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

closes the handle to directory.

**Returns:**  
none

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

Opens a directory for reading.

**Parameters:**

- directory* directory to search (utf16).
- pattern* wildcard pattern filter (utf16). passing NULL, results in a universal match.
- buf* buffer for returned pathname (utf16).
- buflen* size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E\_BUFFER\_TOO\_SMALL.

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

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

Finds first element matching the pattern.

**Parameters:**

*directory* directory to search (utf8).

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

*buf* buffer for returned pathname (utf8).

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

**Returns:**

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

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

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

**Parameters:**

*buf* buffer to hold directory name(utf16)

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

**Returns:**

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

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

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

**Parameters:**

*buf* buffer to hold directory name(utf8)

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

**Returns:**

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

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

Returns the element type for the last element returned

**Returns:**

see enumeration above for more info.

#### 7.46.4.7 OSCL\_IMPORT\_REF [error\\_type](#) Osci\_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:

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

## 7.47 Osl\_FileServer Class Reference

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

### Public Methods

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

### Friends

- class [Osl\\_File](#)
- class [OslNativeFile](#)

### 7.47.1 Constructor & Destructor Documentation

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

Constructor

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

Destructor

### 7.47.2 Member Function Documentation

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

Closes a file server.

**Returns:**

returns 0 on success and a non-zero value otherwise

#### 7.47.2.2 OSCL\_IMPORT\_REF int32 Osl\_FileServer::Connect (bool *aShareSession* = false)

Connects the server. This must be called before a file server can be used.

**Returns:**

returns 0 on success and a non-zero value otherwise

### 7.47.2.3 OSCL\_IMPORT\_REF int32 Osl\_FileServer::Osl\_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 Osl\_FileServer::Osl\_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 Osl\_File [friend]

### 7.47.3.2 friend class OslNativeFile [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

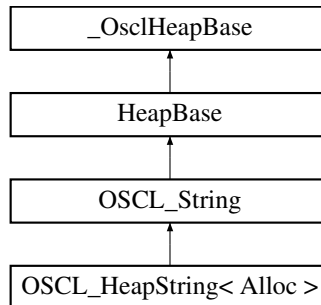
The documentation for this struct was generated from the following file:

- [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](#) optype
- typedef [OSCL\\_wString::chartype](#) other\_chartype

### Public Methods

- [OSCL\\_HeapString](#) ()
- [OSCL\\_HeapString](#) (const OSCL\_HeapString &src)
- [OSCL\\_HeapString](#) (const [OSCL\\_String](#) &src)
- [OSCL\\_HeapString](#) (const [chartype](#) \*cstr)
- [OSCL\\_HeapString](#) (const [chartype](#) \*buf, uint32 length)
- [~OSCL\\_HeapString](#) ()
- uint32 [get\\_size](#) () const
- uint32 [get\\_maxsize](#) () const
- const [chartype](#) \* [get\\_cstr](#) () const
- [chartype](#) \* [get\\_str](#) () const
- OSCL\_HeapString & [operator=](#) (const OSCL\_HeapString &src)
- OSCL\_HeapString & [operator=](#) (const [OSCL\\_String](#) &src)
- OSCL\_HeapString & [operator=](#) (const [chartype](#) \*cstr)
- void [set](#) (const [chartype](#) \*buf, uint32 length)
- void [set](#) (const [other\\_chartype](#) \*buf, [optype](#) op)
- void [set](#) (const [other\\_chartype](#) \*buf, uint32 length, [optype](#) op)

### Friends

- class [OSCL\\_String](#)

### 7.49.1 Detailed Description

```
template<class Alloc> class OSCL_HeapString< Alloc >
```

OSCL\_HeapString is a simple string class, compatible with regular character array strings.

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

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

#### Parameters:

*Alloc*: memory allocator, derived from [OscL\\_DefAlloc](#).

### 7.49.2 Member Typedef Documentation

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

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

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

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

### 7.49.3 Friends And Related Function Documentation

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

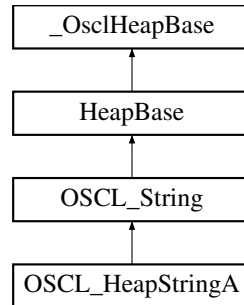
The documentation for this class was generated from the following file:

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

## 7.50 OSCL\_HeapStringA Class Reference

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

Inheritance diagram for OSCL\_HeapStringA::



### Public Types

- typedef OSCL\_String::chartype [chartype](#)
- typedef [TOSCL\\_StringOp](#) optype
- typedef [OSCL\\_wString::chartype](#) other\_chartype

### Public Methods

- OSCL\_IMPORT\_REF [OSCL\\_HeapStringA](#) ()
- OSCL\_IMPORT\_REF [OSCL\\_HeapStringA](#) ([Osc\\_DefAlloc](#) \*alloc, [OscRefCount](#) \*ref=NULL)
- OSCL\_IMPORT\_REF [OSCL\\_HeapStringA](#) (const [OSCL\\_HeapStringA](#) &src)
- OSCL\_IMPORT\_REF [OSCL\\_HeapStringA](#) (const [OSCL\\_HeapStringA](#) &src, [Osc\\_DefAlloc](#) \*alloc, [OscRefCount](#) \*ref=NULL)
- OSCL\_IMPORT\_REF [OSCL\\_HeapStringA](#) (const [OSCL\\_String](#) &src, [Osc\\_DefAlloc](#) \*alloc=NULL, [OscRefCount](#) \*ref=NULL)
- OSCL\_IMPORT\_REF [OSCL\\_HeapStringA](#) (const [chartype](#) \*cstr, [Osc\\_DefAlloc](#) \*alloc=NULL, [OscRefCount](#) \*ref=NULL)
- OSCL\_IMPORT\_REF [OSCL\\_HeapStringA](#) (const [chartype](#) \*buf, uint32 length, [Osc\\_DefAlloc](#) \*alloc=NULL, [OscRefCount](#) \*ref=NULL)
- OSCL\_IMPORT\_REF [~OSCL\\_HeapStringA](#) ()
- OSCL\_IMPORT\_REF uint32 [get\\_size](#) () const
- OSCL\_IMPORT\_REF uint32 [get\\_maxsize](#) () const
- OSCL\_IMPORT\_REF const [chartype](#) \* [get\\_cstr](#) () const
- OSCL\_IMPORT\_REF [chartype](#) \* [get\\_str](#) () const
- OSCL\_IMPORT\_REF [OSCL\\_HeapStringA](#) & [operator=](#) (const [OSCL\\_HeapStringA](#) &src)
- OSCL\_IMPORT\_REF [OSCL\\_HeapStringA](#) & [operator=](#) (const [OSCL\\_String](#) &src)
- OSCL\_IMPORT\_REF [OSCL\\_HeapStringA](#) & [operator=](#) (const [chartype](#) \*cstr)
- OSCL\_IMPORT\_REF void [set](#) (const [chartype](#) \*buf, uint32 length)
- OSCL\_IMPORT\_REF void [set](#) (const [other\\_chartype](#) \*buf, [optype](#) op)
- OSCL\_IMPORT\_REF void [set](#) (const [other\\_chartype](#) \*buf, uint32 length, [optype](#) op)

### Friends

- class [OSCL\\_String](#)

## 7.50.1 Detailed Description

OSCL\_HeapStringA is a simple string class, compatible with regular character array strings. It is similar to [OSCL\\_HeapString](#), except that the allocator is passed at run-time instead of compile-time. The allocator pointer is passed in the constructor, and may be a reference-counted object. If the allocator is not a reference-counted object then it must persist over the lifetime of all OSCL\_HeapStringA objects that use it. If no allocator is provided, then an [OscMemAllocator](#) will be used.

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

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

## 7.50.2 Member Typedef Documentation

### 7.50.2.1 typedef OSCL\_String::chartype OSCL\_HeapStringA::chartype

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

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

### 7.50.3.2 OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA ([Osc\\_DefAlloc](#) \* *alloc*, [OscRefCount](#) \* *ref* = NULL)

### 7.50.3.3 OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const OSCL\_HeapStringA & *src*)

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

**Parameters:**

**src:** input string.

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

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

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

**7.50.3.4** OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const OSCL\_HeapStringA & *src*, [OscDefAlloc](#) \* *alloc*, [OscRefCount](#) \* *ref* = NULL)

**7.50.3.5** OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const [OSCL\\_String](#) & *src*, [OscDefAlloc](#) \* *alloc* = NULL, [OscRefCount](#) \* *ref* = NULL)

**7.50.3.6** OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const [chartype](#) \* *cstr*, [OscDefAlloc](#) \* *alloc* = NULL, [OscRefCount](#) \* *ref* = NULL)

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

**Parameters:**

*cp*: null-terminated string.

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

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

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

**7.50.3.7** OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const [chartype](#) \* *buf*, [uint32](#) *length*, [OscDefAlloc](#) \* *alloc* = NULL, [OscRefCount](#) \* *ref* = NULL)

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

**Parameters:**

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

*length*: number of characters to copy.

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

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

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

**7.50.3.8** OSCL\_IMPORT\_REF OSCL\_HeapStringA::~OSCL\_HeapStringA ()

## 7.50.4 Member Function Documentation

**7.50.4.1** OSCL\_IMPORT\_REF const [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 \* *ctr*)

Assignment operator

**am:** null-terminated string

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

#### 7.50.4.6 OSCL\_IMPORT\_REF OSCL\_HeapStringA& OSCL\_HeapStringA::operator= (const OSCL\_String & *src*)

Assignment operator

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

#### 7.50.4.7 OSCL\_IMPORT\_REF OSCL\_HeapStringA& OSCL\_HeapStringA::operator= (const OSCL\_HeapStringA & *src*)

Assignment operators

#### 7.50.4.8 OSCL\_IMPORT\_REF void OSCL\_HeapStringA::set (const other\_chartype \* *buf*, uint32 *length*, optype *op*)

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

##### Parameters:

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

***length*:** number of characters to copy.

***op*:** conversion operation to apply

#### 7.50.4.9 OSCL\_IMPORT\_REF void OSCL\_HeapStringA::set (const [other\\_chartype](#) \* *buf*, [optype](#) *op*)

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

##### Parameters:

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

#### 7.50.4.10 OSCL\_IMPORT\_REF void OSCL\_HeapStringA::set (const [chartype](#) \* *buf*, uint32 *length*)

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

##### Parameters:

- buf*: string or character array.
- length*: number of characters to copy.

### 7.50.5 Friends And Related Function Documentation

#### 7.50.5.1 friend class OSCL\_String [friend]

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

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



## 7.51 Osl\_Int64\_Utils Class Reference

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

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

### Static Public Methods

- OSCL\_IMPORT\_REF void [set\\_int64](#) (int64 &input\_value, const int32 upper, const uint32 lower)
- OSCL\_IMPORT\_REF int32 [get\\_int64\\_upper32](#) (const int64 &input\_value)
- OSCL\_IMPORT\_REF uint32 [get\\_int64\\_lower32](#) (const int64 &input\_value)
- OSCL\_IMPORT\_REF uint32 [get\\_int64\\_middle32](#) (const int64 &input\_value)
- OSCL\_IMPORT\_REF void [set\\_uint64](#) (uint64 &input\_value, const uint32 upper, const uint32 lower)
- OSCL\_IMPORT\_REF uint32 [get\\_uint64\\_upper32](#) (const uint64 &input\_value)
- OSCL\_IMPORT\_REF uint32 [get\\_uint64\\_lower32](#) (const uint64 &input\_value)
- OSCL\_IMPORT\_REF uint32 [get\\_uint64\\_middle32](#) (const uint64 &input\_value)

### 7.51.1 Detailed Description

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

The Osl\_Int64\_Utils class:

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

## 7.51.2 Member Function Documentation

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

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

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

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

```
#include <osl_map.h>
```

### Public Methods

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

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

### 7.52.1 Member Function Documentation

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

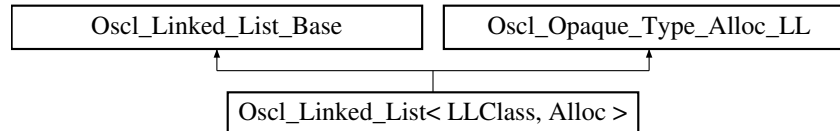
The documentation for this struct was generated from the following file:

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

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

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

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



### Public Methods

- [Osl\\_Linked\\_List \(\)](#)
- [~Osl\\_Linked\\_List \(\)](#)
- void [clear \(\)](#)
- int32 [dequeue\\_element](#) (LLClass &element)
- int32 [get\\_first](#) (LLClass &ele)
- int32 [get\\_next](#) (LLClass &ele)
- int32 [check\\_list](#) ()
- int32 [get\\_num\\_elements](#) ()
- int32 [add\\_element](#) (LLClass &new\_element)
- int32 [add\\_to\\_front](#) (const LLClass &new\_element)
- int32 [insert\\_element](#) (const LLClass &new\_element, int index)
- int32 [get\\_element](#) (int32 index, LLClass &element)
- int32 [remove\\_element](#) (const LLClass &data\_to\_remove)
- int32 [get\\_index](#) (const LLClass &data)
- int32 [remove\\_element](#) (const int32 index\_to\_remove)
- int32 [move\\_to\\_end](#) (const LLClass &data\_to\_move)
- int32 [move\\_to\\_front](#) (const LLClass &data\_to\_move)

### 7.53.1 Detailed Description

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

Osl Linked List Class

### 7.53.2 Constructor & Destructor Documentation

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

Initialized the protected variables of list.

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

The destructor.

### 7.53.3 Member Function Documentation

**7.53.3.1** `template<class LLClass, class Alloc> int32 Osl_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 Osl_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 Osl_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 [Osl\\_Linked\\_List\\_Base](#).

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

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

**7.53.3.6** `template<class LLClass, class Alloc> int32 Osl_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 Osl_Linked_List< LLClass, Alloc >::get_first (LLClass & ele) [inline]`

Return the first element of list in passed parameter,

### Parameters:

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

### Returns:

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

**7.53.3.8** `template<class LLClass, class Alloc> int32 Osl_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 Osl_Linked_List< LLClass, Alloc >::get_next (LLClass & ele) [inline]`

Return the next element of list in passed parameter,

### Parameters:

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

### Returns:

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

**7.53.3.10** `template<class LLClass, class Alloc> int32 Osl_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 Osl_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 Osl_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 Osl_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 Osl_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 [Osl\\_Linked\\_List\\_Base](#).

**7.53.3.15** `template<class LLClass, class Alloc> int32 Osci_Linked_List< LLClass, Alloc  
>::remove_element (const LLClass & data_to_remove) [inline]`

Removes the element from the list.

**Parameters:**

*data\_to\_remove*

**Returns:**

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

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

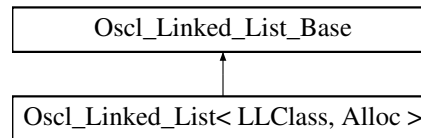
- [osci\\_linked\\_list.h](#)



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

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

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



### Protected Methods

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

### Protected Attributes

- [OslAny](#) \* [head](#)
- [OslAny](#) \* [tail](#)
- [OslAny](#) \* [iterator](#)
- int32 [num\\_elements](#)
- uint32 [sizeof\\_T](#)

### 7.54.1 Detailed Description

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

## 7.54.2 Constructor & Destructor Documentation

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

## 7.54.3 Member Function Documentation

**7.54.3.1** `OSCL_IMPORT_REF int32 Osl_Linked_List_Base::add_element (const OslAny * new_element) [protected]`

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

### Parameters:

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

### Returns:

32-bit integer on the success returns 1.

**7.54.3.2** `OSCL_IMPORT_REF int32 Osl_Linked_List_Base::add_to_front (const OslAny * new_element) [protected]`

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

### Parameters:

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

### Returns:

32-bit integer on the success returns 1.

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

Debug routine: Checks the list for elements.

### Returns:

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

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

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

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

**7.54.3.6** `OSCL_IMPORT_REF int32 Osl_Linked_List_Base::get_element (int32 index, OslAny * element) [protected]`

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

**Parameters:**

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

**Returns:**

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

### 7.54.3.7 OSCL\_IMPORT\_REF int32 Osl\_Linked\_List\_Base::get\_first (OslAny \* *ele*) [protected]

Return the first element of list in passed parameter,

**Parameters:**

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

**Returns:**

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

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

Returns the index for requested element.

**Parameters:**

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

**Returns:**

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

### 7.54.3.9 OSCL\_IMPORT\_REF int32 Osl\_Linked\_List\_Base::get\_next (OslAny \* *ele*) [protected]

Return the next element of list in passed parameter,

**Parameters:**

*ele* return the value of next element of list in this parameter

**Returns:**

32-bit integer ,if next element is found in list, it returns 1 otherwise it returns 0

### 7.54.3.10 OSCL\_IMPORT\_REF int32 Osl\_Linked\_List\_Base::insert\_element (const OslAny \* *new\_element*, int *index*) [protected]

Inserts new element in the list. If the index is past the end of the list it creates the list and add the element as first element of list.

**Parameters:**

*new\_element* the element to be add in the list.

**Returns:**

32-bit integer on the success returns 1.

**7.54.3.11 OSCL\_IMPORT\_REF int32 Osl\_Linked\_List\_Base::move\_to\_end (const [OslAny](#) \* *data\_to\_move*) [protected]**

Moves the element to end of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**7.54.3.12 OSCL\_IMPORT\_REF int32 Osl\_Linked\_List\_Base::move\_to\_front (const [OslAny](#) \* *data\_to\_move*) [protected]**

Moves the element to front of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**7.54.3.13 OSCL\_IMPORT\_REF int32 Osl\_Linked\_List\_Base::remove\_element (const int32 *index\_to\_remove*) [protected]**

Removes the element for requested index.

**Parameters:**

*index\_to\_remove*

**Returns:**

on success return 1 otherwise return 0.

Reimplemented in [Osl\\_Linked\\_List< LLClass, Alloc >](#).

**7.54.3.14 OSCL\_IMPORT\_REF int32 Osl\_Linked\_List\_Base::remove\_element (const [OslAny](#) \* *data\_to\_remove*) [protected]**

Removes the element from the list.

**Parameters:**

*data\_to\_remove*

**Returns:**

32-bit integer on if element found in the list returns 1 otherwise returns 0.

### 7.54.4 Field Documentation

7.54.4.1 [OslAny\\*](#) Osl\_Linked\_List\_Base::head [protected]

7.54.4.2 [OslAny\\*](#) Osl\_Linked\_List\_Base::iterator [protected]

7.54.4.3 int32 Osl\_Linked\_List\_Base::num\_elements [protected]

7.54.4.4 uint32 Osl\_Linked\_List\_Base::sizeof\_T [protected]

7.54.4.5 [OslAny\\*](#) Osl\_Linked\_List\_Base::tail [protected]

The documentation for this class was generated from the following file:

- [osl\\_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 = Osl\_Less<Key>> class Osl\_Map< Key, T, Alloc, Compare >**

Osl\_Map Class. A subset of STL::Map methods. Osl\_Map is a sorted associative container that associates objects of type Key with objects of type T. It is also a unique associative container, meaning that no two elements have the same key. Osl\_Map uses the key to speed lookup, insertion, and deletion of elements. It is often superior to all other containers when you need to lookup an element by key value. Due to the underlying tree structure, inserts and erases can be performed in logarithmic time, where a vector would take linear time in some cases.

## 7.55.2 Member Typedef Documentation

- 7.55.2.1 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::const_iterator Osl_Map< Key, T, Alloc, Compare >::const_iterator`
- 7.55.2.2 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::const_reference Osl_Map< Key, T, Alloc, Compare >::const_reference`
- 7.55.2.3 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::iterator Osl_Map< Key, T, Alloc, Compare >::iterator`
- 7.55.2.4 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Compare Osl_Map< Key, T, Alloc, Compare >::key_compare`
- 7.55.2.5 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Key Osl_Map< Key, T, Alloc, Compare >::key_type`
- 7.55.2.6 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Osl_Pair<const_iterator, const_iterator> Osl_Map< Key, T, Alloc, Compare >::pair_citerator_citerator`
- 7.55.2.7 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Osl_Pair<iterator, bool> Osl_Map< Key, T, Alloc, Compare >::pair_iterator_bool`
- 7.55.2.8 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Osl_Pair<iterator, iterator> Osl_Map< Key, T, Alloc, Compare >::pair_iterator_iterator`
- 7.55.2.9 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::pointer Osl_Map< Key, T, Alloc, Compare >::pointer`
- 7.55.2.10 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::reference Osl_Map< Key, T, Alloc, Compare >::reference`
- 7.55.2.11 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Osl_Map<Key, T, Alloc, Compare> Osl_Map< Key, T, Alloc, Compare >::self`
- 7.55.2.12 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef rep_type::size_type Osl_Map< Key, T, Alloc, Compare >::size_type`
- 7.55.2.13 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> typedef Osl_Pair<const Key, T> Osl_Map< Key, T, Alloc, Compare >::value_type`

## 7.55.3 Constructor & Destructor Documentation

- 7.55.3.1 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> Osl_Map< Key, T, Alloc, Compare >::Osl_Map (const Compare & comp = Compare()) [inline]`

Creates an empty map using comp as the key compare object



**7.55.3.2** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> Osl_Map<Key, T, Alloc, Compare >::Osl_Map (const self & x) [inline]`

Osl\_Map copy constructor

## 7.55.4 Member Function Documentation

**7.55.4.1** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> const\_iterator Osl_Map< Key, T, Alloc, Compare >::begin () const [inline]`

Returns a const iterator pointing to the beginning of the map

**7.55.4.2** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> iterator Osl_Map< Key, T, Alloc, Compare >::begin () [inline]`

Returns an iterator pointing to the beginning of the map

**7.55.4.3** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> void Osl_Map< Key, T, Alloc, Compare >::clear () [inline]`

Erases all elements

**7.55.4.4** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> size\_type Osl_Map< Key, T, Alloc, Compare >::count (const key\_type & x) const [inline]`

Returns the number of elements with key x. For map this will either be 0 or 1.

**7.55.4.5** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> bool Osl_Map< Key, T, Alloc, Compare >::empty () const [inline]`

Returns true if map size is 0

**7.55.4.6** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> const\_iterator Osl_Map< Key, T, Alloc, Compare >::end () const [inline]`

Returns a const iterator pointing to the end of the map.

**7.55.4.7** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> iterator Osl_Map< Key, T, Alloc, Compare >::end () [inline]`

Returns an iterator pointing to the end of the map.

**7.55.4.8** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> pair\_citerator\_citerator Osl_Map< Key, T, Alloc, Compare >::equal_range (const key\_type & x) const [inline]`

Finds a range containing all elements whose key is x

**7.55.4.9** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>  
pair_iterator_iterator Osl_Map< Key, T, Alloc, Compare >::equal_range (const  
key_type &x) [inline]`

Finds a range containing all elements whose key is x

**7.55.4.10** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> void  
Osl_Map< Key, T, Alloc, Compare >::erase (iterator first, iterator last) [inline]`

Erases all elements in the range [first,last)

**7.55.4.11** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> size_type  
Osl_Map< Key, T, Alloc, Compare >::erase (const key_type &x) [inline]`

Erases the element with key x

**7.55.4.12** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> void  
Osl_Map< Key, T, Alloc, Compare >::erase (iterator position) [inline]`

Erases the element pointed to by position

**7.55.4.13** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>  
const_iterator Osl_Map< Key, T, Alloc, Compare >::find (const key_type &x) const  
[inline]`

Finds an element whose key is x

**7.55.4.14** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> iterator  
Osl_Map< Key, T, Alloc, Compare >::find (const key_type &x) [inline]`

Finds an element whose key is x

**7.55.4.15** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> void  
Osl_Map< Key, T, Alloc, Compare >::insert (const value_type *first, const value_type  
*last) [inline]`

Inserts the range [first,last) into the map

**7.55.4.16** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> iterator  
Osl_Map< Key, T, Alloc, Compare >::insert (iterator position, const value_type &x)  
[inline]`

Inserts x into the map using position as a hint as to where it should be inserted

**7.55.4.17** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>  
pair_iterator_bool Osl_Map< Key, T, Alloc, Compare >::insert (const value_type & x)  
[inline]`

Inserts x into the map

**7.55.4.18** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>  
key_compare Osl_Map< Key, T, Alloc, Compare >::key_comp () const [inline]`

Returns the key compare object used by the map

**7.55.4.19** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>  
const_iterator Osl_Map< Key, T, Alloc, Compare >::lower_bound (const key_type &  
x) const [inline]`

Finds the first element whose key is not less than x

**7.55.4.20** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> iterator  
Osl_Map< Key, T, Alloc, Compare >::lower_bound (const key_type & x) [inline]`

Finds the first element whose key is not less than x

**7.55.4.21** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> size_type  
Osl_Map< Key, T, Alloc, Compare >::max_size () const [inline]`

Returns the maximum possible size of the map

**7.55.4.22** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> self&  
Osl_Map< Key, T, Alloc, Compare >::operator= (const self & x) [inline]`

Osl\_Map assignment operator

**7.55.4.23** ]

`template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> T& Osl_Map< Key, T,  
Alloc, Compare >::operator[] (const key_type & k) [inline]`

Returns a reference to the object that is associated with a particular key. If the map does not already contain such an object, operator[] inserts the default object value\_type().

**7.55.4.24** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> size_type  
Osl_Map< Key, T, Alloc, Compare >::size () const [inline]`

Returns the size of the map

**7.55.4.25** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>  
const_iterator Osl_Map< Key, T, Alloc, Compare >::upper_bound (const key_type &  
x) const [inline]`

Finds the first element whose key is not greater than x

**7.55.4.26** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> iterator  
Osl_Map< Key, T, Alloc, Compare >::upper_bound (const key_type &x) [inline]`

Finds the first element whose key is not greater than x

**7.55.4.27** `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>>  
value_compare Osl_Map< Key, T, Alloc, Compare >::value_comp () const  
[inline]`

Returns the value compare object used by the map

The documentation for this class was generated from the following file:

- [osl\\_map.h](#)

## 7.56 Osl\_Map< Key, T, Alloc, Compare >::value\_compare Class Reference

```
#include <osl_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 [Osl\\_Map< Key, T, Alloc, Compare >](#)

```
template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> class Osl_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 = Osl_Less<Key>> Osl\_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 = Osl_Less<Key>> bool  
Osl\_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 = Osl_Less<Key>> friend class  
Osl\_Map< Key, T, Alloc, Compare > [friend]`

### 7.56.4 Field Documentation

7.56.4.1 `template<class Key, class T, class Alloc, class Compare = Osl_Less<Key>> Compare  
Osl\_Map< Key, T, Alloc, Compare >::value_compare::comp [protected]`

The documentation for this class was generated from the following file:

- [osl\\_map.h](#)

## 7.57 Osl\_MTLINKED\_List< LLClass, Alloc, TheLock > Class Template Reference

```
#include <osl_linked_list.h>
```

### Public Methods

- [Osl\\_MTLINKED\\_List \(\)](#)
- [~Osl\\_MTLINKED\\_List \(\)](#)
- [int32 dequeue\\_element \(LLClass &element\)](#)
- [int32 add\\_element \(LLClass &new\\_element\)](#)
- [int32 add\\_to\\_front \(LLClass &new\\_element\)](#)
- [uint32 get\\_element \(int32 index, LLClass &element\)](#)
- [int32 remove\\_element \(const LLClass &data\\_to\\_remove\)](#)
- [int32 get\\_index \(const LLClass &data\)](#)
- [int32 remove\\_element \(const int32 index\\_to\\_remove\)](#)
- [int32 move\\_to\\_end \(const LLClass &data\\_to\\_move\)](#)
- [int32 move\\_to\\_front \(const LLClass &data\\_to\\_move\)](#)

### Protected Attributes

- [Osl\\_Linked\\_List< LLClass, Alloc > the\\_list](#)

### 7.57.1 Detailed Description

**template<class LLClass, class Alloc, class TheLock> class Osl\_MTLINKED\_List< LLClass, Alloc, TheLock >**

Osl\_MTLINKED\_List is a multi-threaded version of the LinkedList. It has mutex protection to allow access by different threads.

### 7.57.2 Constructor & Destructor Documentation

**7.57.2.1 template<class LLClass, class Alloc, class TheLock> Osl\_MTLINKED\_List< LLClass, Alloc, TheLock >::Osl\_MTLINKED\_List () [inline]**

Constructor for Osl\_MTLINKED\_List

**7.57.2.2 template<class LLClass, class Alloc, class TheLock> Osl\_MTLINKED\_List< LLClass, Alloc, TheLock >::~~Osl\_MTLINKED\_List () [inline]**

Destructor for Osl\_MTLINKED\_List

## 7.57.3 Member Function Documentation

**7.57.3.1** `template<class LLClass, class Alloc, class TheLock> int32 Osci_MTLINKED_List<LLClass, Alloc, TheLock >::add_element (LLClass & new_element) [inline]`

Adds new element to the Multi Threaded Linked list. if list is already there then it adds element at end of list otherwise it 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 Osci_MTLINKED_List<LLClass, Alloc, TheLock >::add_to_front (LLClass & new_element) [inline]`

Adds new element at the start of the Multi Threaded Linked list. if list 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 Osci_MTLINKED_List<LLClass, Alloc, TheLock >::dequeue_element (LLClass & element) [inline]`

**7.57.3.4** `template<class LLClass, class Alloc, class TheLock> uint32 Osci_MTLINKED_List<LLClass, Alloc, TheLock >::get_element (int32 index, LLClass & element) [inline]`

Search and returns the element in the Multi Threaded Linked List for passed index.

**Parameters:**

*index, element* The index is the count for the node.

**Returns:**

32-bit integer on success returns 1 otherwise returns 0.

**7.57.3.5** `template<class LLClass, class Alloc, class TheLock> int32 Osci_MTLINKED_List<LLClass, Alloc, TheLock >::get_index (const LLClass & data) [inline]`

Returns the index for requested element.

**Parameters:**

*data* the element for which index to be 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 Osci_MTLINKED_List< LLClass, Alloc, TheLock >::move_to_end (const LLClass & data_to_move) [inline]`

Moves the element to end of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**7.57.3.7** `template<class LLClass, class Alloc, class TheLock> int32 Osci_MTLINKED_List< LLClass, Alloc, TheLock >::move_to_front (const LLClass & data_to_move) [inline]`

Moves the element to front of the list

**Parameters:**

*data\_to\_move*

**Returns:**

On success returns 1 otherwise returns 0.

**7.57.3.8** `template<class LLClass, class Alloc, class TheLock> int32 Osci_MTLINKED_List< LLClass, Alloc, TheLock >::remove_element (const int32 index_to_remove) [inline]`

Removes the element for requested index.

**Parameters:**

*index\_to\_remove*

**Returns:**

on success return 1 otherwise return 0.

**7.57.3.9** `template<class LLClass, class Alloc, class TheLock> int32 Osci_MTLINKED_List< LLClass, Alloc, TheLock >::remove_element (const LLClass & data_to_remove) [inline]`

Removes the element from the list.

**Parameters:**

*data\_to\_remove*

**Returns:**

32-bit integer on if element found in the list returns 1 otherwise returns 0.

## 7.57.4 Field Documentation

**7.57.4.1** `template<class LLClass, class Alloc, class TheLock> Osci\_Linked\_List<LLClass, Alloc> Osci_MTLINKED_List< LLClass, Alloc, TheLock >::the_list [protected]`

The documentation for this class was generated from the following file:



## 7.57 Osci\_MTLINKed\_List< LLClass, Alloc, TheLock > Class Template Reference

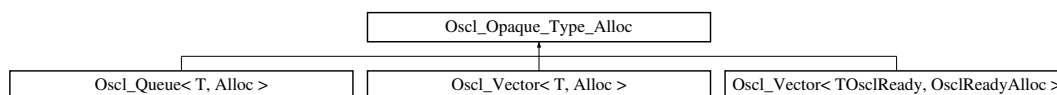
---

- [osci\\_linked\\_list.h](#)

## 7.58 Osci\_Opaque\_Type\_Alloc Class Reference

```
#include <osci_opaque_type.h>
```

Inheritance diagram for Osci\_Opaque\_Type\_Alloc::



### Public Methods

- virtual `~Osci_Opaque_Type_Alloc()`
- virtual void `construct(OsciAny *p, const OsciAny *init_val)=0`
- virtual void `destroy(OsciAny *p)=0`
- virtual `OsciAny *allocate(const uint32 size)=0`
- virtual void `deallocate(OsciAny *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 `Osci_Opaque_Type_Alloc::~~Osci_Opaque_Type_Alloc()` [inline, virtual]

### 7.58.3 Member Function Documentation

**7.58.3.1** virtual `OsciAny* Osci_Opaque_Type_Alloc::allocate(const uint32 size)` [pure virtual]

Allocate "size" bytes

**7.58.3.2** virtual void `Osci_Opaque_Type_Alloc::construct(OsciAny *p, const OsciAny *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 `Osci_Opaque_Type_Alloc::deallocate(OsciAny *p)` [pure virtual]

Deallocate memory previously allocated with "allocate"

**7.58.3.4** virtual void `Osci_Opaque_Type_Alloc::destroy(OsciAny *p)` [pure virtual]

Destroy element at p.

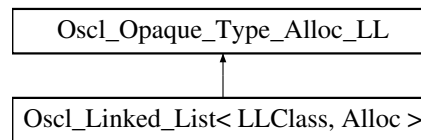
The documentation for this class was generated from the following file:

- [oscl\\_opaque\\_type.h](#)

## 7.59 Osci\_Opaque\_Type\_Alloc\_LL Class Reference

```
#include <osci_opaque_type.h>
```

Inheritance diagram for Osci\_Opaque\_Type\_Alloc\_LL::



### Public Methods

- virtual `~Osci_Opaque_Type_Alloc_LL()`
- virtual void `construct` (`OsciAny` \*p, const `OsciAny` \*init\_val)=0
- virtual void `destroy` (`OsciAny` \*p)=0
- virtual `OsciAny` \* `allocate` (const uint32 size)=0
- virtual void `deallocate` (`OsciAny` \*p)=0
- virtual `OsciAny` \* `get_next` (const `OsciAny` \*elem) const=0
- virtual void `set_next` (`OsciAny` \*elem, const `OsciAny` \*nextelem)=0
- virtual void `get_data` (`OsciAny` \*elem, `OsciAny` \*data\_val)=0
- virtual bool `compare_data` (const `OsciAny` \*elem, const `OsciAny` \*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 `Osci_Opaque_Type_Alloc_LL::~~Osci_Opaque_Type_Alloc_LL()` [inline, virtual]

### 7.59.3 Member Function Documentation

**7.59.3.1** virtual `OsciAny`\* `Osci_Opaque_Type_Alloc_LL::allocate` (const uint32 size) [pure virtual]

Allocate "size" bytes

**7.59.3.2** virtual bool `Osci_Opaque_Type_Alloc_LL::compare_data` (const `OsciAny` \* elem, const `OsciAny` \*data\_val) const [pure virtual]

Compare data.

**7.59.3.3** virtual void `Osci_Opaque_Type_Alloc_LL::construct` (`OsciAny` \* p, const `OsciAny` \*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 Osci_Opaque_Type_Alloc_LL::deallocate (OsciAny * p)` [pure virtual]

Deallocate memory previously allocated with "allocate"

**7.59.3.5** `virtual void Osci_Opaque_Type_Alloc_LL::destroy (OsciAny * p)` [pure virtual]

Destroy element at *p*.

**7.59.3.6** `virtual void Osci_Opaque_Type_Alloc_LL::get_data (OsciAny * elem, OsciAny * data_val)` [pure virtual]

Get data

**7.59.3.7** `virtual OsciAny* Osci_Opaque_Type_Alloc_LL::get_next (const OsciAny * elem) const` [pure virtual]

Get next element in linked list.

**7.59.3.8** `virtual void Osci_Opaque_Type_Alloc_LL::set_next (OsciAny * elem, const OsciAny * nextelem)` [pure virtual]

Set next element in linked list.

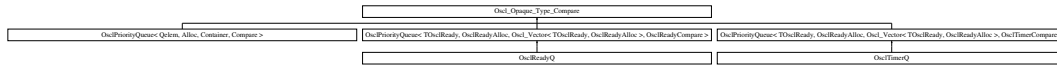
The documentation for this class was generated from the following file:

- [osci\\_opaque\\_type.h](#)

## 7.60 Osci\_Opaque\_Type\_Compare Class Reference

```
#include <osci_opaque_type.h>
```

Inheritance diagram for Osci\_Opaque\_Type\_Compare::



### Public Methods

- virtual [~Osci\\_Opaque\\_Type\\_Compare\(\)](#)
- virtual void [swap](#) (OsciAny \*a, const OsciAny \*b)=0
- virtual int [compare\\_LT](#) (OsciAny \*a, OsciAny \*b) const=0
- virtual int [compare\\_EQ](#) (const OsciAny \*a, const OsciAny \*b) const=0

### 7.60.1 Detailed Description

Opaque type operations with swap & comparisons.

### 7.60.2 Constructor & Destructor Documentation

**7.60.2.1** virtual Osci\_Opaque\_Type\_Compare::~~Osci\_Opaque\_Type\_Compare() [inline, virtual]

### 7.60.3 Member Function Documentation

**7.60.3.1** virtual int Osci\_Opaque\_Type\_Compare::compare\_EQ (const OsciAny \*a, const OsciAny \*b) const [pure virtual]

Return a==b.

Implemented in [OsciPriorityQueue< Qelem, Alloc, Container, Compare >](#), [OsciPriorityQueue< TOsciReady, OsciReadyAlloc, Osci\\_Vector< TOsciReady, OsciReadyAlloc >, OsciReadyCompare >](#), and [OsciPriorityQueue< TOsciReady, OsciReadyAlloc, Osci\\_Vector< TOsciReady, OsciReadyAlloc >, OsciTimerCompare >](#).

**7.60.3.2** virtual int Osci\_Opaque\_Type\_Compare::compare\_LT (OsciAny \*a, OsciAny \*b) const [pure virtual]

Return a<b.

Implemented in [OsciPriorityQueue< Qelem, Alloc, Container, Compare >](#), [OsciPriorityQueue< TOsciReady, OsciReadyAlloc, Osci\\_Vector< TOsciReady, OsciReadyAlloc >, OsciReadyCompare >](#), and [OsciPriorityQueue< TOsciReady, OsciReadyAlloc, Osci\\_Vector< TOsciReady, OsciReadyAlloc >, OsciTimerCompare >](#).

### 7.60.3.3 `virtual void Osl_Opaque_Type_Compare::swap (OslAny * a, const OslAny * b)` [pure virtual]

Swap element at "*a*" with element at "*b*". Both pointers must be non-NULL.

Implemented in [OslPriorityQueue< Qelem, Alloc, Container, Compare >](#), [OslPriorityQueue< TOslReady, OslReadyAlloc, Osl\\_Vector< TOslReady, OslReadyAlloc >, OslReadyCompare >](#), and [OslPriorityQueue< TOslReady, OslReadyAlloc, Osl\\_Vector< TOslReady, OslReadyAlloc >, Osl-TimerCompare >](#).

The documentation for this class was generated from the following file:

- [osl\\_opaque\\_type.h](#)



## 7.61 Osl\_Pair< T1, T2 > Struct Template Reference

```
#include <oscl_tree.h>
```

### Public Methods

- [Osl\\_Pair](#) ()
- [Osl\\_Pair](#) (const T1 &a, const T2 &b)

### Data Fields

- T1 [first](#)
- T2 [second](#)

```
template<class T1, class T2> struct Osl_Pair< T1, T2 >
```

### 7.61.1 Constructor & Destructor Documentation

7.61.1.1 `template<class T1, class T2> Osl_Pair< T1, T2 >::Osl_Pair () [inline]`

7.61.1.2 `template<class T1, class T2> Osl_Pair< T1, T2 >::Osl_Pair (const T1 &a, const T2 &b) [inline]`

### 7.61.2 Field Documentation

7.61.2.1 `template<class T1, class T2> T1 Osl_Pair< T1, T2 >::first`

7.61.2.2 `template<class T1, class T2> T2 Osl_Pair< T1, T2 >::second`

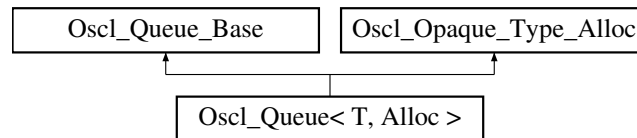
The documentation for this struct was generated from the following file:

- [oscl\\_tree.h](#)

## 7.62 Osl\_Queue< T, Alloc > Class Template Reference

```
#include <osl_queue.h>
```

Inheritance diagram for Osl\_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

- [Osl\\_Queue](#) ()
- [Osl\\_Queue](#) (uint32 n)
- virtual [~Osl\\_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 Osl_Queue< T, Alloc >
```

Osl\_Queue Class. A subset of STL::Queue methods. Osl\_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& Osl_Queue< T, Alloc >::const_reference`

**7.62.2.2** `template<class T, class Alloc> typedef T* Osl_Queue< T, Alloc >::pointer`

**7.62.2.3** `template<class T, class Alloc> typedef T& Osl_Queue< T, Alloc >::reference`

**7.62.2.4** `template<class T, class Alloc> typedef uint32 Osl_Queue< T, Alloc >::size_type`

**7.62.2.5** `template<class T, class Alloc> typedef T Osl_Queue< T, Alloc >::value_type`

## 7.62.3 Constructor & Destructor Documentation

**7.62.3.1** `template<class T, class Alloc> Osl_Queue< T, Alloc >::Osl_Queue () [inline]`

Creates an empty queue.

**7.62.3.2** `template<class T, class Alloc> Osl_Queue< T, Alloc >::Osl_Queue (uint32 n) [inline]`

Creates an empty queue with capacity *n*.

### Parameters:

*n* creates a queue with *n* elements. The main reason for specifying *n* is efficiency. If you know the capacity to which your queue must grow, then it is more efficient to allocate the queue all at once rather than rely on the automatic reallocation scheme.

**7.62.3.3** `template<class T, class Alloc> virtual Osl_Queue< T, Alloc >::~~Osl_Queue () [inline, virtual]`

The destructor.

## 7.62.4 Member Function Documentation

**7.62.4.1** `template<class T, class Alloc> const_reference Osl_Queue< T, Alloc >::back () const [inline]`

Returns the last element (const)

**7.62.4.2** `template<class T, class Alloc> reference Osl_Queue< T, Alloc >::back () [inline]`

Returns the last element: "back" (generally not too useful, but some debugging aids might want to find out what was just added)

**7.62.4.3** `template<class T, class Alloc> void Osl_Queue< T, Alloc >::clear () [inline]`

Removes all elements.

Reimplemented from [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

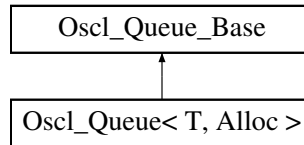
The documentation for this class was generated from the following file:

- [oscl\\_queue.h](#)

## 7.63 Osl\_Queue\_Base Class Reference

```
#include <osl_queue.h>
```

Inheritance diagram for Osl\_Queue\_Base::



### Public Methods

- uint32 [size](#) () const
- uint32 [capacity](#) () const
- bool [empty](#) () const
- OSCL\_IMPORT\_REF void [reserve](#) (uint32 n)

### Protected Methods

- OSCL\_IMPORT\_REF void [construct](#) (Osl\_Opaque\_Type\_Alloc \*aType)
- OSCL\_IMPORT\_REF void [construct](#) (Osl\_Opaque\_Type\_Alloc \*aType, uint32 n)
- virtual [~Osl\\_Queue\\_Base](#) ()
- OSCL\_IMPORT\_REF void [destroy](#) ()
- OSCL\_IMPORT\_REF void [push](#) (const OslAny \*x)
- OSCL\_IMPORT\_REF void [pop](#) ()
- OSCL\_IMPORT\_REF void [clear](#) ()

### Protected Attributes

- uint32 [numelems](#)
- uint32 [bufsize](#)
- OslAny \* [elems](#)
- uint32 [sizeof\\_T](#)
- uint32 [ifront](#)
- uint32 [irear](#)

### 7.63.1 Detailed Description

Osl\_Queue\_Base is a non-templated base class for [Osl\\_Queue](#). The purpose of this base class is to avoid large inline routines in the [Osl\\_Queue](#) implementation. This class is not intended for direct instantiation except by [Osl\\_Queue](#).

### 7.63.2 Constructor & Destructor Documentation

**7.63.2.1** virtual Osl\_Queue\_Base::~~Osl\_Queue\_Base () [inline, protected, virtual]

The destructor.

## 7.63.3 Member Function Documentation

### 7.63.3.1 uint32 Osl\_Queue\_Base::capacity () const [inline]

Returns the allocated memory of the queue.

### 7.63.3.2 OSCL\_IMPORT\_REF void Osl\_Queue\_Base::clear () [protected]

Removes all elements.

Reimplemented in [Osl\\_Queue< T, Alloc >](#).

### 7.63.3.3 OSCL\_IMPORT\_REF void Osl\_Queue\_Base::construct (Osl\_Opaque\_Type\_Alloc \* aType, uint32 n) [protected]

### 7.63.3.4 OSCL\_IMPORT\_REF void Osl\_Queue\_Base::construct (Osl\_Opaque\_Type\_Alloc \* aType) [protected]

### 7.63.3.5 OSCL\_IMPORT\_REF void Osl\_Queue\_Base::destroy () [protected]

Like an explicit destructor call.

### 7.63.3.6 bool Osl\_Queue\_Base::empty () const [inline]

True if there are no elements in the queue

### 7.63.3.7 OSCL\_IMPORT\_REF void Osl\_Queue\_Base::pop () [protected]

Removes the first element

Reimplemented in [Osl\\_Queue< T, Alloc >](#).

### 7.63.3.8 OSCL\_IMPORT\_REF void Osl\_Queue\_Base::push (const OslAny \* x) [protected]

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL\_LEAVE will occur

#### Parameters:

*x* new element

### 7.63.3.9 OSCL\_IMPORT\_REF void Osl\_Queue\_Base::reserve (uint32 n)

Reallocates memory if necessary to a capacity of n elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

#### Parameters:

*n* size of vector

**7.63.3.10** `uint32 Osl_Queue_Base::size () const` `[inline]`

Returns the size of the queue.

**7.63.4 Field Documentation****7.63.4.1** `uint32 Osl_Queue_Base::bufsize` `[protected]`**7.63.4.2** `OslAny* Osl_Queue_Base::elems` `[protected]`**7.63.4.3** `uint32 Osl_Queue_Base::ifront` `[protected]`**7.63.4.4** `uint32 Osl_Queue_Base::irear` `[protected]`**7.63.4.5** `uint32 Osl_Queue_Base::numelems` `[protected]`**7.63.4.6** `uint32 Osl_Queue_Base::sizeof_T` `[protected]`

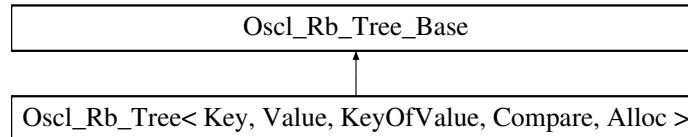
The documentation for this class was generated from the following file:

- [osl\\_queue.h](#)

## 7.64 Osl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

```
#include <osl_tree.h>
```

Inheritance diagram for Osl\_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 Osl\_Rb\_Tree\_Node< Value >::link\_type [link\\_type](#)
- typedef Osl\_Rb\_Tree\_Iterator< [value\\_type](#) > [iterator](#)
- typedef Osl\_Rb\_Tree\_Const\_Iterator< [value\\_type](#) > [const\\_iterator](#)
- typedef uint32 [size\\_type](#)
- typedef int32 [difference\\_type](#)

### Public Methods

- [Osl\\_Rb\\_Tree](#) (const Compare &comp=Compare())
- [Osl\\_Rb\\_Tree](#) (const Osl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > &x)
- [~Osl\\_Rb\\_Tree](#) ()
- Osl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > & [operator=](#) (const Osl\_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
- Osl\_Pair< [iterator](#), bool > [insert\\_unique](#) (const [value\\_type](#) &v)
- [iterator insert\\_unique](#) ([iterator](#) position, const [value\\_type](#) &v)
- void [insert\\_unique](#) (const [const\\_iterator](#) first, const [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 `OscI_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`
- `OscI_Pair< iterator, iterator > equal_range (const Key &k)`
- `OscI_Pair< const_iterator, const_iterator > equal_range (const Key &k) const`

```
template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> class Osl_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 Osl\_Rb\_Tree\_Const\_Iterator<value\_type> Osl_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* Osl_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& Osl_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 Osl_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 Osl\_Rb\_Tree\_Iterator<value\_type> Osl_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 Osl_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 Osl\_Rb\_Tree\_Node<Value>::link_type Osl_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* Osl_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& Osl_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 Osl_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 Osl_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> Osl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::Osl_Rb_Tree (const Compare & comp = Compare()) [inline]`

7.64.2.2 `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Osl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::Osl_Rb_Tree (const Osl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > & x) [inline]`

7.64.2.3 `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Osl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::~Osl_Rb_Tree () [inline]`



## 7.64 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

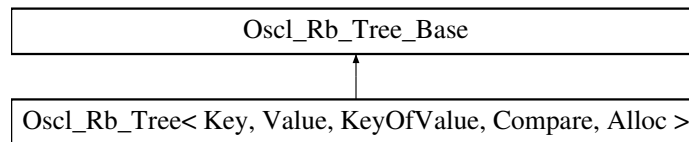
---

- [oscl\\_tree.h](#)

## 7.65 Osl\_Rb\_Tree\_Base Class Reference

```
#include <osl_tree.h>
```

Inheritance diagram for Osl\_Rb\_Tree\_Base::



### Public Types

- typedef Osl\_Rb\_Tree\_Node\_Base::base\_link\_type [base\\_link\\_type](#)

### Public Methods

- OSCL\_IMPORT\_REF void [rotate\\_left](#) ([base\\_link\\_type](#) x, [base\\_link\\_type](#) &root)
- OSCL\_IMPORT\_REF void [rotate\\_right](#) ([base\\_link\\_type](#) x, [base\\_link\\_type](#) &root)
- OSCL\_IMPORT\_REF void [rebalance](#) ([base\\_link\\_type](#) x, [base\\_link\\_type](#) &root)
- OSCL\_IMPORT\_REF [base\\_link\\_type](#) [rebalance\\_for\\_erase](#) ([base\\_link\\_type](#) z, [base\\_link\\_type](#) &root, [base\\_link\\_type](#) &leftmost, [base\\_link\\_type](#) &rightmost)

### 7.65.1 Member Typedef Documentation

7.65.1.1 typedef Osl\_Rb\_Tree\_Node\_Base::base\_link\_type Osl\_Rb\_Tree\_Base::base\_link\_type

### 7.65.2 Member Function Documentation

7.65.2.1 OSCL\_IMPORT\_REF void Osl\_Rb\_Tree\_Base::rebalance ([base\\_link\\_type](#) x, [base\\_link\\_type](#) &root)

7.65.2.2 OSCL\_IMPORT\_REF [base\\_link\\_type](#) Osl\_Rb\_Tree\_Base::rebalance\_for\_erase ([base\\_link\\_type](#) z, [base\\_link\\_type](#) &root, [base\\_link\\_type](#) &leftmost, [base\\_link\\_type](#) &rightmost)

7.65.2.3 OSCL\_IMPORT\_REF void Osl\_Rb\_Tree\_Base::rotate\_left ([base\\_link\\_type](#) x, [base\\_link\\_type](#) &root)

7.65.2.4 OSCL\_IMPORT\_REF void Osl\_Rb\_Tree\_Base::rotate\_right ([base\\_link\\_type](#) x, [base\\_link\\_type](#) &root)

The documentation for this class was generated from the following file:

- [osl\\_tree.h](#)

## 7.66 Osl\_Rb\_Tree\_Const\_Iterator< Value > Struct Template Reference

```
#include <osl_tree.h>
```

### Public Types

- typedef Value [value\\_type](#)
- typedef const [value\\_type](#) & [reference](#)
- typedef const [value\\_type](#) \* [pointer](#)
- typedef Osl\_Rb\_Tree\_Const\_Iterator< Value > [const\\_iterator](#)
- typedef Osl\_Rb\_Tree\_Const\_Iterator< Value > [self](#)
- typedef [Osl\\_Rb\\_Tree\\_Node\\_Base](#) \* [base\\_link\\_type](#)
- typedef [Osl\\_Rb\\_Tree\\_Node](#)< Value > \* [link\\_type](#)

### Public Methods

- [Osl\\_Rb\\_Tree\\_Const\\_Iterator](#) ()
- [Osl\\_Rb\\_Tree\\_Const\\_Iterator](#) ([link\\_type](#) x)
- [Osl\\_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 Osl_Rb_Tree_Const_Iterator< Value >
```

### 7.66.1 Member Typedef Documentation

7.66.1.1 `template<class Value> typedef Osl\_Rb\_Tree\_Node\_Base\*  
Osl_Rb_Tree_Const_Iterator< Value >::base_link_type`

7.66.1.2 `template<class Value> typedef Osl_Rb_Tree_Const_Iterator<Value>  
Osl_Rb_Tree_Const_Iterator< Value >::const_iterator`

7.66.1.3 `template<class Value> typedef Osl\_Rb\_Tree\_Node<Value>*  
Osl_Rb_Tree_Const_Iterator< Value >::link_type`

7.66.1.4 `template<class Value> typedef const value\_type* Osl_Rb_Tree_Const_Iterator< Value  
>::pointer`

7.66.1.5 `template<class Value> typedef const value\_type& Osl_Rb_Tree_Const_Iterator< Value  
>::reference`

7.66.1.6 `template<class Value> typedef Osl_Rb_Tree_Const_Iterator<Value>  
Osl_Rb_Tree_Const_Iterator< Value >::self`

7.66.1.7 `template<class Value> typedef Value Osl_Rb_Tree_Const_Iterator< Value  
>::value_type`

### 7.66.2 Constructor & Destructor Documentation

7.66.2.1 `template<class Value> Osl_Rb_Tree_Const_Iterator< Value  
>::Osl_Rb_Tree_Const_Iterator() [inline]`

7.66.2.2 `template<class Value> Osl_Rb_Tree_Const_Iterator< Value  
>::Osl_Rb_Tree_Const_Iterator(link\_type x) [inline]`

7.66.2.3 `template<class Value> Osl_Rb_Tree_Const_Iterator< Value  
>::Osl_Rb_Tree_Const_Iterator(const const\_iterator &it) [inline]`

### 7.66.3 Member Function Documentation

7.66.3.1 `template<class Value> reference Osl_Rb_Tree_Const_Iterator< Value >::operator * ()  
const [inline]`

7.66.3.2 `template<class Value> bool Osl_Rb_Tree_Const_Iterator< Value >::operator!= (const  
self &x) [inline]`

7.66.3.3 `template<class Value> self Osl_Rb_Tree_Const_Iterator< Value >::operator++ (int)  
[inline]`

7.66.3.4 `template<class Value> self& Osl_Rb_Tree_Const_Iterator< Value >::operator++ ()  
[inline]`

7.66.3.5 `template<class Value> self Osl_Rb_Tree_Const_Iterator< Value >::operator- (int)  
[inline]`

7.66.3.6 `template<class Value> self& Osl_Rb_Tree_Const_Iterator< Value >::operator- ()  
[inline]`

- [osl\\_tree.h](#)

## 7.67 Osl\_Rb\_Tree\_Iterator< Value > Struct Template Reference

```
#include <osl_tree.h>
```

### Public Types

- typedef Value [value\\_type](#)
- typedef [value\\_type](#) & [reference](#)
- typedef [value\\_type](#) \* [pointer](#)
- typedef Osl\_Rb\_Tree\_Iterator< Value > [iterator](#)
- typedef Osl\_Rb\_Tree\_Iterator< Value > [self](#)
- typedef [Osl\\_Rb\\_Tree\\_Node\\_Base](#) \* [base\\_link\\_type](#)
- typedef [Osl\\_Rb\\_Tree\\_Node](#)< Value > \* [link\\_type](#)

### Public Methods

- [Osl\\_Rb\\_Tree\\_Iterator](#) ()
- [Osl\\_Rb\\_Tree\\_Iterator](#) ([link\\_type](#) x)
- [Osl\\_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]`

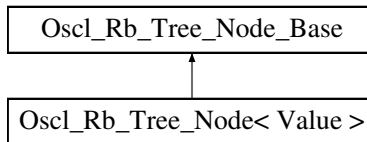
7.67.3.7 `template<class Value> pointer Oscl_Rb_Tree_Iterator< Value >::operator → () const [inline]`

- [osl\\_tree.h](#)

## 7.68 Osl\_Rb\_Tree\_Node< Value > Struct Template Reference

```
#include <osl_tree.h>
```

Inheritance diagram for Osl\_Rb\_Tree\_Node< Value >::



### Public Types

- typedef Value [value\\_type](#)
- typedef Osl\_Rb\_Tree\_Node< Value > \* [link\\_type](#)

### Data Fields

- [value\\_type](#) value

```
template<class Value> struct Osl_Rb_Tree_Node< Value >
```

### 7.68.1 Member Typedef Documentation

**7.68.1.1** `template<class Value> typedef Osl_Rb_Tree_Node<Value>* Osl_Rb_Tree_Node<Value >::link_type`

**7.68.1.2** `template<class Value> typedef Value Osl_Rb_Tree_Node< Value >::value_type`

### 7.68.2 Field Documentation

**7.68.2.1** `template<class Value> value\_type Osl_Rb_Tree_Node< Value >::value`

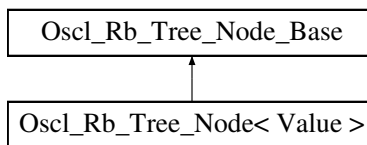
The documentation for this struct was generated from the following file:

- [osl\\_tree.h](#)

## 7.69 Osl\_Rb\_Tree\_Node\_Base Struct Reference

```
#include <osl_tree.h>
```

Inheritance diagram for Osl\_Rb\_Tree\_Node\_Base::



### Public Types

- typedef Osl\_Rb\_Tree\_Node\_Base \* [base\\_link\\_type](#)
- typedef enum [Osl\\_Rb\\_Tree\\_Node\\_Base::RedBl](#) [color\\_type](#)
- enum [RedBl](#) { [red](#), [black](#) }

### Static Public Methods

- [base\\_link\\_type](#) [minimum](#) ([base\\_link\\_type](#) x)
- [base\\_link\\_type](#) [maximum](#) ([base\\_link\\_type](#) x)

### Data Fields

- [color\\_type](#) [color](#)
- [base\\_link\\_type](#) [parent](#)
- [base\\_link\\_type](#) [left](#)
- [base\\_link\\_type](#) [right](#)

### 7.69.1 Member Typedef Documentation

**7.69.1.1** typedef Osl\_Rb\_Tree\_Node\_Base\* Osl\_Rb\_Tree\_Node\_Base::base\_link\_type

**7.69.1.2** typedef enum [Osl\\_Rb\\_Tree\\_Node\\_Base::RedBl](#) Osl\_Rb\_Tree\_Node\_Base::color\_type

### 7.69.2 Member Enumeration Documentation

**7.69.2.1** enum Osl\_Rb\_Tree\_Node\_Base::RedBl

Enumeration values:

**red**

**black**

### 7.69.3 Member Function Documentation

7.69.3.1 [base\\_link\\_type](#) Osl\_Rb\_Tree\_Node\_Base::maximum ([base\\_link\\_type](#) *x*) [inline, static]

7.69.3.2 [base\\_link\\_type](#) Osl\_Rb\_Tree\_Node\_Base::minimum ([base\\_link\\_type](#) *x*) [inline, static]

### 7.69.4 Field Documentation

7.69.4.1 [color\\_type](#) Osl\_Rb\_Tree\_Node\_Base::color

7.69.4.2 [base\\_link\\_type](#) Osl\_Rb\_Tree\_Node\_Base::left

7.69.4.3 [base\\_link\\_type](#) Osl\_Rb\_Tree\_Node\_Base::parent

7.69.4.4 [base\\_link\\_type](#) Osl\_Rb\_Tree\_Node\_Base::right

The documentation for this struct was generated from the following file:

- [osl\\_tree.h](#)

## 7.70 Osl\_Select1st< V, U > Struct Template Reference

```
#include <osl_map.h>
```

### Public Methods

- const U & [operator\(\)](#) (const V &x) const

```
template<class V, class U> struct Osl_Select1st< V, U >
```

### 7.70.1 Member Function Documentation

**7.70.1.1** `template<class V, class U> const U& Osl_Select1st< V, U >::operator() (const V & x)`  
`const [inline]`

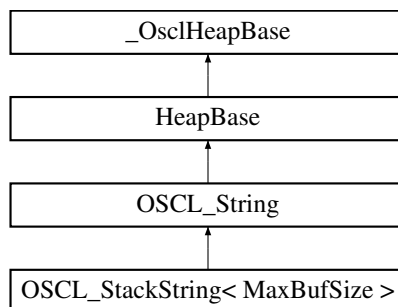
The documentation for this struct was generated from the following file:

- [osl\\_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

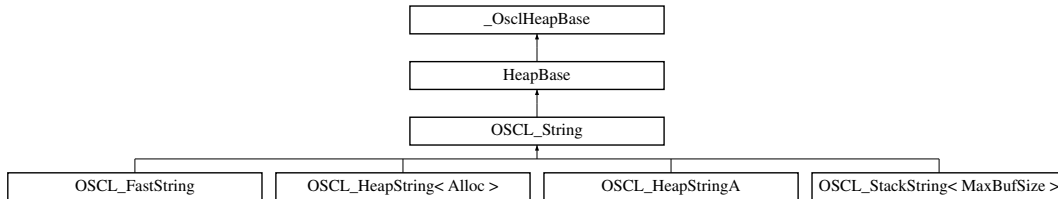
The documentation for this struct was generated from the following file:

- [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< OsciMemAllocator >](#).

### 7.73.3 Constructor & Destructor Documentation

#### 7.73.3.1 OSCL\_String::OSCL\_String () [protected]

#### 7.73.3.2 virtual OSCL\_String::~~OSCL\_String () [protected, virtual]

### 7.73.4 Member Function Documentation

#### 7.73.4.1 virtual void OSCL\_String::append\_rep (const OSCL\_String &src) [protected, pure virtual]

Append the input string to the current string. The string may be truncated to fit the available storage.

#### 7.73.4.2 virtual void OSCL\_String::append\_rep (const [chartype](#) \*cstr) [protected, pure virtual]

Append the input null-terminated string to the current string. The string may be truncated to fit the available storage.

#### 7.73.4.3 virtual const [chartype](#)\* OSCL\_String::get\_cstr () [pure virtual]

This function returns the C-style string for read access.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsciMemAllocator >](#).

#### 7.73.4.4 virtual uint32 OSCL\_String::get\_maxsize () [pure virtual]

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsciMemAllocator >](#).

#### 7.73.4.5 virtual uint32 OSCL\_String::get\_size () [pure virtual]

This function returns the string size not including the null-terminator.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsciMemAllocator >](#).

#### 7.73.4.6 virtual chartype\* OSCL\_String::get\_str () [pure virtual]

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsciMemAllocator >](#).

#### 7.73.4.7 virtual int8 OSCL\_String::hash () [virtual]

This function performs a hash operation on the string. If the string is not writable, the function leaves.

#### 7.73.4.8 virtual bool OSCL\_String::is\_writable () [virtual]

This function returns true if the string is writable.

#### 7.73.4.9 bool OSCL\_String::operator!= (const OSCL\_String & src) const

#### 7.73.4.10 OSCL\_String& OSCL\_String::operator+= (const chartype c)

Append operator. This operator appends the input character to this object. The string may be truncated to fit available storage.

#### 7.73.4.11 OSCL\_String& OSCL\_String::operator+= (const chartype \* cstr)

Append operator. This operator appends the input string to this object. The string may be truncated to fit available storage.

**am: null-terminated string**

#### 7.73.4.12 OSCL\_String& OSCL\_String::operator+= (const OSCL\_String & src)

Append operator. This operator appends the input string to this object. The string may be truncated to fit available storage.

**7.73.4.13** `bool OSCL_String::operator< (const OSCL_String & src) const`

**7.73.4.14** `bool OSCL_String::operator<= (const OSCL_String & src) const`

**7.73.4.15** `OSCL_String& OSCL_String::operator= (const chartype * cstr)`

Assignment operator

**am:** null-terminated string

Reimplemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsciMemAllocator >](#).

**7.73.4.16** `OSCL_String& OSCL_String::operator= (const OSCL_String & src)`

Assignment operator

Reimplemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), and [OSCL\\_HeapString< OsciMemAllocator >](#).

**7.73.4.17** `bool OSCL_String::operator== (const chartype * cstr) const`

Comparison operator

**am:** null-terminated string

**7.73.4.18** `bool OSCL_String::operator== (const OSCL_String & src) const`

Comparison operators

**7.73.4.19** `bool OSCL_String::operator> (const OSCL_String & src) const`

**7.73.4.20** `bool OSCL_String::operator>= (const OSCL_String & src) const`

**7.73.4.21** `]`

[char](#)*type* 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, OscDefAlloc * 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.

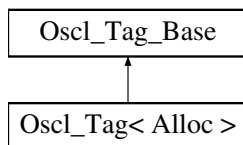
The documentation for this class was generated from the following file:

- [oscl\\_string.h](#)

## 7.74 Osl\_Tag< Alloc > Struct Template Reference

```
#include <osl_tagtree.h>
```

Inheritance diagram for Osl\_Tag< Alloc >::



### Public Methods

- [Osl\\_Tag](#) (const Osl\_Tag< Alloc > &t)
- [Osl\\_Tag](#) (const [tag\\_base\\_type](#) &t)
- [~Osl\\_Tag](#) ()
- [bool operator<](#) (const Osl\_Tag< Alloc > &x) const

### Data Fields

- [Osl\\_TAlloc](#)< [tag\\_base\\_unit](#), Alloc > [tagAllocator](#)
- [tag\\_base\\_type](#) tag

```
template<class Alloc> struct Osl_Tag< Alloc >
```

### 7.74.1 Constructor & Destructor Documentation

**7.74.1.1** `template<class Alloc> Osl_Tag< Alloc >::Osl_Tag (const Osl_Tag< Alloc > & t)`  
[inline]

**7.74.1.2** `template<class Alloc> Osl_Tag< Alloc >::Osl_Tag (const tag\_base\_type & t)`  
[inline]

**7.74.1.3** `template<class Alloc> Osl_Tag< Alloc >::~~Osl_Tag ()` [inline]

### 7.74.2 Member Function Documentation

**7.74.2.1** `template<class Alloc> bool Osl_Tag< Alloc >::operator< (const Osl_Tag< Alloc > & x) const` [inline]

### 7.74.3 Field Documentation

**7.74.3.1** `template<class Alloc> tag\_base\_type Osl_Tag< Alloc >::tag`

**7.74.3.2** `template<class Alloc> Osl\_TAlloc<tag\_base\_unit, Alloc> Osl_Tag< Alloc >::tagAllocator`

The documentation for this struct was generated from the following file:

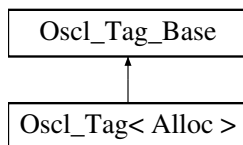
- [osci\\_tagtree.h](#)



## 7.75 Osl\_Tag\_Base Struct Reference

```
#include <osl_tagtree.h>
```

Inheritance diagram for Osl\_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 Osl_Tag_Base::size_type`

7.75.1.2 `typedef tag\_base\_unit* Osl_Tag_Base::tag_base_type`

7.75.1.3 `typedef char Osl_Tag_Base::tag_base_unit`

## 7.75.2 Member Function Documentation

7.75.2.1 `bool Osl_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 Osl_Tag_Base::tag_ancestor (tag\_base\_type & dest, const tag\_base\_type & src) const`

7.75.2.3 `int32 Osl_Tag_Base::tag_cmp (const tag\_base\_type & x, const tag\_base\_type & y) const`  
[inline]

7.75.2.4 `tag\_base\_type Osl_Tag_Base::tag_copy (tag\_base\_type & dest, const tag\_base\_type & src) const` [inline]

7.75.2.5 `OSCL_IMPORT_REF size\_type Osl_Tag_Base::tag_depth (const tag\_base\_type & t) const`

7.75.2.6 `size\_type Osl_Tag_Base::tag_len (const tag\_base\_type & t) const` [inline]

The documentation for this struct was generated from the following file:

- [oscl\\_tagtree.h](#)

## 7.76 Oslc\_TagTree< T, Alloc > Class Template Reference

```
#include <osl_tagtree.h>
```

### Public Types

- typedef [Oslc\\_Tag](#)< Alloc > [tag\\_type](#)
- typedef tag\_type::tag\_base\_type [tag\\_base\\_type](#)
- typedef [Oslc\\_Vector](#)< Node \*, Alloc > [children\\_type](#)
- typedef [Node](#) [node\\_type](#)
- typedef [node\\_type](#) \* [node\\_ptr](#)
- typedef [Oslc\\_Map](#)< const [tag\\_base\\_type](#), [node\\_ptr](#), Alloc, [Oslc\\_Tag\\_Base](#) > [map\\_type](#)
- typedef map\_type::size\_type [size\\_type](#)
- typedef map\_type::value\_type [value\\_type](#)
- typedef [Oslc\\_Pair](#)< [iterator](#), bool > [pair\\_iterator\\_bool](#)

### Public Methods

- [Oslc\\_TagTree](#) ([size\\_type](#) max\_depth=0)
- [Oslc\\_TagTree](#) (const [Oslc\\_TagTree](#)< T, Alloc > &x)
- [Oslc\\_TagTree](#)< T, Alloc > & [operator=](#) (const [Oslc\\_TagTree](#)< T, Alloc > &x)
- [~Oslc\\_TagTree](#) ()
- [iterator](#) [begin](#) ()
- [const\\_iterator](#) [begin](#) () const
- [iterator](#) [end](#) ()
- [const\\_iterator](#) [end](#) () const
- bool [empty](#) () const
- [size\\_type](#) [size](#) () const
- T & [operator\[\]](#) (const [tag\\_base\\_type](#) &t)
- [pair\\_iterator\\_bool](#) [insert](#) (const [tag\\_base\\_type](#) &t, const T &x)
- void [erase](#) ([iterator](#) position)
- [size\\_type](#) [erase](#) (const [tag\\_base\\_type](#) &x)
- void [clear](#) ()
- [iterator](#) [find](#) (const [tag\\_base\\_type](#) &x)
- [size\\_type](#) [count](#) (const [tag\\_base\\_type](#) &x) const

### 7.76.1 Detailed Description

```
template<class T, class Alloc> class Oslc_TagTree< T, Alloc >
```

Oslc\_TagTree Class.

## 7.76.2 Member Typedef Documentation

- 7.76.2.1** `template<class T, class Alloc> typedef Osl\_Vector<Node\*, Alloc> Osl_TagTree< T, Alloc >::children_type`
- 7.76.2.2** `template<class T, class Alloc> typedef Osl\_Map<const tag\_base\_type, node\_ptr, Alloc , Osl\_Tag\_Base> Osl_TagTree< T, Alloc >::map_type`
- 7.76.2.3** `template<class T, class Alloc> typedef node\_type* Osl_TagTree< T, Alloc >::node_ptr`
- 7.76.2.4** `template<class T, class Alloc> typedef Node Osl_TagTree< T, Alloc >::node_type`
- 7.76.2.5** `template<class T, class Alloc> typedef Osl\_Pair<iterator, bool> Osl_TagTree< T, Alloc >::pair_iterator_bool`
- 7.76.2.6** `template<class T, class Alloc> typedef map_type::size_type Osl_TagTree< T, Alloc >::size_type`
- 7.76.2.7** `template<class T, class Alloc> typedef tag_type::tag_base_type Osl_TagTree< T, Alloc >::tag_base_type`
- 7.76.2.8** `template<class T, class Alloc> typedef Osl\_Tag<Alloc> Osl_TagTree< T, Alloc >::tag_type`
- 7.76.2.9** `template<class T, class Alloc> typedef map_type::value_type Osl_TagTree< T, Alloc >::value_type`

## 7.76.3 Constructor & Destructor Documentation

- 7.76.3.1** `template<class T, class Alloc> Osl_TagTree< T, Alloc >::Osl_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> Osl_TagTree< T, Alloc >::Osl_TagTree (const Osl_TagTree< T, Alloc > & x) [inline]`

Copy constructor

- 7.76.3.3** `template<class T, class Alloc> Osl_TagTree< T, Alloc >::~~Osl_TagTree () [inline]`

Destructor

## 7.76.4 Member Function Documentation

- 7.76.4.1** `template<class T, class Alloc> const\_iterator Osl_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 Osl_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 Osl_TagTree< T, Alloc >::clear () [inline]`

Erases the entire tag tree.

**7.76.4.4** `template<class T, class Alloc> size_type Osl_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 Osl_TagTree< T, Alloc >::empty () const [inline]`

Returns true if tree size is 0

**7.76.4.6** `template<class T, class Alloc> const_iterator Osl_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 Osl_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 Osl_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 Osl_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 Osl_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 Osl_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> Osl_TagTree<T, Alloc>& Osl_TagTree< T, Alloc >::operator= (const Osl_TagTree< T, Alloc > & x) [inline]`

Assignment operator

**7.76.4.13** `]`

`template<class T, class Alloc> T& Osl_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 Osl_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 Osl\_TagTree< T, Alloc >::const\_iterator Struct Reference

```
#include <osl_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 Osl_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 Osl\_TagTree< T, Alloc >::const_iterator::mapiter`

7.77.1.2 `template<class T, class Alloc> typedef const node\_type* Osl\_TagTree< T, Alloc >::const_iterator::pointer`

7.77.1.3 `template<class T, class Alloc> typedef const node\_type& Osl\_TagTree< T, Alloc >::const_iterator::reference`

7.77.1.4 `template<class T, class Alloc> typedef const_iterator Osl\_TagTree< T, Alloc >::const_iterator::self`

### 7.77.2 Constructor & Destructor Documentation

7.77.2.1 `template<class T, class Alloc> Osl\_TagTree< T, Alloc >::const_iterator::const_iterator () [inline]`

7.77.2.2 `template<class T, class Alloc> Osl\_TagTree< T, Alloc >::const_iterator::const_iterator (mapiter x) [inline]`

7.77.2.3 `template<class T, class Alloc> Osl\_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 Osl\_TagTree< T, Alloc >::const_iterator::operator * () const [inline]`

7.77.3.2 `template<class T, class Alloc> bool Osl\_TagTree< T, Alloc >::const_iterator::operator!= (const self & x) [inline]`

7.77.3.3 `template<class T, class Alloc> self Osl\_TagTree< T, Alloc >::const_iterator::operator++ (int) [inline]`

7.77.3.4 `template<class T, class Alloc> self& Osl\_TagTree< T, Alloc >::const_iterator::operator++ () [inline]`

7.77.3.5 `template<class T, class Alloc> self Osl\_TagTree< T, Alloc >::const_iterator::operator- (int) [inline]`

7.77.3.6 `template<class T, class Alloc> self& Osl\_TagTree< T, Alloc >::const_iterator::operator- () [inline]`

7.77.3.7 `template<class T, class Alloc> pointer Osl\_TagTree< T, Alloc >::const_iterator::operator → () const [inline]`

7.77.3.8 `template<class T, class Alloc> bool Osl\_TagTree< T, Alloc >::const_iterator::operator== (const self & x) [inline]`

### 7.77.4 Field Documentation

7.77.4.1 `template<class T, class Alloc> manifer Osl\_TagTree< T, Alloc >::const_iterator::manit`



- [oscl\\_tagtree.h](#)

## 7.78 Osci\_TagTree< T, Alloc >::iterator Struct Reference

```
#include <osci_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 Osci_TagTree< T, Alloc >::iterator
```

## 7.78.1 Member Typedef Documentation

**7.78.1.1** `template<class T, class Alloc> typedef map\_type::iterator Osci\_TagTree< T, Alloc >::iterator::mapiter`

**7.78.1.2** `template<class T, class Alloc> typedef node\_type\* Osci\_TagTree< T, Alloc >::iterator::pointer`

**7.78.1.3** `template<class T, class Alloc> typedef node\_type& Osci\_TagTree< T, Alloc >::iterator::reference`

**7.78.1.4** `template<class T, class Alloc> typedef iterator Osci\_TagTree< T, Alloc >::iterator::self`

## 7.78.2 Constructor & Destructor Documentation

**7.78.2.1** `template<class T, class Alloc> Osci\_TagTree< T, Alloc >::iterator::iterator ()`  
[inline]

**7.78.2.2** `template<class T, class Alloc> Osci\_TagTree< T, Alloc >::iterator::iterator (mapiter x)`  
[inline]

**7.78.2.3** `template<class T, class Alloc> Osci\_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 Osci\_TagTree< T, Alloc >::iterator::operator * () const` [inline]

**7.78.3.2** `template<class T, class Alloc> bool Osci\_TagTree< T, Alloc >::iterator::operator!= (const self &x)` [inline]

**7.78.3.3** `template<class T, class Alloc> self Osci\_TagTree< T, Alloc >::iterator::operator++ (int)`  
[inline]

**7.78.3.4** `template<class T, class Alloc> self& Osci\_TagTree< T, Alloc >::iterator::operator++ ()`  
[inline]

**7.78.3.5** `template<class T, class Alloc> self Osci\_TagTree< T, Alloc >::iterator::operator- (int)`  
[inline]

**7.78.3.6** `template<class T, class Alloc> self& Osci\_TagTree< T, Alloc >::iterator::operator- ()`  
[inline]

**7.78.3.7** `template<class T, class Alloc> pointer Osci\_TagTree< T, Alloc >::iterator::operator → () const` [inline]

**7.78.3.8** `template<class T, class Alloc> bool Osci\_TagTree< T, Alloc >::iterator::operator== (const self &x)` [inline]

## 7.78.4 Field Documentation

**7.78.4.1** `template<class T, class Alloc> mapiter Osci\_TagTree< T, Alloc >::iterator::mapit`

- [oscl\\_tagtree.h](#)

## 7.79 Osci\_TagTree< T, Alloc >::Node Struct Reference

```
#include <osci_tagtree.h>
```

### Public Types

- typedef [Osci\\_Vector](#)< Node \*, Alloc > [children\\_type](#)

### Public Methods

- [Node](#) ()
- void [sort\\_children](#) ()
- [tag\\_type::size\\_type](#) [depth](#) ()

### Data Fields

- [tag\\_type](#) [tag](#)
- T [value](#)
- Node \* [parent](#)
- [children\\_type](#) [children](#)

```
template<class T, class Alloc> struct Osci_TagTree< T, Alloc >::Node
```

### 7.79.1 Member Typedef Documentation

**7.79.1.1** `template<class T, class Alloc> typedef Osci\_Vector<Node*, Alloc> Osci\_TagTree< T, Alloc >::Node::children_type`

### 7.79.2 Constructor & Destructor Documentation

**7.79.2.1** `template<class T, class Alloc> Osci\_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 Osci\_TagTree< T, Alloc >::Node::depth () [inline]`

**7.79.3.2** `template<class T, class Alloc> void Osci\_TagTree< T, Alloc >::Node::sort_children () [inline]`

### 7.79.4 Field Documentation

**7.79.4.1** `template<class T, class Alloc> children\_type Osci\_TagTree< T, Alloc >::Node::children`

**7.79.4.2** `template<class T, class Alloc> Node* Osci\_TagTree< T, Alloc >::Node::parent`

**7.79.4.3** `template<class T, class Alloc> tag\_type Osci\_TagTree< T, Alloc >::Node::tag`

**7.79.4.4** `template<class T, class Alloc> T Osci\_TagTree< T, Alloc >::Node::value`

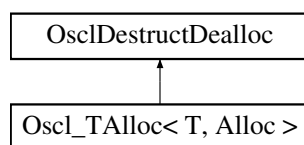
The documentation for this struct was generated from the following file:

- [osci\\_tagtree.h](#)

## 7.80 Osl\_TAlloc< T, Alloc > Class Template Reference

```
#include <osl_defalloc.h>
```

Inheritance diagram for Osl\_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 [~Osl\\_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](#) ([OslAny](#) \*p)
- void [deallocate](#) ([OslAny](#) \*p, [size\\_type](#) n)
- void [destruct\\_and\\_dealloc](#) ([OslAny](#) \*p)
- [pointer address](#) ([reference](#) r)
- [const\\_pointer address](#) ([const\\_reference](#) r) const
- void [construct](#) ([pointer](#) p, [const\\_reference](#) val)
- void [destroy](#) ([pointer](#) p)

```
template<class T, class Alloc> class Osci_TAlloc< T, Alloc >
```

## 7.80.1 Member Typedef Documentation

7.80.1.1 `template<class T, class Alloc> typedef const T* Osci_TAlloc< T, Alloc >::const_pointer`

7.80.1.2 `template<class T, class Alloc> typedef const T& Osci_TAlloc< T, Alloc >::const_reference`

7.80.1.3 `template<class T, class Alloc> typedef T* Osci_TAlloc< T, Alloc >::pointer`

7.80.1.4 `template<class T, class Alloc> typedef T& Osci_TAlloc< T, Alloc >::reference`

7.80.1.5 `template<class T, class Alloc> typedef uint32 Osci_TAlloc< T, Alloc >::size_type`

7.80.1.6 `template<class T, class Alloc> typedef T Osci_TAlloc< T, Alloc >::value_type`

## 7.80.2 Constructor & Destructor Documentation

7.80.2.1 `template<class T, class Alloc> virtual Osci_TAlloc< T, Alloc >::~~Osci_TAlloc ()`  
[inline, virtual]

## 7.80.3 Member Function Documentation

7.80.3.1 `template<class T, class Alloc> const_pointer Osci_TAlloc< T, Alloc >::address`  
(const\_reference r) const [inline]

7.80.3.2 `template<class T, class Alloc> pointer Osci_TAlloc< T, Alloc >::address (reference r)`  
[inline]

7.80.3.3 `template<class T, class Alloc> pointer Osci_TAlloc< T, Alloc >::alloc_and_construct`  
(const\_reference val) [inline]

7.80.3.4 `template<class T, class Alloc> pointer Osci_TAlloc< T, Alloc >::alloc_and_construct_fl`  
(const\_reference val, const char \*file\_name, const int line\_num) [inline]

7.80.3.5 `template<class T, class Alloc> pointer Osci_TAlloc< T, Alloc >::allocate (uint32 size)`  
[inline]

7.80.3.6 `template<class T, class Alloc> pointer Osci_TAlloc< T, Alloc >::allocate_fl (uint32 size,`  
const char \*file\_name, const int line\_num) [inline]

7.80.3.7 `template<class T, class Alloc> void Osci_TAlloc< T, Alloc >::construct (pointer p,`  
const\_reference val) [inline]

7.80.3.8 `template<class T, class Alloc> void Osci_TAlloc< T, Alloc >::deallocate (OsciAny * p,`  
size\_type n) [inline]

7.80.3.9 `template<class T, class Alloc> void Osci_TAlloc< T, Alloc >::deallocate (OsciAny * p)`  
[inline]

7.80.3.10 `template<class T, class Alloc> void Osci_TAlloc< T, Alloc >::destroy (pointer p)`  
[inline]

7.80.3.11 `template<class T, class Alloc> void Osci_TAlloc< T, Alloc >::destruct_and_dealloc`  
(OsciAny \* p) [inline, virtual]



The documentation for this class was generated from the following file:

- [oscl\\_defalloc.h](#)

## 7.81 `OscI_TAlloc< T, Alloc >::rebind< U, V >` Struct Template Reference

```
#include <oscl_defalloc.h>
```

### Public Types

- typedef `OscI_TAlloc< U, V >` `other`

```
template<class T, class Alloc>template<class U, class V> struct OscI_TAlloc< T, Alloc >::rebind< U, V >
```

### 7.81.1 Member Typedef Documentation

**7.81.1.1** `template<class T, class Alloc> template<class U, class V> typedef OscI_TAlloc<U, V> OscI_TAlloc< T, Alloc >::rebind< U, V >::other`

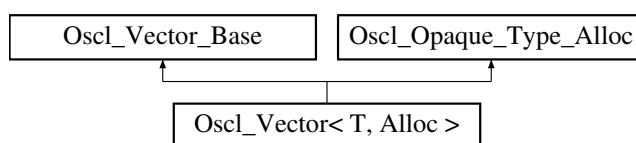
The documentation for this struct was generated from the following file:

- `oscl_defalloc.h`

## 7.82 Osl\_Vector< T, Alloc > Class Template Reference

```
#include <osl_vector.h>
```

Inheritance diagram for Osl\_Vector< T, Alloc >::



### Public Types

- typedef T [value\\_type](#)
- typedef T \* [pointer](#)
- typedef T & [reference](#)
- typedef const T & [const\\_reference](#)
- typedef T \* [iterator](#)
- typedef const T \* [const\\_iterator](#)

### Public Methods

- [Osl\\_Vector](#) ()
- [Osl\\_Vector](#) (uint32 n)
- [Osl\\_Vector](#) (const Osl\_Vector< T, Alloc > &x)
- virtual [~Osl\\_Vector](#) ()
- Osl\_Vector< T, Alloc > & [operator=](#) (const Osl\_Vector< T, Alloc > &x)
- void [push\\_back](#) (const T &x)
- void [push\\_front](#) (const T &x)
- [iterator](#) [insert](#) ([iterator](#) pos, const T &x)
- T & [operator\[\]](#) (uint32 n)
- const T & [operator\[\]](#) (uint32 n) const
- T & [front](#) ()
- const T & [front](#) () const
- T & [back](#) ()
- const T & [back](#) () const
- void [pop\\_back](#) ()
- void [clear](#) ()
- void [destroy](#) ()
- [iterator](#) [begin](#) () const
- [iterator](#) [end](#) () const
- [iterator](#) [erase](#) ([iterator](#) pos)
- [iterator](#) [erase](#) ([iterator](#) first, [iterator](#) last)

## 7.82.1 Detailed Description

**template<class T, class Alloc> class Osl\_Vector< T, Alloc >**

Osl\_Vector Class. A subset of STL::Vector methods. Osl\_Vector supports random access to elements, constant time insertion and removal of elements at the end of the vector, and linear time insertion and removal of elements at the beginning or middle of the vector. The number of elements in a vector can vary dynamically, and memory management is performed automatically.

## 7.82.2 Member Typedef Documentation

**7.82.2.1 template<class T, class Alloc> typedef const T\* Osl\_Vector< T, Alloc >::const\_iterator**

**7.82.2.2 template<class T, class Alloc> typedef const T& Osl\_Vector< T, Alloc >::const\_reference**

**7.82.2.3 template<class T, class Alloc> typedef T\* Osl\_Vector< T, Alloc >::iterator**

**7.82.2.4 template<class T, class Alloc> typedef T\* Osl\_Vector< T, Alloc >::pointer**

**7.82.2.5 template<class T, class Alloc> typedef T& Osl\_Vector< T, Alloc >::reference**

**7.82.2.6 template<class T, class Alloc> typedef T Osl\_Vector< T, Alloc >::value\_type**

## 7.82.3 Constructor & Destructor Documentation

**7.82.3.1 template<class T, class Alloc> Osl\_Vector< T, Alloc >::Osl\_Vector () [inline]**

Creates an empty vector.

**7.82.3.2 template<class T, class Alloc> Osl\_Vector< T, Alloc >::Osl\_Vector (uint32 n) [inline]**

Creates an empty vector with capacity n.

### Parameters:

**n** creates a vector with n elements. The main reason for specifying n is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

**7.82.3.3 template<class T, class Alloc> Osl\_Vector< T, Alloc >::Osl\_Vector (const Osl\_Vector< T, Alloc > &x) [inline]**

Copy Constructor.

### Parameters:

**x** vector class to copy.

**7.82.3.4** `template<class T, class Alloc> virtual Osl_Vector< T, Alloc >::~~Osl_Vector ()`  
`[inline, virtual]`

The destructor.

## 7.82.4 Member Function Documentation

**7.82.4.1** `template<class T, class Alloc> const T& Osl_Vector< T, Alloc >::back () const`  
`[inline]`

Returns the last element.

**7.82.4.2** `template<class T, class Alloc> T& Osl_Vector< T, Alloc >::back ()` `[inline]`

Returns the last element.

**7.82.4.3** `template<class T, class Alloc> iterator Osl_Vector< T, Alloc >::begin () const`  
`[inline]`

Returns an iterator pointing to the beginning of the vector.

Reimplemented from [Osl\\_Vector\\_Base](#).

**7.82.4.4** `template<class T, class Alloc> void Osl_Vector< T, Alloc >::clear ()` `[inline]`

Removes all elements.

**7.82.4.5** `template<class T, class Alloc> void Osl_Vector< T, Alloc >::destroy ()` `[inline]`

Destroy – this is like an explicit destructor call.

Reimplemented from [Osl\\_Vector\\_Base](#).

**7.82.4.6** `template<class T, class Alloc> iterator Osl_Vector< T, Alloc >::end () const`  
`[inline]`

Returns an iterator pointing to the end of the vector..

Reimplemented from [Osl\\_Vector\\_Base](#).

**7.82.4.7** `template<class T, class Alloc> iterator Osl_Vector< T, Alloc >::erase (iterator first, iterator last)` `[inline]`

Erases elements in range [first, last). Erasing an element invalidates all iterators pointing to elements following the deletion point.

### Parameters:

*first* starting position

*last* ending position, this position is not erased

**7.82.4.8** `template<class T, class Alloc> iterator OscVector< 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& OscVector< T, Alloc >::front () const`  
`[inline]`

Returns the first element.

**7.82.4.10** `template<class T, class Alloc> T& OscVector< T, Alloc >::front ()` `[inline]`

Returns the first element.

**7.82.4.11** `template<class T, class Alloc> iterator OscVector< 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> OscVector<T, Alloc>& OscVector< T, Alloc >::operator= (const OscVector< T, Alloc > & x)` `[inline]`

The assignment operator

**7.82.4.13** `]`

`template<class T, class Alloc> const T& OscVector< 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& OscVector< T, Alloc >::operator[] (uint32 n)` `[inline]`

Returns the *n*'th element.

**Parameters:**

*n* element position to return

**7.82.4.15** `template<class T, class Alloc> void OscVector< T, Alloc >::pop_back ()` [inline]

Removes the last element.

Reimplemented from [OscVectorBase](#).

**7.82.4.16** `template<class T, class Alloc> void OscVector< T, Alloc >::push_back (const T & x)`  
[inline]

Inserts a new element at the end. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* new element

**7.82.4.17** `template<class T, class Alloc> void OscVector< T, Alloc >::push_front (const T & x)`  
[inline]

Inserts a new element at the front. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* new element

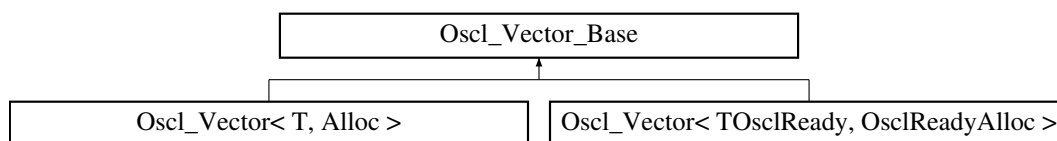
The documentation for this class was generated from the following file:

- [oscl\\_vector.h](#)

## 7.83 Osl\_Vector\_Base Class Reference

```
#include <osl_vector.h>
```

Inheritance diagram for Osl\_Vector\_Base::



### Public Methods

- uint32 [size](#) () const
- uint32 [capacity](#) () const
- bool [empty](#) () const
- OSCL\_IMPORT\_REF void [reserve](#) (uint32 n)

### Protected Methods

- OSCL\_IMPORT\_REF void [construct](#) (Osl\_Opaque\_Type\_Alloc \*aType)
- OSCL\_IMPORT\_REF void [construct](#) (Osl\_Opaque\_Type\_Alloc \*aType, uint32 n)
- OSCL\_IMPORT\_REF void [construct](#) (Osl\_Opaque\_Type\_Alloc \*aType, const Osl\_Vector\_Base &x)
- virtual [~Osl\\_Vector\\_Base](#) ()
- OSCL\_IMPORT\_REF void [push\\_back](#) (const OslAny \*x)
- OSCL\_IMPORT\_REF void [pop\\_back](#) ()
- OSCL\_IMPORT\_REF void [push\\_front](#) (const OslAny \*x)
- OSCL\_IMPORT\_REF OslAny \* [insert](#) (OslAny \*pos, const OslAny \*x)
- OSCL\_IMPORT\_REF OslAny \* [erase](#) (OslAny \*pos)
- OSCL\_IMPORT\_REF OslAny \* [erase](#) (OslAny \*first, OslAny \*last)
- OSCL\_IMPORT\_REF void [assign\\_vector](#) (const Osl\_Vector\_Base &x)
- OSCL\_IMPORT\_REF void [destroy](#) ()

### Protected Attributes

- uint32 [numelems](#)
- uint32 [bufsize](#)
- OslAny \* [elems](#)
- uint32 [sizeof\\_T](#)

### Friends

- class [OslPriorityQueueBase](#)



### 7.83.1 Detailed Description

Osl\_Vector\_Base is a non-templated base class for [Osl\\_Vector](#). The purpose of this base class is to avoid large inline routines in the [Osl\\_Vector](#) implementation. This class is not intended for direct instantiation except by [Osl\\_Vector](#).

### 7.83.2 Constructor & Destructor Documentation

**7.83.2.1** `virtual Osl_Vector_Base::~~Osl_Vector_Base () [inline, protected, virtual]`

The destructor.

### 7.83.3 Member Function Documentation

**7.83.3.1** `OSCL_IMPORT_REF void Osl_Vector_Base::assign_vector (const Osl_Vector_Base & x) [protected]`

**7.83.3.2** `uint32 Osl_Vector_Base::capacity () const [inline]`

Returns the allocated memory of the vector in units of number of elements.

**7.83.3.3** `OSCL_IMPORT_REF void Osl_Vector_Base::construct (Osl_Opaque_Type_Alloc * aType, const Osl_Vector_Base & x) [protected]`

**7.83.3.4** `OSCL_IMPORT_REF void Osl_Vector_Base::construct (Osl_Opaque_Type_Alloc * aType, uint32 n) [protected]`

**7.83.3.5** `OSCL_IMPORT_REF void Osl_Vector_Base::construct (Osl_Opaque_Type_Alloc * aType) [protected]`

**7.83.3.6** `OSCL_IMPORT_REF void Osl_Vector_Base::destroy () [protected]`

Reimplemented in [Osl\\_Vector< T, Alloc >](#), [Osl\\_Vector< OslComponentRegistryElement, OslMemAllocator >](#), [Osl\\_Vector< uint32, OslMemAllocator >](#), [Osl\\_Vector< OslSocketServRequestQElem, OslMemAllocator >](#), [Osl\\_Vector< Node \\*, Alloc >](#), [Osl\\_Vector< OslFixedCacheParam, OslMemAllocator >](#), [Osl\\_Vector< OslSocketRequest \\*, OslMemAllocator >](#), [Osl\\_Vector< entry\\_type \\*, Alloc >](#), [Osl\\_Vector< OSCL\\_HeapString< OslMemAllocator >, OslMemAllocator >](#), [Osl\\_Vector< OslAsyncFileBuffer \\*, OslMemAllocator >](#), [Osl\\_Vector< TOslFileOffset, OslMemAllocator >](#), [Osl\\_Vector< MemPoolBufferInfo \\*, OslMemAllocator >](#), [Osl\\_Vector< OslSharedPtr< PVLoggerFilter >, alloc\\_type >](#), [Osl\\_Vector< TOslReady, OslReadyAlloc >](#), [Osl\\_Vector< OslFileCacheBuffer, OslMemAllocator >](#), [Osl\\_Vector< OslSharedPtr< PVLoggerAppender >, alloc\\_type >](#), [Osl\\_Vector< OslAny \\*, OslMemAllocator >](#), and [Osl\\_Vector< OslNetworkAddress, OslMemAllocator >](#).

**7.83.3.7** `bool Osl_Vector_Base::empty () const [inline]`

True if the vector's size is 0.

### 7.83.3.8 OSCL\_IMPORT\_REF OslAny\* Osl\_Vector\_Base::erase (OslAny \* *first*, OslAny \* *last*) [protected]

Erases elements in range [*first*, *last*). Erasing an element invalidates all iterators pointing to elements following the deletion point.

#### Parameters:

*first* starting position

*last* ending position, this position is not erased

### 7.83.3.9 OSCL\_IMPORT\_REF OslAny\* Osl\_Vector\_Base::erase (OslAny \* *pos*) [protected]

Erases the element pointed to by iterator *pos*. Erasing an element invalidates all iterators pointing to elements following the deletion point.

#### Parameters:

*pos* iterator at erase position

### 7.83.3.10 OSCL\_IMPORT\_REF OslAny\* Osl\_Vector\_Base::insert (OslAny \* *pos*, const OslAny \* *x*) [protected]

Inserts a new element at a specific position.

#### Parameters:

*pos* iterator at insert position.

*x* pointer to new element

### 7.83.3.11 OSCL\_IMPORT\_REF void Osl\_Vector\_Base::pop\_back () [protected]

Removes the last element.

Reimplemented in [Osl\\_Vector< T, Alloc >](#), [Osl\\_Vector< OslComponentRegistryElement, OslMemAllocator >](#), [Osl\\_Vector< uint32, OslMemAllocator >](#), [Osl\\_Vector< OslSocketServRequestQElem, OslMemAllocator >](#), [Osl\\_Vector< Node \\*, Alloc >](#), [Osl\\_Vector< OslFixedCacheParam, OslMemAllocator >](#), [Osl\\_Vector< OslSocketRequest \\*, OslMemAllocator >](#), [Osl\\_Vector< entry\\_type \\*, Alloc >](#), [Osl\\_Vector< OSCL\\_HeapString< OslMemAllocator >, OslMemAllocator >](#), [Osl\\_Vector< OslAsyncFileBuffer \\*, OslMemAllocator >](#), [Osl\\_Vector< TOslFileOffset, OslMemAllocator >](#), [Osl\\_Vector< MemPoolBufferInfo \\*, OslMemAllocator >](#), [Osl\\_Vector< OslSharedPtr< PVLoggerFilter >, alloc\\_type >](#), [Osl\\_Vector< TOslReady, OslReadyAlloc >](#), [Osl\\_Vector< OslFileCacheBuffer, OslMemAllocator >](#), [Osl\\_Vector< OslSharedPtr< PVLoggerAppender >, alloc\\_type >](#), [Osl\\_Vector< OslAny \\*, OslMemAllocator >](#), and [Osl\\_Vector< OslNetworkAddress, OslMemAllocator >](#).

### 7.83.3.12 OSCL\_IMPORT\_REF void Osl\_Vector\_Base::push\_back (const OslAny \* *x*) [protected]

Inserts a new element at the end. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* pointer to the new element

**7.83.3.13** `OSCL_IMPORT_REF void Osl_Vector_Base::push_front (const OslAny * x)`  
[protected]

Inserts a new element at the front. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

**Parameters:**

*x* pointer to new element

**7.83.3.14** `OSCL_IMPORT_REF void Osl_Vector_Base::reserve (uint32 n)`

Reallocates memory if necessary to a capacity of *n* elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

**Parameters:**

*n* size of vector

**7.83.3.15** `uint32 Osl_Vector_Base::size () const` [inline]

Returns the size of the vector in units of number of elements.

## 7.83.4 Friends And Related Function Documentation

**7.83.4.1** `friend class OslPriorityQueueBase` [friend]

## 7.83.5 Field Documentation

**7.83.5.1** `uint32 Osl_Vector_Base::bufsize` [protected]

**7.83.5.2** `OslAny* Osl_Vector_Base::elems` [protected]

**7.83.5.3** `uint32 Osl_Vector_Base::numelems` [protected]

**7.83.5.4** `uint32 Osl_Vector_Base::sizeof_T` [protected]

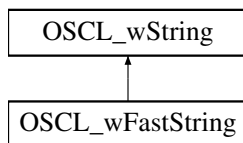
The documentation for this class was generated from the following file:

- [oscl\\_vector.h](#)

## 7.84 OSCL\_wFastString Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_wFastString::



### Public Types

- typedef OSCL\_wString::chartype [chartype](#)
- typedef [TOSCL\\_wStringOp](#) optype
- typedef [OSCL\\_String::chartype](#) other\_chartype

### Public Methods

- OSCL\_IMPORT\_REF [OSCL\\_wFastString](#) ()
- OSCL\_IMPORT\_REF [OSCL\\_wFastString](#) (const OSCL\_wFastString &src)
- OSCL\_IMPORT\_REF [OSCL\\_wFastString](#) (const [chartype](#) \*cstr)
- OSCL\_IMPORT\_REF [OSCL\\_wFastString](#) ([chartype](#) \*buf, uint32 maxlen)
- OSCL\_IMPORT\_REF ~[OSCL\\_wFastString](#) ()
- OSCL\_IMPORT\_REF uint32 [get\\_size](#) () const
- OSCL\_IMPORT\_REF uint32 [get\\_maxsize](#) () const
- OSCL\_IMPORT\_REF const [chartype](#) \* [get\\_cstr](#) () const
- OSCL\_IMPORT\_REF [chartype](#) \* [get\\_str](#) () const
- OSCL\_IMPORT\_REF OSCL\_wFastString & [operator=](#) (const OSCL\_wFastString &src)
- OSCL\_IMPORT\_REF OSCL\_wFastString & [operator=](#) (const [chartype](#) \*cstr)
- OSCL\_IMPORT\_REF void [set](#) ([chartype](#) \*cstr, uint32 maxlen)
- OSCL\_IMPORT\_REF void [set](#) (const [other\\_chartype](#) \*buf, uint32 numofbyte, [optype](#) op)
- OSCL\_IMPORT\_REF void [set\\_length](#) ()

### Friends

- class [OSCL\\_wString](#)

### 7.84.1 Detailed Description

OSCL\_wFastString is identical to [OSCL\\_FastString](#) except that it uses wide-character format. For descriptions, see [OSCL\\_FastString](#).

### 7.84.2 Member Typedef Documentation

#### 7.84.2.1 typedef OSCL\_wString::chartype OSCL\_wFastString::chartype

Reimplemented from [OSCL\\_wString](#).

7.84.2.2 typedef **OSCL\_wStringOp** OSCL\_wFastString::optype

7.84.2.3 typedef **OSCL\_String::chartype** OSCL\_wFastString::other\_chartype

### 7.84.3 Constructor & Destructor Documentation

7.84.3.1 OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString ()

7.84.3.2 OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString (const OSCL\_wFastString & *src*)

7.84.3.3 OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString (const **chartype** \* *cstring*)

7.84.3.4 OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString (**chartype** \* *buf*, uint32 *maxlen*)

7.84.3.5 OSCL\_IMPORT\_REF OSCL\_wFastString::~~OSCL\_wFastString ()

### 7.84.4 Member Function Documentation

7.84.4.1 OSCL\_IMPORT\_REF const **chartype**\* OSCL\_wFastString::get\_cstr () [virtual]

Implements [OSCL\\_wString](#).

7.84.4.2 OSCL\_IMPORT\_REF uint32 OSCL\_wFastString::get\_maxsize () [virtual]

Implements [OSCL\\_wString](#).

7.84.4.3 OSCL\_IMPORT\_REF uint32 OSCL\_wFastString::get\_size () [virtual]

Implements [OSCL\\_wString](#).

7.84.4.4 OSCL\_IMPORT\_REF **chartype**\* OSCL\_wFastString::get\_str () [virtual]

Implements [OSCL\\_wString](#).

7.84.4.5 OSCL\_IMPORT\_REF OSCL\_wFastString& OSCL\_wFastString::operator= (const **chartype** \* *cstring*)

Reimplemented from [OSCL\\_wString](#).

- 7.84.4.6** OSCL\_IMPORT\_REF OSCL\_wFastString& OSCL\_wFastString::operator= (const OSCL\_wFastString & *src*)
- 7.84.4.7** OSCL\_IMPORT\_REF void OSCL\_wFastString::set (const [other\\_chartype](#) \* *buf*, uint32 *numofbyte*, [optype](#) *op*)
- 7.84.4.8** OSCL\_IMPORT\_REF void OSCL\_wFastString::set ([chartype](#) \* *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]

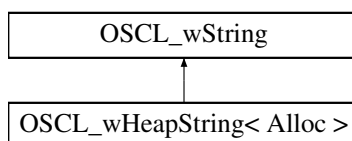
The documentation for this class was generated from the following file:

- [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](#) optype
- typedef [OSCL\\_String::chartype](#) other\_chartype

### Public Methods

- [OSCL\\_wHeapString](#) ()
- [OSCL\\_wHeapString](#) (const OSCL\_wHeapString &src)
- [OSCL\\_wHeapString](#) (const [OSCL\\_wString](#) &src)
- [OSCL\\_wHeapString](#) (const [chartype](#) \*cstr)
- [OSCL\\_wHeapString](#) (const [chartype](#) \*buf, uint32 length)
- [~OSCL\\_wHeapString](#) ()
- uint32 [get\\_size](#) () const
- uint32 [get\\_maxsize](#) () const
- const [chartype](#) \* [get\\_cstr](#) () const
- [chartype](#) \* [get\\_str](#) () const
- [OSCL\\_wHeapString](#) & [operator=](#) (const OSCL\_wHeapString &src)
- [OSCL\\_wHeapString](#) & [operator=](#) (const [OSCL\\_wString](#) &src)
- [OSCL\\_wHeapString](#) & [operator=](#) (const [chartype](#) \*cstr)
- void [set](#) (const [chartype](#) \*buf, uint32 length)
- void [set](#) (const [other\\_chartype](#) \*buf, [optype](#) op)
- void [set](#) (const [other\\_chartype](#) \*buf, uint32 length, [optype](#) op)

### Friends

- class [OSCL\\_wString](#)

### 7.85.1 Detailed Description

**template<class Alloc> class OSCL\_wHeapString< Alloc >**

OSCL\_wHeapString is identical to [OSCL\\_HeapString](#) except that it uses wide-character format. For descriptions, see [OSCL\\_HeapString](#).

## 7.85.2 Member Typedef Documentation

**7.85.2.1** `template<class Alloc> typedef OSCL_wString::chartype OSCL_wHeapString< Alloc >::chartype`

Reimplemented from [OSCL\\_wString](#).

**7.85.2.2** `template<class Alloc> typedef TOSCL\_wStringOp OSCL_wHeapString< Alloc >::optype`

**7.85.2.3** `template<class Alloc> typedef OSCL\_String::chartype OSCL_wHeapString< Alloc >::other_chartype`

## 7.85.3 Friends And Related Function Documentation

**7.85.3.1** `template<class Alloc> friend class OSCL_wString [friend]`

The documentation for this class was generated from the following file:

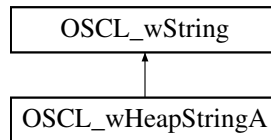
- [oscl\\_string\\_containers.h](#)



## 7.86 OSCL\_wHeapStringA Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_wHeapStringA::



### Public Types

- typedef OSCL\_wString::chartype [chartype](#)
- typedef [TOSCL\\_wStringOp](#) optype
- typedef [OSCL\\_String::chartype](#) other\_chartype

### Public Methods

- OSCL\_IMPORT\_REF [OSCL\\_wHeapStringA](#) ()
- OSCL\_IMPORT\_REF [OSCL\\_wHeapStringA](#) ([OscDefAlloc](#) \*alloc, [OscRefCount](#) \*ref=NULL)
- OSCL\_IMPORT\_REF [OSCL\\_wHeapStringA](#) (const [OSCL\\_wHeapStringA](#) &src)
- OSCL\_IMPORT\_REF [OSCL\\_wHeapStringA](#) (const [OSCL\\_wHeapStringA](#) &src, [OscDefAlloc](#) \*alloc, [OscRefCount](#) \*ref=NULL)
- OSCL\_IMPORT\_REF [OSCL\\_wHeapStringA](#) (const [OSCL\\_wString](#) &src, [OscDefAlloc](#) \*alloc=NULL, [OscRefCount](#) \*ref=NULL)
- OSCL\_IMPORT\_REF [OSCL\\_wHeapStringA](#) (const [chartype](#) \*cstr, [OscDefAlloc](#) \*alloc=NULL, [OscRefCount](#) \*ref=NULL)
- OSCL\_IMPORT\_REF [OSCL\\_wHeapStringA](#) (const [chartype](#) \*buf, uint32 length, [OscDefAlloc](#) \*alloc=NULL, [OscRefCount](#) \*ref=NULL)
- OSCL\_IMPORT\_REF [~OSCL\\_wHeapStringA](#) ()
- OSCL\_IMPORT\_REF uint32 [get\\_size](#) () const
- OSCL\_IMPORT\_REF uint32 [get\\_maxsize](#) () const
- OSCL\_IMPORT\_REF const [chartype](#) \* [get\\_cstr](#) () const
- OSCL\_IMPORT\_REF [chartype](#) \* [get\\_str](#) () const
- OSCL\_IMPORT\_REF [OSCL\\_wHeapStringA](#) & [operator=](#) (const [OSCL\\_wHeapStringA](#) &src)
- OSCL\_IMPORT\_REF [OSCL\\_wHeapStringA](#) & [operator=](#) (const [OSCL\\_wString](#) &src)
- OSCL\_IMPORT\_REF [OSCL\\_wHeapStringA](#) & [operator=](#) (const [chartype](#) \*cstr)
- OSCL\_IMPORT\_REF void [set](#) (const [chartype](#) \*buf, uint32 length)
- OSCL\_IMPORT\_REF void [set](#) (const [other\\_chartype](#) \*buf, [optype](#) op)
- OSCL\_IMPORT\_REF void [set](#) (const [other\\_chartype](#) \*buf, uint32 length, [optype](#) op)

### Friends

- class [OSCL\\_wString](#)

## 7.86.1 Detailed Description

OSCL\_wHeapStringA is identical to [OSCL\\_HeapStringA](#) except that it uses wide-character format. For descriptions, see [OSCL\\_HeapStringA](#).

## 7.86.2 Member Typedef Documentation

### 7.86.2.1 typedef OSCL\_wString::chartype OSCL\_wHeapStringA::chartype

Reimplemented from [OSCL\\_wString](#).

### 7.86.2.2 typedef [TOSCL\\_wStringOp](#) OSCL\_wHeapStringA::optype

### 7.86.2.3 typedef [OSCL\\_String::chartype](#) OSCL\_wHeapStringA::other\_chartype

## 7.86.3 Constructor & Destructor Documentation

### 7.86.3.1 OSCL\_IMPORT\_REF OSCL\_wHeapStringA::OSCL\_wHeapStringA ()

### 7.86.3.2 OSCL\_IMPORT\_REF OSCL\_wHeapStringA::OSCL\_wHeapStringA ([OscDefAlloc](#) \* *alloc*, [OscRefCount](#) \* *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*, [OscDefAlloc](#) \* *alloc*, [OscRefCount](#) \* *ref* = NULL)

### 7.86.3.5 OSCL\_IMPORT\_REF OSCL\_wHeapStringA::OSCL\_wHeapStringA (const [OSCL\\_wString](#) & *src*, [OscDefAlloc](#) \* *alloc* = NULL, [OscRefCount](#) \* *ref* = NULL)

### 7.86.3.6 OSCL\_IMPORT\_REF OSCL\_wHeapStringA::OSCL\_wHeapStringA (const [chartype](#) \* *cstr*, [OscDefAlloc](#) \* *alloc* = NULL, [OscRefCount](#) \* *ref* = NULL)

### 7.86.3.7 OSCL\_IMPORT\_REF OSCL\_wHeapStringA::OSCL\_wHeapStringA (const [chartype](#) \* *buf*, uint32 *length*, [OscDefAlloc](#) \* *alloc* = NULL, [OscRefCount](#) \* *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 [char](#)\* OSCL\_wHeapStringA::get\_str () [virtual]

Implements [OSCL\\_wString](#).

#### 7.86.4.5 OSCL\_IMPORT\_REF OSCL\_wHeapStringA& OSCL\_wHeapStringA::operator= (const [char](#)\* *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\\_char](#)\* *buf*, uint32 *length*, [optype](#) *op*)

#### 7.86.4.9 OSCL\_IMPORT\_REF void OSCL\_wHeapStringA::set (const [other\\_char](#)\* *buf*, [optype](#) *op*)

#### 7.86.4.10 OSCL\_IMPORT\_REF void OSCL\_wHeapStringA::set (const [char](#)\* *buf*, uint32 *length*)

### 7.86.5 Friends And Related Function Documentation

#### 7.86.5.1 friend class [OSCL\\_wString](#) [friend]

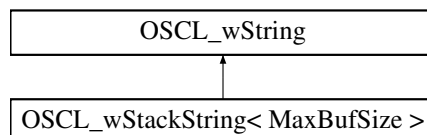
The documentation for this class was generated from the following file:

- [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]`

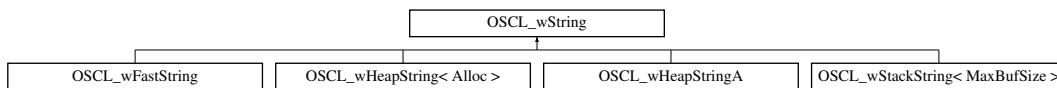
The documentation for this class was generated from the following file:

- [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, [OscDefAlloc](#) \*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 [char](#)**type** *c*)

**7.88.4.11** OSCL\_wString& OSCL\_wString::operator+= (const [char](#)**type** \* *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 [char](#)**type** \* *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 [char](#)**type** \* *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** ]

[char](#)**type** OSCL\_wString::operator[] (uint32 *index*) const



- 7.88.4.22 **virtual [char](#)type OSCL\_wString::read (uint32 *index*) const** [virtual]
- 7.88.4.23 **virtual void OSCL\_wString::set\_len (uint32 *len*)** [protected, pure virtual]
- 7.88.4.24 **virtual void OSCL\_wString::set\_rep (const OSCL\_wString & *src*)** [protected, pure virtual]
- 7.88.4.25 **virtual void OSCL\_wString::set\_rep (const [char](#)type \* *ctr*)** [protected, pure virtual]
- 7.88.4.26 **virtual uint32 OSCL\_wString::setrep\_to\_wide\_char (const char \* *src*, uint32 *len*, [TOSCL\\_wStringOp](#) *op*, [OscDefAlloc](#) \* *aAlloc*)** [virtual]
- 7.88.4.27 **virtual void OSCL\_wString::write (uint32 *offset*, uint32 *length*, const [char](#)type \* *buf*)** [virtual]
- 7.88.4.28 **virtual void OSCL\_wString::write (uint32 *index*, [char](#)type *c*)** [virtual]

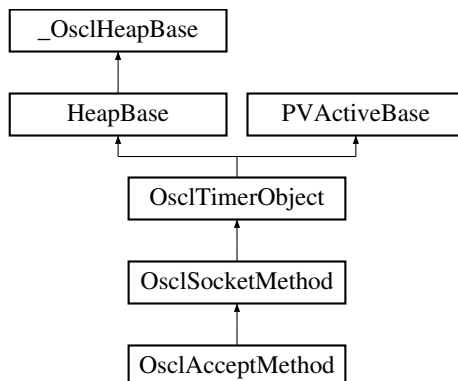
The documentation for this class was generated from the following file:

- [oscl\\_string.h](#)

## 7.89 OsciAcceptMethod Class Reference

```
#include <osci_socket_accept.h>
```

Inheritance diagram for OsciAcceptMethod::



### Public Methods

- [~OsciAcceptMethod\(\)](#)
- [TPVSocketEvent Accept](#) (int32 aTimeout)
- void [DiscardAcceptedSocket\(\)](#)
- [OsciSocketI \\* GetAcceptedSocket\(\)](#)
- [OsciAcceptRequest \\* AcceptRequest\(\)](#)

### Static Public Methods

- [OsciAcceptMethod \\* NewL](#) ([OsciIPSocketI](#) &c)

### 7.89.1 Constructor & Destructor Documentation

#### 7.89.1.1 OsciAcceptMethod::~~OsciAcceptMethod()

### 7.89.2 Member Function Documentation

#### 7.89.2.1 [TPVSocketEvent](#) OsciAcceptMethod::Accept (int32 aTimeout)

#### 7.89.2.2 [OsciAcceptRequest\\*](#) OsciAcceptMethod::AcceptRequest() [inline]

#### 7.89.2.3 void OsciAcceptMethod::DiscardAcceptedSocket()

#### 7.89.2.4 [OsciSocketI\\*](#) OsciAcceptMethod::GetAcceptedSocket()

#### 7.89.2.5 [OsciAcceptMethod\\*](#) OsciAcceptMethod::NewL ([OsciIPSocketI](#) &c) [static]

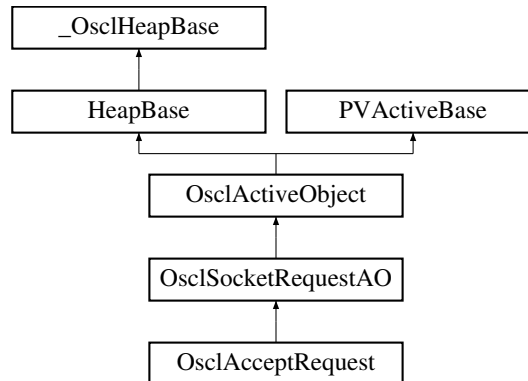
The documentation for this class was generated from the following file:

- [osci\\_socket\\_accept.h](#)

## 7.90 OsciAcceptRequest Class Reference

```
#include <osci_socket_accept.h>
```

Inheritance diagram for OsciAcceptRequest::



### Public Methods

- [OsciAcceptRequest](#) ([OsciSocketMethod](#) &c)
- void [Accept](#) ([OsciSocketI](#) &aSocket)

### 7.90.1 Constructor & Destructor Documentation

**7.90.1.1** OsciAcceptRequest::OsciAcceptRequest ([OsciSocketMethod](#) &c) [inline]

### 7.90.2 Member Function Documentation

**7.90.2.1** void OsciAcceptRequest::Accept ([OsciSocketI](#) &aSocket)

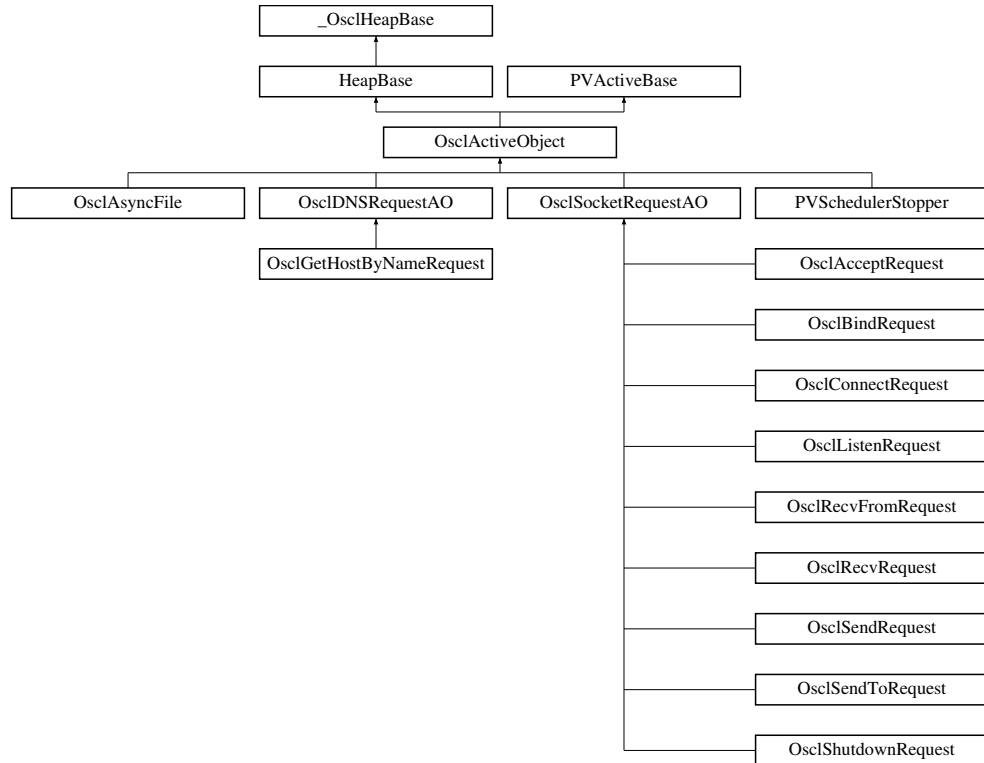
The documentation for this class was generated from the following file:

- [osci\\_socket\\_accept.h](#)

## 7.91 OsciActiveObject Class Reference

```
#include <osci_scheduler_ao.h>
```

Inheritance diagram for OsciActiveObject::



### Public Types

- enum `OsciActivePriority` { `EPriorityIdle` = -100, `EPriorityLow` = -20, `EPriorityNominal` = 0, `EPriorityHigh` = 10, `EPriorityHighest` = 20 }

### Public Methods

- OSCL\_IMPORT\_REF `OsciActiveObject` (int32 aPriority, const char name[ ])
- virtual OSCL\_IMPORT\_REF `~OsciActiveObject` ()
- 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 `OsciAOStatus & StatusRef` ()

## Protected Methods

- virtual OSCL\_IMPORT\_REF void [DoCancel](#) ()
- virtual OSCL\_IMPORT\_REF int32 [RunError](#) (int32 aError)

### 7.91.1 Detailed Description

User base class for execution objects. OsclActiveObject defines an execution object without any timer. This AO can be used across threads, i.e. the request can be activated in one thread and completed in another.

### 7.91.2 Member Enumeration Documentation

#### 7.91.2.1 enum OsclActiveObject::OsclActivePriority

Scheduling priorities.

##### Enumeration values:

**EPriorityIdle** A low priority, useful for execution objects representing background processing.

**EPriorityLow** A priority higher than EPriorityIdle but lower than EPriorityNominal.

**EPriorityNominal** Most exec objects will have this priority.

**EPriorityHigh** A priority higher than EPriorityNominal; useful for execution objects handling user input.

**EPriorityHighest** A priority higher than EPriorityHighest.

### 7.91.3 Constructor & Destructor Documentation

#### 7.91.3.1 OSCL\_IMPORT\_REF OsclActiveObject::OsclActiveObject (int32 *aPriority*, const char *name*[])

Constructor.

##### Parameters:

*aPriority* (input param): scheduling priority

*name* (input param): optional name for this AO.

#### 7.91.3.2 virtual OSCL\_IMPORT\_REF OsclActiveObject::~~OsclActiveObject () [virtual]

Destructor.

### 7.91.4 Member Function Documentation

#### 7.91.4.1 OSCL\_IMPORT\_REF void OsclActiveObject::AddToScheduler ()

Add this exec object to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

#### 7.91.4.2 OSCL\_IMPORT\_REF void OsclActiveObject::Cancel ()

Cancel any pending request. If the request is readied, this will call the DoCancel routine, wait for the request to cancel, then set the request idle. The AO will not run. If the request is not readied, it does nothing. Request must be canceled from the same thread in which it is scheduled.

Reimplemented from [PActiveBase](#).

#### 7.91.4.3 virtual OSCL\_IMPORT\_REF void OsclActiveObject::DoCancel () [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Implements [PActiveBase](#).

Reimplemented in [OsclDNSRequestAO](#), and [OsclSocketRequestAO](#).

#### 7.91.4.4 OSCL\_IMPORT\_REF bool OsclActiveObject::IsBusy ()

Return true if this AO is pending, false otherwise.

#### 7.91.4.5 OSCL\_IMPORT\_REF void OsclActiveObject::PendComplete (int32 aStatus)

Complete the active request for the AO. This API is thread-safe. If the request is not pending, this call will leave.

##### Parameters:

*aStatus*: request completion status.

#### 7.91.4.6 OSCL\_IMPORT\_REF void OsclActiveObject::PendForExec ()

Set request active for this AO and set the status to pending. PendForExec is identical to SetActive, but it additionally sets the request status to OSCL\_REQUEST\_PENDING.

#### 7.91.4.7 OSCL\_IMPORT\_REF int32 OsclActiveObject::Priority ()

Return scheduling priority of this exec object.

#### 7.91.4.8 OSCL\_IMPORT\_REF void OsclActiveObject::RemoveFromScheduler ()

Remove this AO from its scheduler. Will leave if the calling thread context does not match the scheduling thread. Cancels any readied request before removing.

Reimplemented from [PActiveBase](#).

#### 7.91.4.9 virtual OSCL\_IMPORT\_REF int32 OsclActiveObject::RunError (int32 *aError*) [protected, virtual]

Run Error handler. This gets called by scheduler when the Run routine leaves. The default implementation simply returns the leave code. If the derived class wants to handle errors from Run, it may override this. The RunError should return OsclErrNone if it handles the error, otherwise it should return the input error code.

##### Parameters:

***aError:*** the leave code generated by the Run.

Implements [PVActiveBase](#).

#### 7.91.4.10 OSCL\_IMPORT\_REF void OsclActiveObject::RunIfNotReady ()

Complete this AO's request immediately. If the AO is already active, this will do nothing. Will leave if the AO is not added to any scheduler, or if the calling thread context does not match the scheduling thread.

#### 7.91.4.11 OSCL\_IMPORT\_REF void OsclActiveObject::SetBusy ()

Set object ready for this AO, additionally sets the request status to OSCL\_REQUEST\_PENDING. Will leave if the request is already readied, or the execution object is not added to any scheduler, or the calling thread context does not match the scheduler thread.

#### 7.91.4.12 OSCL\_IMPORT\_REF void OsclActiveObject::SetStatus (int32)

#### 7.91.4.13 OSCL\_IMPORT\_REF int32 OsclActiveObject::Status ()

Request status access

#### 7.91.4.14 OSCL\_IMPORT\_REF [OsclAOStatus&](#) OsclActiveObject::StatusRef ()

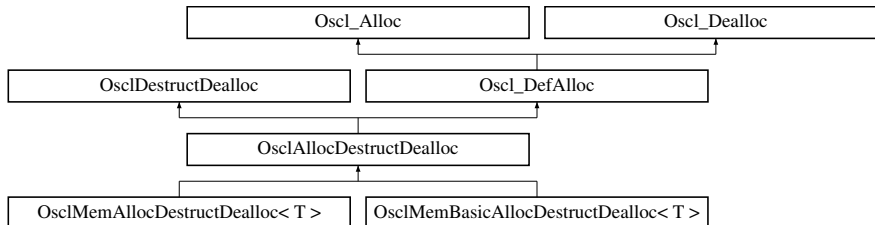
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_ao.h](#)

## 7.92 OsciAllocDestructDealloc Class Reference

```
#include <osci_defalloc.h>
```

Inheritance diagram for OsciAllocDestructDealloc::



### Public Methods

- virtual [~OsciAllocDestructDealloc \(\)](#)

### 7.92.1 Constructor & Destructor Documentation

**7.92.1.1** `virtual OsciAllocDestructDealloc::~~OsciAllocDestructDealloc () [inline, virtual]`

The documentation for this class was generated from the following file:

- [osci\\_defalloc.h](#)



## 7.93 OsclAOSStatus Class Reference

```
#include <oscl_aostatus.h>
```

### Public Methods

- OSCL\_INLINE [OsclAOSStatus](#) ()
- OSCL\_INLINE [OsclAOSStatus](#) (int32 aStatus)
- OSCL\_INLINE int32 [operator=](#) (int32 aStatus)
- OSCL\_INLINE int32 [operator==](#) (int32 aStatus) const
- OSCL\_INLINE int32 [operator!=](#) (int32 aStatus) const
- OSCL\_INLINE int32 [operator>=](#) (int32 aStatus) const
- OSCL\_INLINE int32 [operator<=](#) (int32 aStatus) const
- OSCL\_INLINE int32 [operator>](#) (int32 aStatus) const
- OSCL\_INLINE int32 [operator<](#) (int32 aStatus) const
- OSCL\_INLINE int32 [Value](#) () const

### 7.93.1 Constructor & Destructor Documentation

7.93.1.1 OSCL\_INLINE [OsclAOSStatus::OsclAOSStatus](#) ()

7.93.1.2 OSCL\_INLINE [OsclAOSStatus::OsclAOSStatus](#) (int32 *aStatus*)

### 7.93.2 Member Function Documentation

7.93.2.1 OSCL\_INLINE int32 [OsclAOSStatus::operator!=](#) (int32 *aStatus*) const

7.93.2.2 OSCL\_INLINE int32 [OsclAOSStatus::operator<](#) (int32 *aStatus*) const

7.93.2.3 OSCL\_INLINE int32 [OsclAOSStatus::operator<=](#) (int32 *aStatus*) const

7.93.2.4 OSCL\_INLINE int32 [OsclAOSStatus::operator=](#) (int32 *aStatus*)

7.93.2.5 OSCL\_INLINE int32 [OsclAOSStatus::operator==](#) (int32 *aStatus*) const

7.93.2.6 OSCL\_INLINE int32 [OsclAOSStatus::operator>](#) (int32 *aStatus*) const

7.93.2.7 OSCL\_INLINE int32 [OsclAOSStatus::operator>=](#) (int32 *aStatus*) const

7.93.2.8 OSCL\_INLINE int32 [OsclAOSStatus::Value](#) ()

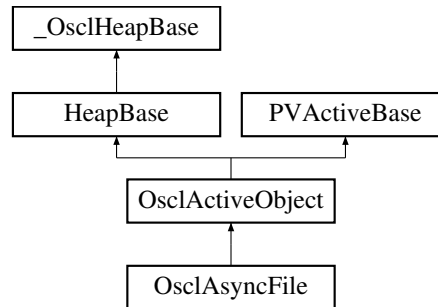
The documentation for this class was generated from the following file:

- [oscl\\_aostatus.h](#)

## 7.94 OsciAsyncFile Class Reference

```
#include <osci_file_async_read.h>
```

Inheritance diagram for OsciAsyncFile::



### Public Methods

- `~OsciAsyncFile ()`
- `int32 Open (const osci_wchar *filename, uint32 mode, const OsciNativeFileParams &params, Osci_FileServer &fileserv)`
- `int32 Open (const char *filename, uint32 mode, const OsciNativeFileParams &params, Osci_FileServer &fileserv)`
- `int32 Seek (TOsciFileOffset offset, Osci_File::seek_type origin)`
- `TOsciFileOffset Tell ()`
- `uint32 Read (OsciAny *aBuffer1, uint32 aDataSize, uint32 aNumElements)`
- `int32 EndOfFile ()`
- `TOsciFileOffset Size ()`
- `int32 Close ()`
- `uint32 Write (const OsciAny *aBuffer1, uint32 aDataSize, uint32 aNumElements)`
- `uint32 Flush ()`

### Static Public Methods

- `OsciAsyncFile * NewL (OsciNativeFile &aAsyncFile, int32 aCacheSize, PVLogger *)`
- `void Delete (OsciAsyncFile *)`

### Data Fields

- `uint32 iNumOfRun`
- `uint32 iNumOfRunErr`

#### 7.94.1 Detailed Description

OsciAsyncFile

## 7.94.2 Constructor & Destructor Documentation

### 7.94.2.1 OsclAsyncFile::~~OsclAsyncFile ()

Destructor.

## 7.94.3 Member Function Documentation

### 7.94.3.1 int32 OsclAsyncFile::Close ()

### 7.94.3.2 void OsclAsyncFile::Delete (OsclAsyncFile \*) [static]

### 7.94.3.3 int32 OsclAsyncFile::EndOfFile ()

### 7.94.3.4 uint32 OsclAsyncFile::Flush () [inline]

### 7.94.3.5 OsclAsyncFile\* OsclAsyncFile::NewL (OsclNativeFile & aAsyncFile, int32 aCacheSize, PVLogger \*) [static]

Two-phased constructor.

#### Parameters:

**aAsyncFile:** open handle for async file read. Note: it is the caller's job to open/close this file handle.

**aSyncFile:** duplicate open handle for sync file read. Note: it is the caller's job to open this file handle, but this class will close the handle.

**aCacheSize:** size of one of the individual cache buffers. The total cached data size will be larger, since multiple buffers are used.

**aStartAsyncRead:** When true, async file read will start immediately. When false, read will not begin until StartAsyncRead is called.

- 7.94.3.6 `int32 OslAsyncFile::Open (const char * filename, uint32 mode, const OslNativeFileParams & params, Osl\_FileServer & fileserv)`
- 7.94.3.7 `int32 OslAsyncFile::Open (const oscl\_wchar * filename, uint32 mode, const OslNativeFileParams & params, Osl\_FileServer & fileserv)`
- 7.94.3.8 `uint32 OslAsyncFile::Read (OslAny * aBuffer1, uint32 aDataSize, uint32 aNumElements)`
- 7.94.3.9 `int32 OslAsyncFile::Seek (TOslFileOffset offset, Osl\_File::seek\_type origin)`
- 7.94.3.10 `TOslFileOffset OslAsyncFile::Size ()`
- 7.94.3.11 `TOslFileOffset OslAsyncFile::Tell ()`
- 7.94.3.12 `uint32 OslAsyncFile::Write (const OslAny * aBuffer1, uint32 aDataSize, uint32 aNumElements) [inline]`

## 7.94.4 Field Documentation

- 7.94.4.1 `uint32 OslAsyncFile::iNumOfRun`
- 7.94.4.2 `uint32 OslAsyncFile::iNumOfRunErr`

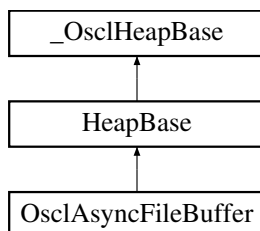
The documentation for this class was generated from the following file:

- [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 OslAuditCB Class Reference

```
#include <osl_mem.h>
```

### Public Methods

- [OslAuditCB](#) ()
- [OslAuditCB](#) (const [OslMemStatsNode](#) \*myStatsNode, [OslMemAudit](#) \*ptr)

### Data Fields

- const [OslMemStatsNode](#) \* pStatsNode
- [OslMemAudit](#) \* pAudit

### 7.96.1 Constructor & Destructor Documentation

**7.96.1.1** [OslAuditCB::OslAuditCB](#) () [inline]

**7.96.1.2** [OslAuditCB::OslAuditCB](#) (const [OslMemStatsNode](#) \* myStatsNode, [OslMemAudit](#) \* ptr) [inline]

### 7.96.2 Field Documentation

**7.96.2.1** [OslMemAudit](#)\* [OslAuditCB::pAudit](#)

**7.96.2.2** const [OslMemStatsNode](#)\* [OslAuditCB::pStatsNode](#)

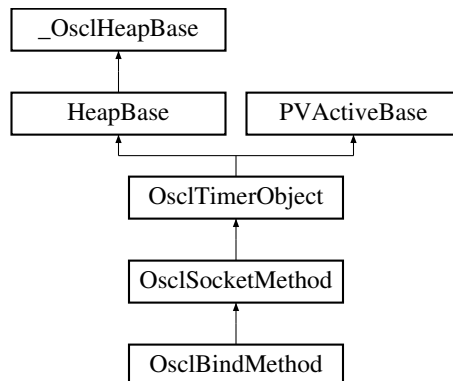
The documentation for this class was generated from the following file:

- [osl\\_mem.h](#)

## 7.97 OsciBindMethod Class Reference

```
#include <osci_socket_bind.h>
```

Inheritance diagram for OsciBindMethod::



### Public Methods

- [~OsciBindMethod \(\)](#)
- [TPVSocketEvent Bind \(OsciNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsciBindRequest \\* BindRequest \(\)](#)

### Static Public Methods

- [OsciBindMethod \\* NewL \(OsciIPSocketI &c\)](#)

### 7.97.1 Constructor & Destructor Documentation

#### 7.97.1.1 OsciBindMethod::~~OsciBindMethod ()

### 7.97.2 Member Function Documentation

#### 7.97.2.1 [TPVSocketEvent](#) OsciBindMethod::Bind ([OsciNetworkAddress](#) & *aAddress*, int32 *aTimeout*)

#### 7.97.2.2 [OsciBindRequest\\*](#) OsciBindMethod::BindRequest () [inline]

#### 7.97.2.3 [OsciBindMethod\\*](#) OsciBindMethod::NewL ([OsciIPSocketI](#) & *c*) [static]

The documentation for this class was generated from the following file:

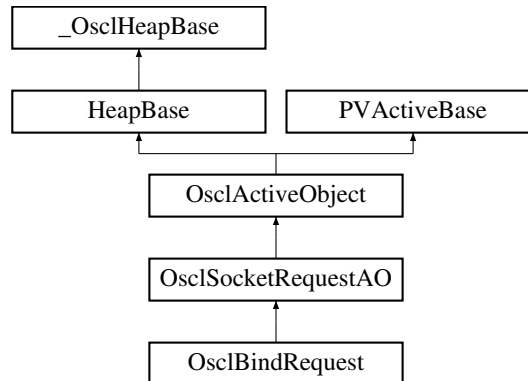
- [osci\\_socket\\_bind.h](#)



## 7.98 OsciBindRequest Class Reference

```
#include <osci_socket_bind.h>
```

Inheritance diagram for OsciBindRequest::



### Public Methods

- [OsciBindRequest](#) ([OsciSocketMethod](#) &c)
- void [Bind](#) ([OsciNetworkAddress](#) &aAddress)

### 7.98.1 Detailed Description

This is the AO that interacts with the socket server

### 7.98.2 Constructor & Destructor Documentation

**7.98.2.1** [OsciBindRequest::OsciBindRequest](#) ([OsciSocketMethod](#) &c) [inline]

### 7.98.3 Member Function Documentation

**7.98.3.1** void [OsciBindRequest::Bind](#) ([OsciNetworkAddress](#) &aAddress)

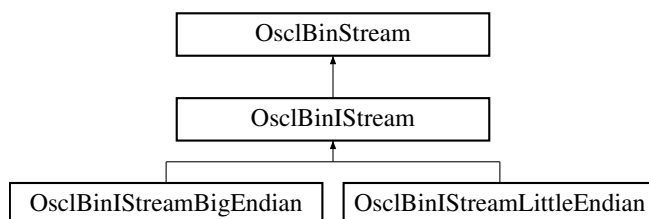
The documentation for this class was generated from the following file:

- [osci\\_socket\\_bind.h](#)

## 7.99 OsciBinIStream Class Reference

```
#include <osci_bin_stream.h>
```

Inheritance diagram for OsciBinIStream::



### Public Methods

- [OsciBinIStream \(\)](#)
- [~OsciBinIStream \(\)](#)
- [uint8 Read\\_uint8 \(\)](#)

*This method reads an unsigned short from the stream.*

- [OsciBinIStream & get \(int8 \\*data, int32 size\)](#)

*This method reads 'length' number of bytes from the stream and places them in 'data'.*

### 7.99.1 Constructor & Destructor Documentation

**7.99.1.1** [OsciBinIStream::OsciBinIStream \(\)](#) [inline]

**7.99.1.2** [OsciBinIStream::~~OsciBinIStream \(\)](#) [inline]

### 7.99.2 Member Function Documentation

**7.99.2.1** [OsciBinIStream& OsciBinIStream::get \(int8 \\* data, int32 size\)](#)

This method reads 'length' number of bytes from the stream and places them in 'data'.

#### Parameters:

*data* is a pointer to the place to store the bytes read

*size* is the number of bytes to read

**7.99.2.2** [uint8 OsciBinIStream::Read\\_uint8 \(\)](#)

This method reads an unsigned short from the stream.

#### Returns:

Unsigned short read from the stream.

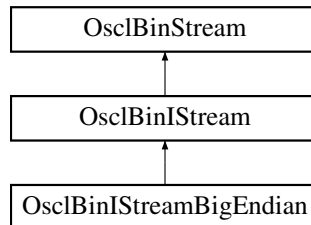
The documentation for this class was generated from the following file:

- [osci\\_bin\\_stream.h](#)

## 7.100 OslBinIStreamBigEndian Class Reference

```
#include <osl_bin_stream.h>
```

Inheritance diagram for OslBinIStreamBigEndian::



### Public Methods

- [OslBinIStreamBigEndian](#) ()
- void [Read](#) (int8 &data)
- void [Read](#) (uint8 &data)
- void [Read](#) (int16 &data)
- void [Read](#) (uint16 &data)
- void [Read](#) (int32 &data)
- void [Read](#) (uint32 &data)
- OslBinIStreamBigEndian & [operator>>](#) (int8 &data)  
*This method reads a int8 from the stream and stores it in 'data'.*
- OslBinIStream & [operator>>](#) (uint8 &data)  
*This method reads a uint8 from the stream and stores it in 'data'.*
- OslBinIStreamBigEndian & [operator>>](#) (int16 &data)  
*This method reads a int16 from the stream and stores it in 'data'.*
- OslBinIStreamBigEndian & [operator>>](#) (uint16 &data)  
*This method reads a uint16 from the stream and stores it in 'data'.*
- OslBinIStreamBigEndian & [operator>>](#) (int32 &data)  
*This method reads a int32 from the stream and stores it in 'data'.*
- OslBinIStreamBigEndian & [operator>>](#) (uint32 &data)  
*This method reads a uint32 from the stream and stores it in 'data'.*
- uint16 [Read\\_uint16](#) ()  
*This method reads an unsigned short from the stream.*
- uint32 [Read\\_uint32](#) ()  
*This method reads an unsigned long from the stream.*

## 7.100.1 Constructor & Destructor Documentation

**7.100.1.1** `OslBinIStreamBigEndian::OslBinIStreamBigEndian ()` `[inline]`

## 7.100.2 Member Function Documentation

**7.100.2.1** `OslBinIStreamBigEndian& OslBinIStreamBigEndian::operator>> (uint32 & data)`

This method reads a uint32 from the stream and stores it in 'data'.

**7.100.2.2** `OslBinIStreamBigEndian& OslBinIStreamBigEndian::operator>> (int32 & data)`

This method reads a int32 from the stream and stores it in 'data'.

**7.100.2.3** `OslBinIStreamBigEndian& OslBinIStreamBigEndian::operator>> (uint16 & data)`

This method reads a uint16 from the stream and stores it in 'data'.

**7.100.2.4** `OslBinIStreamBigEndian& OslBinIStreamBigEndian::operator>> (int16 & data)`

This method reads a int16 from the stream and stores it in 'data'.

**7.100.2.5** `OslBinIStream& OslBinIStreamBigEndian::operator>> (uint8 & data)`

This method reads a uint8 from the stream and stores it in 'data'.

**7.100.2.6** `OslBinIStreamBigEndian& OslBinIStreamBigEndian::operator>> (int8 & data)`

This method reads a int8 from the stream and stores it in 'data'.

**7.100.2.7** `void OslBinIStreamBigEndian::Read (uint32 & data)`

**7.100.2.8** `void OslBinIStreamBigEndian::Read (int32 & data)`

**7.100.2.9** `void OslBinIStreamBigEndian::Read (uint16 & data)`

**7.100.2.10** `void OslBinIStreamBigEndian::Read (int16 & data)`

**7.100.2.11** `void OslBinIStreamBigEndian::Read (uint8 & data)`

**7.100.2.12** `void OslBinIStreamBigEndian::Read (int8 & data)`

**7.100.2.13** `uint16 OslBinIStreamBigEndian::Read_uint16 ()`

This method reads an unsigned short from the stream.

### Returns:

Unsigned short read from the stream.

**7.100.2.14 uint32 OslBinIStreamBigEndian::Read\_uint32 ()**

This method reads an unsigned long from the stream.

**Returns:**

unsigned long read from the stream.

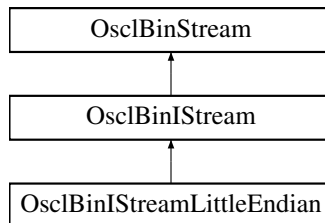
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 7.101 OscBinIStreamLittleEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OscBinIStreamLittleEndian::



### Public Methods

- [OscBinIStreamLittleEndian \(\)](#)
- OscBinIStreamLittleEndian & [operator>>](#) (int8 &data)  
*This method reads a int8 from the stream and stores it in 'data'.*
- OscBinIStreamLittleEndian & [operator>>](#) (uint8 &data)  
*This method reads a uint8 from the stream and stores it in 'data'.*
- OscBinIStreamLittleEndian & [operator>>](#) (int16 &data)  
*This method reads a int16 from the stream and stores it in 'data'.*
- OscBinIStreamLittleEndian & [operator>>](#) (uint16 &data)  
*This method reads a uint16 from the stream and stores it in 'data'.*
- OscBinIStreamLittleEndian & [operator>>](#) (int32 &data)  
*This method reads a int32 from the stream and stores it in 'data'.*
- OscBinIStreamLittleEndian & [operator>>](#) (uint32 &data)  
*This method reads a uint32 from the stream and stores it in 'data'.*

### Protected Methods

- uint16 [Read\\_uint16 \(\)](#)
- uint32 [Read\\_uint32 \(\)](#)

## 7.101.1 Constructor & Destructor Documentation

**7.101.1.1** `OsciBinInputStreamLittleEndian::OsciBinInputStreamLittleEndian ()` `[inline]`

## 7.101.2 Member Function Documentation

**7.101.2.1** `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (uint32 & data)`

This method reads a uint32 from the stream and stores it in 'data'.

**7.101.2.2** `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (int32 & data)`

This method reads a int32 from the stream and stores it in 'data'.

**7.101.2.3** `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (uint16 & data)`

This method reads a uint16 from the stream and stores it in 'data'.

**7.101.2.4** `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (int16 & data)`

This method reads a int16 from the stream and stores it in 'data'.

**7.101.2.5** `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (uint8 & data)`

This method reads a uint8 from the stream and stores it in 'data'.

**7.101.2.6** `OsciBinInputStreamLittleEndian& OsciBinInputStreamLittleEndian::operator>> (int8 & data)`

This method reads a int8 from the stream and stores it in 'data'.

**7.101.2.7** `uint16 OsciBinInputStreamLittleEndian::Read_uint16 ()` `[protected]`

**7.101.2.8** `uint32 OsciBinInputStreamLittleEndian::Read_uint32 ()` `[protected]`

The documentation for this class was generated from the following file:

- [osci\\_bin\\_stream.h](#)

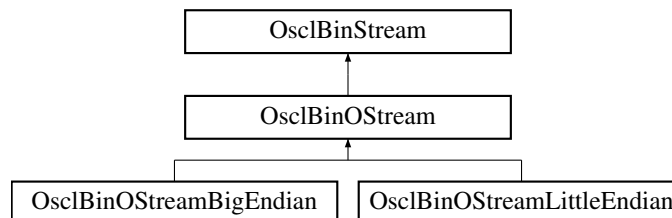


## 7.102 OslBinOStream Class Reference

Class OslBinOStream implements the basic stream functions for an output stream.

```
#include <osl_bin_stream.h>
```

Inheritance diagram for OslBinOStream::



### Public Methods

- [OslBinOStream \(\)](#)
- virtual [~OslBinOStream \(\)](#)
- OslBinOStream & [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 OslBinOStream implements the basic stream functions for an output stream.

### 7.102.2 Constructor & Destructor Documentation

**7.102.2.1** OslBinOStream::OslBinOStream () [inline]

**7.102.2.2** virtual OslBinOStream::~~OslBinOStream () [inline, virtual]

### 7.102.3 Member Function Documentation

**7.102.3.1** OslBinOStream& OslBinOStream::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:

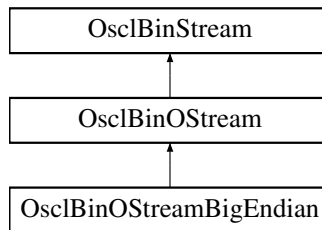
- [osl\\_bin\\_stream.h](#)

## 7.103 OslBinOStreamBigEndian Class Reference

Class OslBinOStreamBigEndian implements a binary output stream using big endian byte ordering.

```
#include <osl_bin_stream.h>
```

Inheritance diagram for OslBinOStreamBigEndian::



### Public Methods

- [OslBinOStreamBigEndian \(\)](#)
- OslBinOStreamBigEndian & [operator<<](#) (const int8 &data)  
*This method writes a int8 from 'data' to the stream.*
- OslBinOStreamBigEndian & [operator<<](#) (const uint8 &data)  
*This method writes a uint8 from 'data' to the stream.*
- OslBinOStreamBigEndian & [operator<<](#) (const int16 &data)  
*This method writes a int16 from 'data' to the stream.*
- OslBinOStreamBigEndian & [operator<<](#) (const uint16 &data)  
*This method writes a uint16 from 'data' to the stream.*
- OslBinOStreamBigEndian & [operator<<](#) (const int32 &data)  
*This method writes a int32 from 'data' to the stream.*
- OslBinOStreamBigEndian & [operator<<](#) (const uint32 &data)  
*This method writes a uint32 from 'data' to the stream.*

### Protected Methods

- void [WriteUnsignedShort](#) (const uint16 data)
- void [WriteUnsignedLong](#) (const uint32 data)

### 7.103.1 Detailed Description

Class OslBinOStreamBigEndian implements a binary output stream using big endian byte ordering.

## 7.103.2 Constructor & Destructor Documentation

**7.103.2.1** `OsciBinOStreamBigEndian::OsciBinOStreamBigEndian ()` `[inline]`

## 7.103.3 Member Function Documentation

**7.103.3.1** `OsciBinOStreamBigEndian& OsciBinOStreamBigEndian::operator<< (const uint32 & data)`

This method writes a uint32 from 'data' to the stream.

**7.103.3.2** `OsciBinOStreamBigEndian& OsciBinOStreamBigEndian::operator<< (const int32 & data)`

This method writes a int32 from 'data' to the stream.

**7.103.3.3** `OsciBinOStreamBigEndian& OsciBinOStreamBigEndian::operator<< (const uint16 & data)`

This method writes a uint16 from 'data' to the stream.

**7.103.3.4** `OsciBinOStreamBigEndian& OsciBinOStreamBigEndian::operator<< (const int16 & data)`

This method writes a int16 from 'data' to the stream.

**7.103.3.5** `OsciBinOStreamBigEndian& OsciBinOStreamBigEndian::operator<< (const uint8 & data)`

This method writes a uint8 from 'data' to the stream.

**7.103.3.6** `OsciBinOStreamBigEndian& OsciBinOStreamBigEndian::operator<< (const int8 & data)`

This method writes a int8 from 'data' to the stream.

**7.103.3.7** `void OsciBinOStreamBigEndian::WriteUnsignedLong (const uint32 data)`  
`[protected]`

**7.103.3.8** `void OsciBinOStreamBigEndian::WriteUnsignedShort (const uint16 data)`  
`[protected]`

The documentation for this class was generated from the following file:

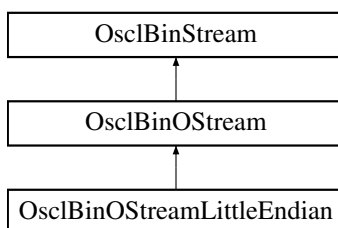
- [osci\\_bin\\_stream.h](#)

## 7.104 OsciBinOStreamLittleEndian Class Reference

Class OsciBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.

```
#include <osci_bin_stream.h>
```

Inheritance diagram for OsciBinOStreamLittleEndian::



### Public Methods

- [OsciBinOStreamLittleEndian \(\)](#)
- OsciBinOStreamLittleEndian & [operator<<](#) (const int8 &data)  
*This method writes a int8 from 'data' to the stream.*
- OsciBinOStreamLittleEndian & [operator<<](#) (const uint8 &data)  
*This method writes a uint8 from 'data' to the stream.*
- OsciBinOStreamLittleEndian & [operator<<](#) (const int16 &data)  
*This method writes a int16 from 'data' to the stream.*
- OsciBinOStreamLittleEndian & [operator<<](#) (const uint16 &data)  
*This method writes a uint16 from 'data' to the stream.*
- OsciBinOStreamLittleEndian & [operator<<](#) (const int32 &data)  
*This method writes a int32 from 'data' to the stream.*
- OsciBinOStreamLittleEndian & [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 OsciBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.

## 7.104.2 Constructor & Destructor Documentation

**7.104.2.1** `OsciBinOStreamLittleEndian::OsciBinOStreamLittleEndian ()` `[inline]`

## 7.104.3 Member Function Documentation

**7.104.3.1** `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const uint32 & data)`

This method writes a uint32 from 'data' to the stream.

**7.104.3.2** `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const int32 & data)`

This method writes a int32 from 'data' to the stream.

**7.104.3.3** `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const uint16 & data)`

This method writes a uint16 from 'data' to the stream.

**7.104.3.4** `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const int16 & data)`

This method writes a int16 from 'data' to the stream.

**7.104.3.5** `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const uint8 & data)`

This method writes a uint8 from 'data' to the stream.

**7.104.3.6** `OsciBinOStreamLittleEndian& OsciBinOStreamLittleEndian::operator<< (const int8 & data)`

This method writes a int8 from 'data' to the stream.

**7.104.3.7** `void OsciBinOStreamLittleEndian::WriteUnsignedLong (const uint32 data)`  
`[protected]`

This method writes 'data' (unsigned long) to the stream.

**7.104.3.8** `void OsciBinOStreamLittleEndian::WriteUnsignedShort (const uint16 data)`  
`[protected]`

This method writes 'data' (unsigned short) to the stream.

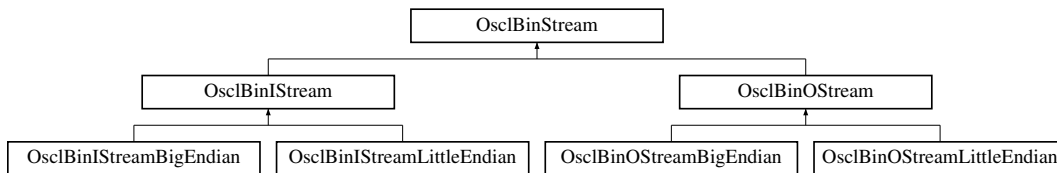
The documentation for this class was generated from the following file:

- [osci\\_bin\\_stream.h](#)

## 7.105 OsciBinStream Class Reference

```
#include <osci_bin_stream.h>
```

Inheritance diagram for OsciBinStream::



### Public Methods

- [OsciBinStream](#) ()
- bool [good](#) ()  
*This method determines if the stream is ok.*
- bool [eof](#) ()  
*This method determines if end of stream has been reached.*
- bool [fail](#) ()  
*This method determines if an error has occurred in the stream.*
- void [Attach](#) (void \*buffer, uint32 l\_length)  
*This methods specifies the data buffer to attach to the stream.*
- void [Attach](#) (const uint32 numFragments, const [OsciMemoryFragment](#) \*fragPtr)  
*This method specifies the memory fragment array to use for input.*
- uint32 [tellg](#) ()  
*This method returns the current stream position.*
- void [Seek](#) (uint32 absPosition)  
*This method seeks to the specified stream position.*
- uint32 [PositionInBlock](#) ()  
*This method returns the current stream position.*
- void [seekFromCurrentPosition](#) (int32 offset)  
*This method seeks to the specified offset from the current location.*

### Protected Types

- enum [state\\_t](#) { [GOOD\\_STATE](#), [EOF\\_STATE](#), [FAIL\\_STATE](#) }

## Protected Methods

- bool [ReserveSpace](#) (uint32 size)
- bool [HaveRoomInCurrentBlock](#) (uint32 size)

## Protected Attributes

- [state\\_t](#) *state*
- uint8 \* [pBasePosition](#)
- uint8 \* [pPosition](#)
- uint32 [length](#)
- const [OsciMemoryFragment](#) \* [nextFragPtr](#)
- int [fragsLeft](#)
- const [OsciMemoryFragment](#) \* [firstFragPtr](#)
- int [numFrag](#)s
- [OsciMemoryFragment](#) [specialFragBuffer](#)

## 7.105.1 Member Enumeration Documentation

### 7.105.1.1 enum OsciBinStream::state\_t [protected]

Enumeration values:

GOOD\_STATE

EOF\_STATE

FAIL\_STATE

## 7.105.2 Constructor & Destructor Documentation

### 7.105.2.1 OsciBinStream::OsciBinStream () [inline]

## 7.105.3 Member Function Documentation

### 7.105.3.1 void OsciBinStream::Attach (const uint32 *numFragments*, const [OsciMemoryFragment](#) \* *fragPtr*)

This method specifies the memory fragment array to use for input.

This array should remain static while the stream refers to it.

**Parameters:**

*numFragments* is the number of elements in the array

*fragPtr* is the pointer to the MemoryFragment array

### 7.105.3.2 void OsciBinStream::Attach (void \* *buffer*, uint32 *l\_length*)

This methods specifies the data buffer to attach to the stream.

**Parameters:**

*buffer* will provide the input

*length* of the buffer

**7.105.3.3 bool OsciBinStream::eof ()**

This method determines if end of stream has been reached.

**Returns:**

true if end of stream has been reached.

**7.105.3.4 bool OsciBinStream::fail ()**

This method determines if an error has occurred in the stream.

**Returns:**

true if an error occurred in the stream.

**7.105.3.5 bool OsciBinStream::good ()**

This method determines if the stream is ok.

**Returns:**

true if stream is ok.

**7.105.3.6 bool OsciBinStream::HaveRoomInCurrentBlock (uint32 *size*) [protected]****7.105.3.7 uint32 OsciBinStream::PositionInBlock ()**

This method returns the current stream position.

**Returns:**

stream position.

**7.105.3.8 bool OsciBinStream::ReserveSpace (uint32 *size*) [protected]****7.105.3.9 void OsciBinStream::Seek (uint32 *absPosition*)**

This method seeks to the specified stream position.

**Returns:**

Stream position.

**7.105.3.10 void OsciBinStream::seekFromCurrentPosition (int32 *offset*)**

This method seeks to the specified offset from the current location.

**Parameters:**

*offset* from current stream location



**7.105.3.11 uint32 OsciBinStream::tellg ()**

This method returns the current stream position.

This method is to be used if the input stream is a pointer to the MemoryFragment array

**Returns:**

Stream position.

**7.105.4 Field Documentation**

**7.105.4.1** const [OsciMemoryFragment](#)\* OsciBinStream::firstFragPtr [protected]

**7.105.4.2** int OsciBinStream::fragsLeft [protected]

**7.105.4.3** uint32 OsciBinStream::length [protected]

**7.105.4.4** const [OsciMemoryFragment](#)\* OsciBinStream::nextFragPtr [protected]

**7.105.4.5** int OsciBinStream::numFrag [protected]

**7.105.4.6** uint8\* OsciBinStream::pBasePosition [protected]

**7.105.4.7** uint8\* OsciBinStream::pPosition [protected]

**7.105.4.8** [OsciMemoryFragment](#) OsciBinStream::specialFragBuffer [protected]

**7.105.4.9** [state\\_t](#) OsciBinStream::state [protected]

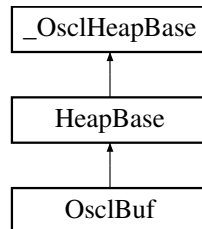
The documentation for this class was generated from the following file:

- [osci\\_bin\\_stream.h](#)

## 7.106 OslBuf Class Reference

```
#include <osl_file_async_read.h>
```

Inheritance diagram for OslBuf::



### Public Methods

- [OslBuf](#) (int32 size)
- int32 [Length](#) ()
- [OslPtr Des](#) ()
- [OslPtrC DesC](#) ()

### Static Public Methods

- OslBuf \* [NewL](#) (int32 size)
- void [Delete](#) (OslBuf \*a)

### Data Fields

- uint8 \* [iBuffer](#)
- int32 [iMaxLength](#)
- int32 [iLength](#)

## 7.106.1 Constructor & Destructor Documentation

7.106.1.1 `OslBuf::OslBuf (int32 size)` [inline]

## 7.106.2 Member Function Documentation

7.106.2.1 `void OslBuf::Delete (OslBuf * a)` [inline, static]

7.106.2.2 [OslPtr](#) `OslBuf::Des ()` [inline]

7.106.2.3 [OslPtrC](#) `OslBuf::DesC ()` [inline]

7.106.2.4 `int32 OslBuf::Length ()` [inline]

7.106.2.5 `OslBuf* OslBuf::NewL (int32 size)` [inline, static]

## 7.106.3 Field Documentation

7.106.3.1 `uint8* OslBuf::iBuffer`

7.106.3.2 `int32 OslBuf::iLength`

7.106.3.3 `int32 OslBuf::iMaxLength`

The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 7.107 OsciCompareLess< T > Class Template Reference

```
#include <osci_priqueue.h>
```

### Public Methods

- int [compare](#) (T &a, T &b) const

```
template<class T> class OsciCompareLess< T >
```

### 7.107.1 Member Function Documentation

**7.107.1.1** `template<class T> int OsciCompareLess< T >::compare (T & a, T & b) const`  
[inline]

The documentation for this class was generated from the following file:

- [osci\\_priqueue.h](#)

## 7.108 OsciComponentRegistry Class Reference

```
#include <osci_registry_serv_impl.h>
```

### Public Methods

- [OsciComponentRegistry \(\)](#)
- [~OsciComponentRegistry \(\)](#)
- [int32 Register](#) ([uint32 &aId](#), [OSCL\\_String &](#), [OsciComponentFactory](#))
- [int32 Unregister](#) ([OSCL\\_String &](#))
- [int32 Unregister](#) ([uint32](#))
- [OsciComponentFactory FindExact](#) ([OSCL\\_String &](#))
- [void FindHierarchical](#) ([OSCL\\_String &](#), [Osci\\_Vector< OsciRegistryAccessElement, OsciMem-Allocator > &](#))
- [void OpenSession \(\)](#)
- [void CloseSession \(\)](#)

### Data Fields

- [OsciComponentRegistryData iData](#)
- [OsciMutex iMutex](#)
- [uint32 iComponentIdCounter](#)
- [uint32 iNumSessions](#)

### 7.108.1 Detailed Description

Thread-safe singleton registry object.

## 7.108.2 Constructor & Destructor Documentation

7.108.2.1 OslComponentRegistry::OslComponentRegistry ()

7.108.2.2 OslComponentRegistry::~~OslComponentRegistry ()

## 7.108.3 Member Function Documentation

7.108.3.1 void OslComponentRegistry::CloseSession ()

7.108.3.2 [OslComponentFactory](#) OslComponentRegistry::FindExact ([OSCL\\_String](#) &)

7.108.3.3 void OslComponentRegistry::FindHierarchical ([OSCL\\_String](#) &, [Osl\\_Vector](#)<  
[OslRegistryAccessElement](#), [OslMemAllocator](#) > &)

7.108.3.4 void OslComponentRegistry::OpenSession ()

7.108.3.5 int32 OslComponentRegistry::Register (uint32 & *aId*, [OSCL\\_String](#) &, [OslComponentFactory](#))

7.108.3.6 int32 OslComponentRegistry::Unregister (uint32)

7.108.3.7 int32 OslComponentRegistry::Unregister ([OSCL\\_String](#) &)

## 7.108.4 Field Documentation

7.108.4.1 uint32 OslComponentRegistry::iComponentIdCounter

7.108.4.2 [OslComponentRegistryData](#) OslComponentRegistry::iData

7.108.4.3 [OslMutex](#) OslComponentRegistry::iMutex

7.108.4.4 uint32 OslComponentRegistry::iNumSessions

The documentation for this class was generated from the following file:

- [oscl\\_registry\\_serv\\_impl.h](#)

## 7.109 OsciComponentRegistryData Class Reference

```
#include <osci_registry_serv_impl.h>
```

### Public Methods

- [OsciComponentRegistryElement](#) \* Find ([OSCL\\_String](#) &, bool aExact)

### Data Fields

- [Osci\\_Vector](#)< [OsciComponentRegistryElement](#), [OsciMemAllocator](#) > iVec

### 7.109.1 Detailed Description

Registry

### 7.109.2 Member Function Documentation

- 7.109.2.1 [OsciComponentRegistryElement](#)\* OsciComponentRegistryData::Find ([OSCL\\_String](#) &, bool *aExact*)

### 7.109.3 Field Documentation

- 7.109.3.1 [Osci\\_Vector](#)<[OsciComponentRegistryElement](#), [OsciMemAllocator](#)>  
OsciComponentRegistryData::iVec

The documentation for this class was generated from the following file:

- [osci\\_registry\\_serv\\_impl.h](#)

## 7.110 OsciComponentRegistryElement Class Reference

```
#include <osci_registry_serv_impl.h>
```

### Public Methods

- [OsciComponentRegistryElement](#) ([OSCL\\_String](#) &, [OsciComponentFactory](#))
- [OsciComponentRegistryElement](#) (const [OsciComponentRegistryElement](#) &)
- [OsciComponentRegistryElement](#) & [operator=](#) (const [OsciComponentRegistryElement](#) &src)
- [~OsciComponentRegistryElement](#) ()
- bool [Match](#) ([OSCL\\_String](#) &aStr, bool aExact)

### Data Fields

- [OSCL\\_String](#) \* iId
- [OsciComponentFactory](#) iFactory
- uint32 [iComponentId](#)

### 7.110.1 Detailed Description

Data for each registered component.

### 7.110.2 Constructor & Destructor Documentation

7.110.2.1 [OsciComponentRegistryElement::OsciComponentRegistryElement](#) ([OSCL\\_String](#) &, [OsciComponentFactory](#))

7.110.2.2 [OsciComponentRegistryElement::OsciComponentRegistryElement](#) (const [OsciComponentRegistryElement](#) &)

7.110.2.3 [OsciComponentRegistryElement::~~OsciComponentRegistryElement](#) ()

### 7.110.3 Member Function Documentation

7.110.3.1 bool [OsciComponentRegistryElement::Match](#) ([OSCL\\_String](#) & aStr, bool aExact)

7.110.3.2 [OsciComponentRegistryElement&](#) [OsciComponentRegistryElement::operator=](#) (const [OsciComponentRegistryElement](#) & src)

### 7.110.4 Field Documentation

7.110.4.1 uint32 [OsciComponentRegistryElement::iComponentId](#)

7.110.4.2 [OsciComponentFactory](#) [OsciComponentRegistryElement::iFactory](#)

7.110.4.3 [OSCL\\_String\\*](#) [OsciComponentRegistryElement::iId](#)

The documentation for this class was generated from the following file:

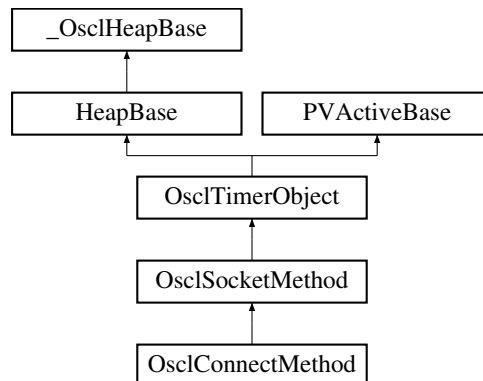


- [osci\\_registry\\_serv\\_impl.h](#)

## 7.111 OsciConnectMethod Class Reference

```
#include <osci_socket_connect.h>
```

Inheritance diagram for OsciConnectMethod::



### Public Methods

- [~OsciConnectMethod \(\)](#)
- [TPVSocketEvent Connect \(OsciNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsciConnectRequest \\* ConnectRequest \(\)](#)

### Static Public Methods

- [OsciConnectMethod \\* NewL \(OsciIPSocketI &c\)](#)

### 7.111.1 Constructor & Destructor Documentation

#### 7.111.1.1 OsciConnectMethod::~~OsciConnectMethod ()

### 7.111.2 Member Function Documentation

#### 7.111.2.1 [TPVSocketEvent](#) OsciConnectMethod::Connect ([OsciNetworkAddress](#) & *aAddress*, *int32 aTimeout*)

#### 7.111.2.2 [OsciConnectRequest\\*](#) OsciConnectMethod::ConnectRequest () [inline]

#### 7.111.2.3 [OsciConnectMethod\\*](#) OsciConnectMethod::NewL ([OsciIPSocketI](#) & *c*) [static]

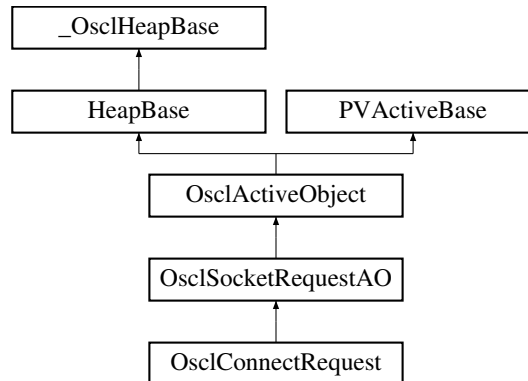
The documentation for this class was generated from the following file:

- [osci\\_socket\\_connect.h](#)

## 7.112 OsciConnectRequest Class Reference

```
#include <osci_socket_connect.h>
```

Inheritance diagram for OsciConnectRequest::



### Public Methods

- [OsciConnectRequest](#) ([OsciSocketMethod](#) &c)
- void [Connect](#) ([OsciNetworkAddress](#) &aAddress)

### 7.112.1 Detailed Description

This is the AO that interacts with the socket server

### 7.112.2 Constructor & Destructor Documentation

7.112.2.1 [OsciConnectRequest::OsciConnectRequest](#) ([OsciSocketMethod](#) &c) [inline]

### 7.112.3 Member Function Documentation

7.112.3.1 void [OsciConnectRequest::Connect](#) ([OsciNetworkAddress](#) &aAddress)

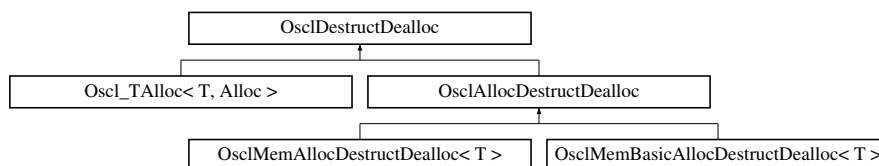
The documentation for this class was generated from the following file:

- [osci\\_socket\\_connect.h](#)

## 7.113 OsciDestructDealloc Class Reference

```
#include <osci_defalloc.h>
```

Inheritance diagram for OsciDestructDealloc::



### Public Methods

- virtual [~OsciDestructDealloc](#) ()
- virtual void [destruct\\_and\\_dealloc](#) ([OsciAny](#) \*ptr)=0

### 7.113.1 Constructor & Destructor Documentation

**7.113.1.1** virtual [OsciDestructDealloc::~OsciDestructDealloc](#) () [inline, virtual]

### 7.113.2 Member Function Documentation

**7.113.2.1** virtual void [OsciDestructDealloc::destruct\\_and\\_dealloc](#) ([OsciAny](#) \* ptr) [pure virtual]

Implemented in [Osci\\_TAlloc< T, Alloc >](#), [OsciMemAllocDestructDealloc< T >](#), [OsciMemBasicAllocDestructDealloc< T >](#), [Osci\\_TAlloc< entry\\_type, Alloc >](#), [Osci\\_TAlloc< node\\_type, TagTree\\_Allocator >](#), [Osci\\_TAlloc< node\\_type, alloc\\_type >](#), [Osci\\_TAlloc< MM\\_StatsNodeTagTreeType, OsciMemBasicAllocator >](#), [Osci\\_TAlloc< char, alloc\\_type >](#), [Osci\\_TAlloc< tag\\_base\\_unit, Alloc >](#), [Osci\\_TAlloc< PVLogger, alloc\\_type >](#), and [Osci\\_TAlloc< node\\_type, Alloc >](#).

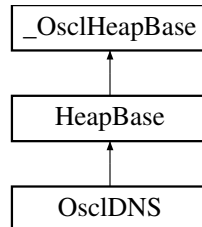
The documentation for this class was generated from the following file:

- [osci\\_defalloc.h](#)

## 7.114 OsciDNS Class Reference

```
#include <osci_dns.h>
```

Inheritance diagram for OsciDNS::



### Public Methods

- OSCI\_IMPORT\_REF [~OsciDNS \(\)](#)
- OSCI\_IMPORT\_REF [TPVDNSEvent GetHostByName](#) (char \*name, [OsciNetworkAddress](#) &addr, int32 aTimeoutMsec=-1, [Osci\\_Vector](#)< [OsciNetworkAddress](#), [OsciMemAllocator](#) > \*aAddressList=NULL)
- OSCI\_IMPORT\_REF void [CancelGetHostByName \(\)](#)

### Static Public Methods

- OSCI\_IMPORT\_REF OsciDNS \* [NewL](#) ([Osci\\_DefAlloc](#) &alloc, [OsciSocketServ](#) &aServ, [OsciDNSObserver](#) &aObserver, uint32 aId)

### Friends

- class [OsciDNSRequestAO](#)

### 7.114.1 Detailed Description

The DNS class

### 7.114.2 Constructor & Destructor Documentation

#### 7.114.2.1 OSCI\_IMPORT\_REF OsciDNS::~~OsciDNS ()

Destructor.

Note: the application must de-allocate the DNS object using the same allocator that was passed in the NewL object creation call.

### 7.114.3 Member Function Documentation

#### 7.114.3.1 OSCI\_IMPORT\_REF void OsciDNS::CancelGetHostByName ()

Cancel GetHostByName

This method will cancel any pending GetHostByName operation on the current object, causing the GetHostByName to complete with error EPVDNSCancel. If there is no pending GetHostByName operation, this method will have no effect.

**7.114.3.2** `OSCL_IMPORT_REF TPVDNSEvent OsciDNS::GetHostByName (char * name, OsciNetworkAddress & addr, int32 aTimeoutMsec = -1, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > * aAddressList = NULL)`

GetHostByName. This is an asynchronous method.

#### Parameters:

- name:** Null-terminated string containing the host name.
- addr:** The output address corresponding to the host. The ipAddr field will contain the network address of the host in dotted decimal notation.
- aTimeoutMsec:** A timeout for the request in milliseconds, or (-1) to indicate infinite wait.
- aAddressList :** A list of addresses for the host. @returns: EPVDNSPending for success, EPVDNS-Failure for failure.

**7.114.3.3** `OSCL_IMPORT_REF OsciDNS* OsciDNS::NewL (Osci_DefAlloc & alloc, OsciSocketServ & aServ, OsciDNSObserver & aObserver, uint32 aId) [static]`

DNS object creation.

#### Parameters:

- alloc:** Memory allocator
- aServ:** Socket server.
- aObserver:** DNS Event observer
- aId:** Unique ID for this DNS object. This ID will be included in all callbacks associated with this DNS object.

## 7.114.4 Friends And Related Function Documentation

**7.114.4.1** `friend class OsciDNSRequestAO [friend]`

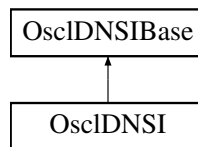
The documentation for this class was generated from the following file:

- [osci\\_dns.h](#)

## 7.115 OsciDNSI Class Reference

```
#include <osci_dns_imp_pv.h>
```

Inheritance diagram for OsciDNSI::



### Public Methods

- [~OsciDNSI](#) ()
- int32 [Open](#) (OsciSocketServI &aServer)
- int32 [Close](#) ()
- void [GetHostByName](#) (GetHostByNameParam &, OsciDNSRequestAO &)
- void [GetHostByNameSuccess](#) (GetHostByNameParam &)
- void [GetNextHost](#) (OsciDNSRequestAO &)
- void [GetNextHostSuccess](#) (GetHostByNameParam &)
- bool [GetHostByNameResponseContainsAliasInfo](#) ()

### Static Public Methods

- OsciDNSI \* [NewL](#) (Osci\_DefAlloc &a)

### Friends

- class [OsciDNSRequest](#)
- class [OsciGetHostByNameRequest](#)
- class [DNSRequestParam](#)

### 7.115.1 Detailed Description

OsciDNSI, non-Symbian implementation

### 7.115.2 Constructor & Destructor Documentation

#### 7.115.2.1 OsciDNSI::~~OsciDNSI ()

### 7.115.3 Member Function Documentation

#### 7.115.3.1 int32 OsciDNSI::Close () [virtual]

Implements [OsciDNSIBase](#).

**7.115.3.2** void OsciDNSI::GetHostByName ([GetHostByNameParam](#) &, [OsciDNSRequestAO](#) &) [virtual]

Implements [OsciDNSIBase](#).

**7.115.3.3** bool OsciDNSI::GetHostByNameResponseContainsAliasInfo () [virtual]

Implements [OsciDNSIBase](#).

**7.115.3.4** void OsciDNSI::GetHostByNameSuccess ([GetHostByNameParam](#) &) [virtual]

Implements [OsciDNSIBase](#).

**7.115.3.5** void OsciDNSI::GetNextHost ([OsciDNSRequestAO](#) &) [virtual]

Implements [OsciDNSIBase](#).

**7.115.3.6** void OsciDNSI::GetNextHostSuccess ([GetHostByNameParam](#) &) [virtual]

Implements [OsciDNSIBase](#).

**7.115.3.7** OsciDNSI\* OsciDNSI::NewL ([Osci\\_DefAlloc](#) & a) [static]

**7.115.3.8** int32 OsciDNSI::Open ([OsciSocketServI](#) & aServer) [virtual]

Implements [OsciDNSIBase](#).

## 7.115.4 Friends And Related Function Documentation

**7.115.4.1** friend class DNSRequestParam [friend]

**7.115.4.2** friend class OsciDNSRequest [friend]

Reimplemented from [OsciDNSIBase](#).

**7.115.4.3** friend class OsciGetHostByNameRequest [friend]

Reimplemented from [OsciDNSIBase](#).

The documentation for this class was generated from the following file:

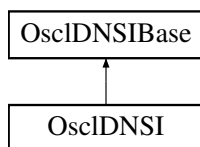
- [osci\\_dns\\_imp\\_pv.h](#)



## 7.116 OsciDNSIBase Class Reference

```
#include <osci_dns_imp_base.h>
```

Inheritance diagram for OsciDNSIBase::



### Public Methods

- virtual [~OsciDNSIBase](#) ()
- virtual int32 [Open](#) ([OsciSocketServI](#) &aServer)=0
- virtual int32 [Close](#) ()=0
- virtual void [GetHostByName](#) ([GetHostByNameParam](#) &, [OsciDNSRequestAO](#) &)=0
- virtual void [GetHostByNameSuccess](#) ([GetHostByNameParam](#) &)=0
- virtual bool [GetHostByNameResponseContainsAliasInfo](#) ()=0
- virtual void [GetNextHost](#) ([OsciDNSRequestAO](#) &)=0
- virtual void [GetNextHostSuccess](#) ([GetHostByNameParam](#) &)=0
- void [CancelFxn](#) ([TPVDNSFxn](#))

### Protected Methods

- [OsciDNSIBase](#) ([Osci\\_DefAlloc](#) &a)
- virtual bool [IsReady](#) ([OsciDNSRequestAO](#) &aObject)=0
- virtual void [CancelGetHostByName](#) ()=0

### Protected Attributes

- [Osci\\_DefAlloc](#) & iAlloc
- [OsciSocketServI](#) \* iSocketServ

### Friends

- class [OsciDNSRequest](#)
- class [OsciGetHostByNameRequest](#)

### 7.116.1 Detailed Description

OsciDNSIBase is a common base class for all implementations.

## 7.116.2 Constructor & Destructor Documentation

**7.116.2.1** `virtual OsciDNSIBase::~~OsciDNSIBase ()` [virtual]

**7.116.2.2** `OsciDNSIBase::OsciDNSIBase (Osci_DefAlloc & a)` [protected]

## 7.116.3 Member Function Documentation

**7.116.3.1** `void OsciDNSIBase::CancelFxn (TPVDNSFxn)`

**7.116.3.2** `virtual void OsciDNSIBase::CancelGetHostByName ()` [protected, pure virtual]

**7.116.3.3** `virtual int32 OsciDNSIBase::Close ()` [pure virtual]

Implemented in [OsciDNSI](#).

**7.116.3.4** `virtual void OsciDNSIBase::GetHostByName (GetHostByNameParam &, OsciDNSRequestAO &)` [pure virtual]

Implemented in [OsciDNSI](#).

**7.116.3.5** `virtual bool OsciDNSIBase::GetHostByNameResponseContainsAliasInfo ()` [pure virtual]

Implemented in [OsciDNSI](#).

**7.116.3.6** `virtual void OsciDNSIBase::GetHostByNameSuccess (GetHostByNameParam &)` [pure virtual]

Implemented in [OsciDNSI](#).

**7.116.3.7** `virtual void OsciDNSIBase::GetNextHost (OsciDNSRequestAO &)` [pure virtual]

Implemented in [OsciDNSI](#).

**7.116.3.8** `virtual void OsciDNSIBase::GetNextHostSuccess (GetHostByNameParam &)` [pure virtual]

Implemented in [OsciDNSI](#).

**7.116.3.9** `virtual bool OsciDNSIBase::IsReady (OsciDNSRequestAO & aObject)` [protected, pure virtual]

**7.116.3.10** `virtual int32 OsciDNSIBase::Open (OsciSocketServI & aServer)` [pure virtual]

Implemented in [OsciDNSI](#).

## 7.116.4 Friends And Related Function Documentation

### 7.116.4.1 friend class OsciDNSRequest [friend]

Reimplemented in [OsciDNSI](#).

### 7.116.4.2 friend class OsciGetHostByNameRequest [friend]

Reimplemented in [OsciDNSI](#).

## 7.116.5 Field Documentation

### 7.116.5.1 [OsciDefAlloc](#)& OsciDNSIBase::iAlloc [protected]

### 7.116.5.2 [OsciSocketServI](#)\* OsciDNSIBase::iSocketServ [protected]

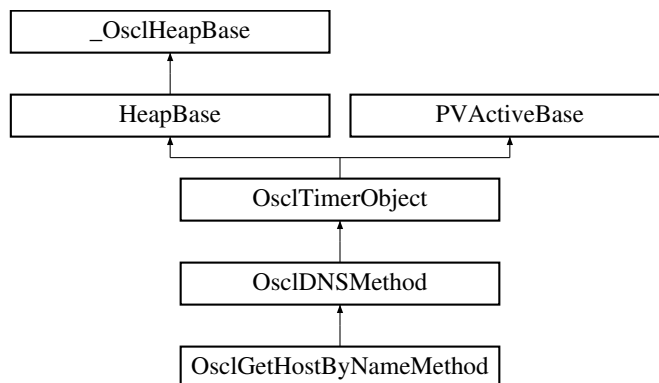
The documentation for this class was generated from the following file:

- [osci\\_dns\\_imp\\_base.h](#)

## 7.117 OsciDNSMethod Class Reference

```
#include <osci_dns_method.h>
```

Inheritance diagram for OsciDNSMethod::



### Public Methods

- [OsciDNSMethod](#) ([Osci\\_DefAlloc](#) &a, const char \*name, [TPVDNSFxn](#) fxn)
- void [Abort](#) ()
- void [AbortAll](#) ()
- void [CancelMethod](#) ()
- void [Run](#) ()

### Data Fields

- [OsciDNSObserver](#) \* [iDNSObserver](#)
- uint32 [iId](#)
- [Osci\\_DefAlloc](#) & [iAlloc](#)
- [TPVDNSFxn](#) [iDNSFxn](#)
- [PVLogger](#) \* [iLogger](#)

### Protected Methods

- void [ConstructL](#) ([OsciDNSObserver](#) \*aObserver, [OsciDNSRequestAO](#) \*aAO, uint32 aId)
- bool [StartMethod](#) (int32 aTimeoutMsec)
- void [MethodDone](#) ()

### Protected Attributes

- [OsciDNSRequestAO](#) \* [iDNSRequestAO](#)

#### 7.117.1 Detailed Description

This is the base class for all socket methods. It provides the timeout on socket requests.

## 7.117.2 Constructor & Destructor Documentation

**7.117.2.1** OsciDNSMethod::OsciDNSMethod ([Osci\\_DefAlloc](#) & *a*, const char \* *name*, [TPVDNSFxn](#) *fxn*) [inline]

## 7.117.3 Member Function Documentation

**7.117.3.1** void OsciDNSMethod::Abort ()

**7.117.3.2** void OsciDNSMethod::AbortAll ()

**7.117.3.3** void OsciDNSMethod::CancelMethod ()

**7.117.3.4** void OsciDNSMethod::ConstructL ([OsciDNSObserver](#) \* *aObserver*, [OsciDNSRequestAO](#) \* *aAO*, uint32 *aId*) [protected]

**7.117.3.5** void OsciDNSMethod::MethodDone () [protected]

**7.117.3.6** void OsciDNSMethod::Run () [virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's [Run\(\)](#) function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PActiveBase](#).

**7.117.3.7** `bool OsciDNSMethod::StartMethod (int32 aTimeoutMsec)` [protected]

## **7.117.4 Field Documentation**

**7.117.4.1** [Osci\\_DefAlloc](#)& OsciDNSMethod::iAlloc

**7.117.4.2** [TPVDNSFxn](#) OsciDNSMethod::iDNSFxn

**7.117.4.3** [OsciDNSObserver](#)\* OsciDNSMethod::iDNSObserver

**7.117.4.4** [OsciDNSRequestAO](#)\* OsciDNSMethod::iDNSRequestAO [protected]

**7.117.4.5** uint32 OsciDNSMethod::iId

**7.117.4.6** [PVLogger](#)\* OsciDNSMethod::iLogger

The documentation for this class was generated from the following file:

- [osci\\_dns\\_method.h](#)

## 7.118 OsciDNSObserver Class Reference

```
#include <osci_dns.h>
```

### Public Methods

- virtual OSCL\_IMPORT\_REF void [HandleDNSEvent](#) (int32 aId, [TPVDNSFxn](#) aFxn, [TPVDNSEvent](#) aEvent, int32 aError)=0
- virtual [~OsciDNSObserver](#) ()

### 7.118.1 Detailed Description

DNS event observer. The client implements this to get asynchronous command completion.

### 7.118.2 Constructor & Destructor Documentation

**7.118.2.1** virtual OsciDNSObserver::~~OsciDNSObserver () [inline, virtual]

### 7.118.3 Member Function Documentation

**7.118.3.1** virtual OSCL\_IMPORT\_REF void OsciDNSObserver::HandleDNSEvent (int32 aId, [TPVDNSFxn](#) aFxn, [TPVDNSEvent](#) aEvent, int32 aError) [pure virtual]

DNS Event callback.

#### Parameters:

- aId:** The ID that was supplied when the DNS object was created.
- aEvent:** Function completion event. Will be EPVDNSSuccess, EPVDNSTimeout, or EPVDNSFailure.
- aError:** When the event is EPVDNSFailure, this may contain a platform-specific error code, or zero if none is available.

The documentation for this class was generated from the following file:

- [osci\\_dns.h](#)

## 7.119 OsciDNSRequest Class Reference

```
#include <osci_dns_request.h>
```

### Public Methods

- [OsciDNSRequest \(\)](#)
- [~OsciDNSRequest \(\)](#)
- void [CancelRequest \(\)](#)
- void [Complete](#) (bool, int32 aStatus, int32 aSockErr)
- void [Activate](#) ([DNSRequestParam](#) \*iParam, [OsciDNSRequestAO](#) &a)

### Data Fields

- [OsciDNSRequestAO](#) \* iDNSRequestAO
- [DNSRequestParam](#) \* iDNSRequestParam
- bool iActive

### 7.119.1 Detailed Description

This class defines the interface to the dns implementation threads.

### 7.119.2 Constructor & Destructor Documentation

**7.119.2.1** [OsciDNSRequest::OsciDNSRequest \(\)](#) [inline]

**7.119.2.2** [OsciDNSRequest::~~OsciDNSRequest \(\)](#) [inline]

### 7.119.3 Member Function Documentation

**7.119.3.1** void [OsciDNSRequest::Activate](#) ([DNSRequestParam](#) \* iParam, [OsciDNSRequestAO](#) &a)

**7.119.3.2** void [OsciDNSRequest::CancelRequest \(\)](#)

**7.119.3.3** void [OsciDNSRequest::Complete](#) (bool, int32 aStatus, int32 aSockErr)

### 7.119.4 Field Documentation

**7.119.4.1** bool [OsciDNSRequest::iActive](#)

**7.119.4.2** [OsciDNSRequestAO](#)\* [OsciDNSRequest::iDNSRequestAO](#)

**7.119.4.3** [DNSRequestParam](#)\* [OsciDNSRequest::iDNSRequestParam](#)

The documentation for this class was generated from the following file:

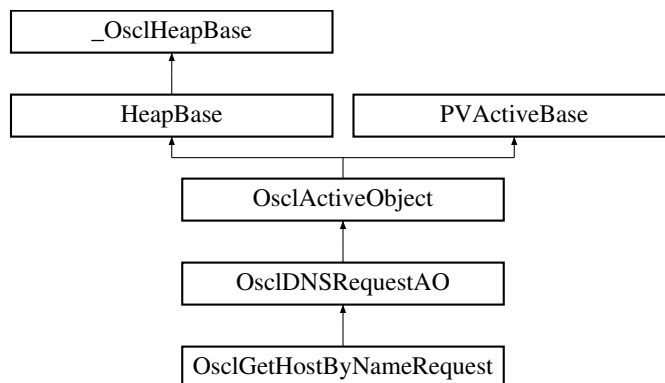
- [osci\\_dns\\_request.h](#)



## 7.120 OsciDNSRequestAO Class Reference

```
#include <osci_dns_method.h>
```

Inheritance diagram for OsciDNSRequestAO::



### Protected Methods

- [OsciDNSRequestAO](#) (const char \*name)
- void [ConstructL](#) ([OsciDNSI](#) \*aDNS, [OsciDNSMethod](#) \*aMethod)
- void [Abort](#) ()
- void [NewRequest](#) ()
- void [RequestDone](#) ()
- int [GetSocketError](#) ()
- [OsciSocketServI](#) \* [Serv](#) ()
- void [DoCancel](#) ()
- void [Run](#) ()
- virtual void [Success](#) ()
- virtual void [Failure](#) ()
- virtual void [Cancelled](#) ()

### Protected Attributes

- [OsciDNSI](#) \* [iDNSI](#)
- [OsciDNSMethod](#) \* [iDNSMethod](#)
- int32 [iSocketError](#)
- [PVLogger](#) \* [iLogger](#)

### Friends

- class [OsciDNSI](#)
- class [OsciDNSMethod](#)
- class [OsciDNSRequest](#)
- class [GetHostByNameParam](#)

## 7.120.1 Detailed Description

This is the base class for all requests to the socket server.

## 7.120.2 Constructor & Destructor Documentation

**7.120.2.1** `OsciDNSRequestAO::OsciDNSRequestAO (const char * name)` [inline, protected]

## 7.120.3 Member Function Documentation

**7.120.3.1** `void OsciDNSRequestAO::Abort ()` [inline, protected]

**7.120.3.2** `virtual void OsciDNSRequestAO::Cancelled ()` [inline, protected, virtual]

**7.120.3.3** `void OsciDNSRequestAO::ConstructL (OsciDNSI * aDNS, OsciDNSMethod * aMethod)` [inline, protected]

**7.120.3.4** `void OsciDNSRequestAO::DoCancel ()` [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Reimplemented from [OsciActiveObject](#).

**7.120.3.5** `virtual void OsciDNSRequestAO::Failure ()` [inline, protected, virtual]

**7.120.3.6** `int OsciDNSRequestAO::GetSocketError ()` [protected]

**7.120.3.7** `void OsciDNSRequestAO::NewRequest ()` [protected]

**7.120.3.8** `void OsciDNSRequestAO::RequestDone ()` [protected]

**7.120.3.9** `void OsciDNSRequestAO::Run ()` [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's [Run\(\)](#) function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

**7.120.3.10**   [OsciSocketServI](#)\* OsciDNSRequestAO::Serv ()   [protected]

**7.120.3.11**   virtual void OsciDNSRequestAO::Success ()   [inline, protected, virtual]

## 7.120.4 Friends And Related Function Documentation

**7.120.4.1**   friend class GetHostByNameParam   [friend]

**7.120.4.2**   friend class OsciDNSI   [friend]

**7.120.4.3**   friend class OsciDNSMethod   [friend]

**7.120.4.4**   friend class OsciDNSRequest   [friend]

## 7.120.5 Field Documentation

**7.120.5.1**   [OsciDNSI](#)\* OsciDNSRequestAO::iDNSI   [protected]

**7.120.5.2**   [OsciDNSMethod](#)\* OsciDNSRequestAO::iDNSMethod   [protected]

**7.120.5.3**   [PVLogger](#)\* OsciDNSRequestAO::iLogger   [protected]

**7.120.5.4**   int32 OsciDNSRequestAO::iSocketError   [protected]

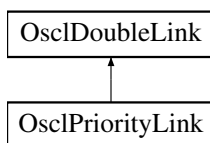
The documentation for this class was generated from the following file:

- [osci\\_dns\\_method.h](#)

## 7.121 OscDoubleLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OscDoubleLink::



### Public Methods

- [OscDoubleLink](#) ()
- void [Remove](#) ()
- void [InsertAfter](#) (OscDoubleLink \*aLink)
- void [InsertBefore](#) (OscDoubleLink \*aLink)

### Data Fields

- OscDoubleLink \* [iNext](#)
- OscDoubleLink \* [iPrev](#)

### 7.121.1 Constructor & Destructor Documentation

7.121.1.1 OscDoubleLink::OscDoubleLink () [inline]

### 7.121.2 Member Function Documentation

7.121.2.1 void OscDoubleLink::InsertAfter (OscDoubleLink \* *aLink*)

7.121.2.2 void OscDoubleLink::InsertBefore (OscDoubleLink \* *aLink*)

7.121.2.3 void OscDoubleLink::Remove ()

### 7.121.3 Field Documentation

7.121.3.1 OscDoubleLink\* OscDoubleLink::iNext

7.121.3.2 OscDoubleLink\* OscDoubleLink::iPrev

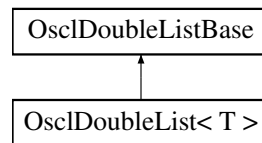
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.122 OsciDoubleList< T > Class Template Reference

```
#include <osci_double_list.h>
```

Inheritance diagram for OsciDoubleList< T >::



### Public Methods

- OSCI\_INLINE [OsciDoubleList](#) ()
- OSCI\_INLINE [OsciDoubleList](#) (int32 anOffset)
- OSCI\_INLINE void [InsertHead](#) (T &aRef)
- OSCI\_INLINE void [InsertTail](#) (T &aRef)
- OSCI\_INLINE bool [IsHead](#) (const T \*aPtr) const
- OSCI\_INLINE bool [IsTail](#) (const T \*aPtr) const
- OSCI\_INLINE T \* [Head](#) () const
- OSCI\_INLINE T \* [Tail](#) () const

```
template<class T> class OsciDoubleList< T >
```

### 7.122.1 Constructor & Destructor Documentation

7.122.1.1 `template<class T> OSCI_INLINE OsciDoubleList< T >::OsciDoubleList ()`

7.122.1.2 `template<class T> OSCI_INLINE OsciDoubleList< T >::OsciDoubleList (int32 anOffset)`

### 7.122.2 Member Function Documentation

7.122.2.1 `template<class T> OSCI_INLINE T* OsciDoubleList< T >::Head ()`

7.122.2.2 `template<class T> OSCI_INLINE void OsciDoubleList< T >::InsertHead (T &aRef)`

7.122.2.3 `template<class T> OSCI_INLINE void OsciDoubleList< T >::InsertTail (T &aRef)`

7.122.2.4 `template<class T> OSCI_INLINE bool OsciDoubleList< T >::IsHead (const T *aPtr) const`

7.122.2.5 `template<class T> OSCI_INLINE bool OsciDoubleList< T >::IsTail (const T *aPtr) const`

7.122.2.6 `template<class T> OSCI_INLINE T* OsciDoubleList< T >::Tail ()`

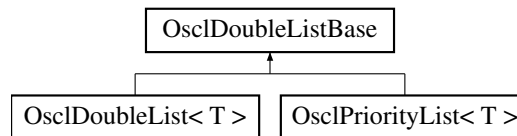
The documentation for this class was generated from the following file:

- [osci\\_double\\_list.h](#)

## 7.123 OsciDoubleListBase Class Reference

```
#include <osci_double_list.h>
```

Inheritance diagram for OsciDoubleListBase::



### Public Methods

- bool [IsEmpty](#) () const
- void [SetOffset](#) (int32 anOffset)
- void [Reset](#) ()
- [OsciDoubleLink](#) \* [getHead](#) ()
- int32 [getOffset](#) ()

### Protected Methods

- [OsciDoubleListBase](#) ()
- [OsciDoubleListBase](#) (int32 anOffset)
- void [InsertHead](#) ([OsciAny](#) \*aPtr)
- void [InsertTail](#) ([OsciAny](#) \*aPtr)
- void [Insert](#) ([OsciAny](#) \*aPtr)

### Protected Attributes

- [OsciDoubleLink](#) iHead
- int32 iOffset

### 7.123.1 Constructor & Destructor Documentation

7.123.1.1 `OsciDoubleListBase::OsciDoubleListBase ()` [protected]

7.123.1.2 `OsciDoubleListBase::OsciDoubleListBase (int32 anOffset)` [protected]

### 7.123.2 Member Function Documentation

7.123.2.1 `OsciDoubleLink* OsciDoubleListBase::getHead ()` [inline]

7.123.2.2 `int32 OsciDoubleListBase::getOffset ()` [inline]

7.123.2.3 `void OsciDoubleListBase::Insert (OsciAny * aPtr)` [protected]

7.123.2.4 `void OsciDoubleListBase::InsertHead (OsciAny * aPtr)` [protected]

7.123.2.5 `void OsciDoubleListBase::InsertTail (OsciAny * aPtr)` [protected]

7.123.2.6 `bool OsciDoubleListBase::IsEmpty ()`

7.123.2.7 `void OsciDoubleListBase::Reset ()`

7.123.2.8 `void OsciDoubleListBase::SetOffset (int32 anOffset)`

### 7.123.3 Field Documentation

7.123.3.1 `OsciDoubleLink OsciDoubleListBase::iHead` [protected]

7.123.3.2 `int32 OsciDoubleListBase::iOffset` [protected]

The documentation for this class was generated from the following file:

- [osci\\_double\\_list.h](#)

## 7.124 OsciDoubleRunner< T > Class Template Reference

```
#include <osci_double_list.h>
```

### Public Methods

- [OsciDoubleRunner](#) ([OsciDoubleListBase](#) &aQue)
- void [Set](#) (T &aLink)
- [operator T \\*](#) ()
- T \* [operator++](#) (int)
- T \* [operator--](#) (int)
- void [SetToHead](#) ()
- void [SetToTail](#) ()

### Protected Attributes

- int32 [iOffset](#)
- [OsciDoubleLink](#) \* [iHead](#)
- [OsciDoubleLink](#) \* [iNext](#)

```
template<class T> class OsciDoubleRunner< T >
```

### 7.124.1 Constructor & Destructor Documentation

**7.124.1.1** `template<class T> OsciDoubleRunner< T >::OsciDoubleRunner (OsciDoubleListBase &aQue) [inline]`

### 7.124.2 Member Function Documentation

**7.124.2.1** `template<class T> OsciDoubleRunner< T >::operator T * () [inline]`

**7.124.2.2** `template<class T> T* OsciDoubleRunner< T >::operator++ (int) [inline]`

**7.124.2.3** `template<class T> T* OsciDoubleRunner< T >::operator-- (int)`

**7.124.2.4** `template<class T> void OsciDoubleRunner< T >::Set (T &aLink) [inline]`

**7.124.2.5** `template<class T> void OsciDoubleRunner< T >::SetToHead () [inline]`

**7.124.2.6** `template<class T> void OsciDoubleRunner< T >::SetToTail () [inline]`

### 7.124.3 Field Documentation

**7.124.3.1** `template<class T> OsciDoubleLink* OsciDoubleRunner< T >::iHead [protected]`

**7.124.3.2** `template<class T> OsciDoubleLink* OsciDoubleRunner< T >::iNext [protected]`

**7.124.3.3** `template<class T> int32 OsciDoubleRunner< T >::iOffset [protected]`

The documentation for this class was generated from the following file:



- [osci\\_double\\_list.h](#)

## 7.125 OslError Class Reference

```
#include <osl_error.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [PushL](#) ([\\_OslHeapBase](#) \*aPtr)
- OSCL\_IMPORT\_REF void [PushL](#) ([OslAny](#) \*aPtr)
- OSCL\_IMPORT\_REF void [PushL](#) ([OslTrapItem](#) anItem)
- OSCL\_IMPORT\_REF void [Pop](#) ()
- OSCL\_IMPORT\_REF void [Pop](#) (int32 aCount)
- OSCL\_IMPORT\_REF void [PopDealloc](#) ()
- OSCL\_IMPORT\_REF void [PopDealloc](#) (int32 aCount)
- OSCL\_IMPORT\_REF void [Leave](#) (int32 aReason)
- OSCL\_IMPORT\_REF void [LeaveIfNull](#) ([OslAny](#) \*a)
- OSCL\_IMPORT\_REF void [LeaveIfError](#) (int32 aReason)

### 7.125.1 Detailed Description

User Error class

### 7.125.2 Member Function Documentation

#### 7.125.2.1 OSCL\_IMPORT\_REF void OslError::Leave (int32 *aReason*) [static]

Do a Leave error, with the given reason code. When a leave occurs, all items on the cleanup stack for the current trap level will be destroyed, and execution will jump to the trap handler.

#### 7.125.2.2 OSCL\_IMPORT\_REF void OslError::LeaveIfError (int32 *aReason*) [static]

Evaluate the input parameter, and if it is an error code (non-zero), then do a Leave with the provided reason code.

#### 7.125.2.3 OSCL\_IMPORT\_REF void OslError::LeaveIfNull ([OslAny](#) \* *a*) [static]

Evaluate the input parameter, and if it is null, do a Leave with OslErrNoMemory reason code.

#### 7.125.2.4 OSCL\_IMPORT\_REF void OslError::Pop (int32 *aCount*) [static]

Pop the cleanup stack N times

#### 7.125.2.5 OSCL\_IMPORT\_REF void OslError::Pop () [static]

Pop the cleanup stack

**7.125.2.6 OSCL\_IMPORT\_REF void OslError::PopDealloc (int32 *aCount*) [static]**

PopDealloc N times

**7.125.2.7 OSCL\_IMPORT\_REF void OslError::PopDealloc () [static]**

Destroy the item on the top of the cleanup stack and pop it

**7.125.2.8 OSCL\_IMPORT\_REF void OslError::PushL ([OslTrapItem](#) *anItem*) [static]**

Push an [OslTrapItem](#) onto the cleanup stack

**7.125.2.9 OSCL\_IMPORT\_REF void OslError::PushL ([OslAny](#) \* *aPtr*) [static]**

Push an OslAny item onto the cleanup stack.

**7.125.2.10 OSCL\_IMPORT\_REF void OslError::PushL ([\\_OslHeapBase](#) \* *aPtr*) [static]**

Push an [\\_OslHeapBase](#) item onto the cleanup stack.

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 7.126 OsciErrorAllocator Class Reference

This class provides static methods to invoke the user defined memory allocation routines.

```
#include <osci_error_allocator.h>
```

### Public Methods

- [OsciErrorAllocator](#) ([Osci\\_DefAlloc](#) \*allocator)  
*constructor method*
- void \* [operator new](#) (uint32 size, [OsciAny](#) \*aPtr)  
*placement new operator that allocates memory using the user defined methods*
- void [operator delete](#) ([OsciAny](#) \*aPtr, [OsciAny](#) \*aPtr2)  
*delete operator that doesn't do anything, user has to deallocate manually*

### Static Public Methods

- [OsciAny](#) \* [allocate](#) (uint32 aSize)  
*static method to allocate a block of memory on heap*
- [OsciAny](#) [deallocate](#) ([OsciAny](#) \*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 OsciErrorAllocator::OsciErrorAllocator ([Osci\\_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 [OsciAny](#)\* OsciErrorAllocator::allocate (uint32 aSize) [inline, static]

static method to allocate a block of memory on heap

#### Parameters:

*aSize* - number of bytes to allocate

**7.126.3.2** [OsciAny](#) OsciErrorAllocator::deallocate ([OsciAny](#) \* *aPointer*) [inline, static]

static method to deallocate a block of memory on heap

**Parameters:**

*aPointer* - pointer to block of memory to be deallocated

**7.126.3.3** void OsciErrorAllocator::operator delete ([OsciAny](#) \* *aPtr*, [OsciAny](#) \* *aPtr2*)  
[inline]

delete operator that doesn't do anything, user has to deallocate manually

**7.126.3.4** void\* OsciErrorAllocator::operator new (uint32 *size*, [OsciAny](#) \* *aPtr*) [inline]

placement new operator that allocates memory using the user defined methods

The documentation for this class was generated from the following file:

- [osci\\_error\\_allocator.h](#)

## 7.127 OsciErrorTrap Class Reference

```
#include <osci_error.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF int32 [Init](#) ([Osci\\_DefAlloc](#) \*aAlloc=NULL)
- OSCL\_IMPORT\_REF int32 [Cleanup](#) ()
- OSCL\_IMPORT\_REF [OsciErrorTrapImp](#) \* [GetErrorTrapImp](#) ()

### 7.127.1 Member Function Documentation

#### 7.127.1.1 OSCL\_IMPORT\_REF int32 OsciErrorTrap::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 [OsciErrorTrapImp](#)\* OsciErrorTrap::GetErrorTrapImp () [static]

Get the ErrorTrapImp for the current thread. Leaves on error.

#### 7.127.1.3 OSCL\_IMPORT\_REF int32 OsciErrorTrap::Init ([Osci\\_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:

- [osci\\_error.h](#)

## 7.128 OslErrorTrapImp Class Reference

```
#include <osl_error_trapcleanup.h>
```

### Public Methods

- OSL\_IMPORT\_REF void [UnTrap](#) ()

### Static Public Methods

- OSL\_IMPORT\_REF OslErrorTrapImp \* [Trap](#) ()
- OSL\_IMPORT\_REF OslErrorTrapImp \* [TrapNoTls](#) (OslErrorTrapImp \*)

### Data Fields

- [OslJump](#) \* [iJumpData](#)
- int32 [iLeave](#)
- [OslTrapStack](#) \* [iTrapStack](#)

### Friends

- class [OslErrorTrap](#)
- class [OslError](#)
- class [OslExecScheduler](#)
- class [OslExecSchedulerCommonBase](#)
- class [OslJump](#)
- class [OslJumpMark](#)
- class [OslTrapStack](#)
- class [CPVInterfaceProxy](#)
- class [OslScheduler](#)

### 7.128.1 Detailed Description

A per-thread cleanup stack with nested trap support.

### 7.128.2 Member Function Documentation

#### 7.128.2.1 OSL\_IMPORT\_REF OslErrorTrapImp\* OslErrorTrapImp::Trap () [static]

PV trap cleanup. Public for use in macros only.

#### 7.128.2.2 OSL\_IMPORT\_REF OslErrorTrapImp\* OslErrorTrapImp::TrapNoTls (OslErrorTrapImp \*) [static]

#### 7.128.2.3 OSL\_IMPORT\_REF void OslErrorTrapImp::UnTrap ()

these are used in public macros, but aren't intended as public methods or members.

### 7.128.3 Friends And Related Function Documentation

7.128.3.1 friend class CPVInterfaceProxy [friend]

7.128.3.2 friend class OsciError [friend]

7.128.3.3 friend class OsciErrorTrap [friend]

7.128.3.4 friend class OsciExecScheduler [friend]

7.128.3.5 friend class OsciExecSchedulerCommonBase [friend]

7.128.3.6 friend class OsciJump [friend]

7.128.3.7 friend class OsciJumpMark [friend]

7.128.3.8 friend class OsciScheduler [friend]

7.128.3.9 friend class OsciTrapStack [friend]

### 7.128.4 Field Documentation

7.128.4.1 [OsciJump\\*](#) OsciErrorTrapImp::iJumpData

7.128.4.2 int32 OsciErrorTrapImp::iLeave

7.128.4.3 [OsciTrapStack\\*](#) OsciErrorTrapImp::iTrapStack

The documentation for this class was generated from the following file:

- [osci\\_error\\_trapcleanup.h](#)



## 7.129 OslException< LeaveCode > Class Template Reference

[oscl\\_exception.h](#) contains all the exception handling macros and classes. This template class provides the base exception class that all exceptions derive from.

```
#include <oscl_exception.h>
```

### Public Methods

- [OslException\(\)](#)

### Static Public Methods

- `int` [getLeaveCode\(\)](#)

### 7.129.1 Detailed Description

**template<int LeaveCode> class OslException< LeaveCode >**

[oscl\\_exception.h](#) contains all the exception handling macros and classes. This template class provides the base exception class that all exceptions derive from.

All PacketVideo exception classes will be derived from the OslException class. Each derived class will have a static function where the leave code can be obtained. This avoids the issue of having static members in a DLL. The function needs to be static so it can be called without an instance of the class.

### 7.129.2 Constructor & Destructor Documentation

**7.129.2.1** **template<int LeaveCode> OslException< LeaveCode >::OslException()**  
[inline]

### 7.129.3 Member Function Documentation

**7.129.3.1** **template<int LeaveCode> int OslException< LeaveCode >::getLeaveCode()**  
[inline, static]

The documentation for this class was generated from the following file:

- [oscl\\_exception.h](#)

## 7.130 OsciExclusiveArrayPtr< T > Class Template Reference

The OsciExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsciExclusiveArrayPtr expires, its destructor uses delete to free the memory.

```
#include <osci_exclusive_ptr.h>
```

### Public Methods

- [OsciExclusiveArrayPtr](#) (T \*inPtr=0)  
*Default constructor Initializes the pointer and takes ownership.*
- [OsciExclusiveArrayPtr](#) (OsciExclusiveArrayPtr< T > &\_Y)  
*Copy constructor.*
- OsciExclusiveArrayPtr< T > & [operator=](#) (OsciExclusiveArrayPtr< T > &\_Y)  
*Assignment operator from an another OsciExclusiveArrayPtr.*
- virtual [~OsciExclusiveArrayPtr](#) ()  
*Destructor.*
- T & [operator \\*](#) () const  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- T \* [operator →](#) () const  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- T \* [get](#) () const  
*get() method returns the pointer, currently owned by the class.*
- T \* [release](#) ()  
*release() method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool [set](#) (T \*ptr)  
*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- T \* [\\_Ptr](#)

#### 7.130.1 Detailed Description

```
template<class T> class OsciExclusiveArrayPtr< T >
```

The OsciExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsciExclusiveArrayPtr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an [OsciExclusivePtr](#) object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The [OsciExclusivePtr](#) is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

## 7.130.2 Constructor & Destructor Documentation

**7.130.2.1** `template<class T> OsciExclusiveArrayPtr< T >::OsciExclusiveArrayPtr (T * inPtr = 0) [inline, explicit]`

Default constructor Initializes the pointer and takes ownership.

**7.130.2.2** `template<class T> OsciExclusiveArrayPtr< T >::OsciExclusiveArrayPtr (OsciExclusiveArrayPtr< T > &_Y) [inline]`

Copy constructor.

Initializes the pointer and takes ownership from another OsciExclusiveArrayPtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

**7.130.2.3** `template<class T> virtual OsciExclusiveArrayPtr< T >::~~OsciExclusiveArrayPtr () [inline, virtual]`

Destructor.

The pointer is deleted in case this class still has ownership

## 7.130.3 Member Function Documentation

**7.130.3.1** `template<class T> T* OsciExclusiveArrayPtr< T >::get () const [inline]`

[get\(\)](#) method returns the pointer, currently owned by the class.

**7.130.3.2** `template<class T> T& OsciExclusiveArrayPtr< T >::operator * () const [inline]`

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsciExclusiveArrayPtr can be used like the regular pointer that it was initialized with.

**7.130.3.3** `template<class T> T* OsciExclusiveArrayPtr< T >::operator → () const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsciExclusiveArrayPtr can be used like the regular pointer that it was initialized with.

**7.130.3.4** `template<class T> OsciExclusiveArrayPtr<T>& OsciExclusiveArrayPtr< T >::operator= (OsciExclusiveArrayPtr< T > &_Y) [inline]`

Assignment operator from an another OsciExclusiveArrayPtr.

**Parameters:**

\_Y The value parameter should be another OsciExclusiveArrayPtr

**Returns:**

Returns a reference to this OsciExclusiveArrayPtr instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the OsciExclusiveArrayPtr given as the input parameter. The ownership of the pointer is transferred.

**7.130.3.5** `template<class T> T* OsciExclusiveArrayPtr< T >::release () [inline]`

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**7.130.3.6** `template<class T> bool OsciExclusiveArrayPtr< T >::set (T *ptr) [inline]`

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

## 7.130.4 Field Documentation

**7.130.4.1** `template<class T> T* OsciExclusiveArrayPtr< T >::_Ptr [protected]`

The documentation for this class was generated from the following file:

- [osci\\_exclusive\\_ptr.h](#)

## 7.131 OsciExclusivePtr< T > Class Template Reference

The OsciExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsciExclusivePtr expires, its destructor uses delete to free the memory.

```
#include <osci_exclusive_ptr.h>
```

### Public Methods

- [OsciExclusivePtr](#) (T \*inPtr=0)  
*Default constructor Initializes the pointer and takes ownership.*
- [OsciExclusivePtr](#) (OsciExclusivePtr< T > &\_Y)  
*Copy constructor.*
- OsciExclusivePtr< T > & [operator=](#) (OsciExclusivePtr< T > &\_Y)  
*Assignment operator from an another OsciExclusivePtr.*
- virtual [~OsciExclusivePtr](#) ()  
*Destructor.*
- T & [operator \\*](#) () const  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- T \* [operator →](#) () const  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- T \* [get](#) () const  
*get() method returns the pointer, currently owned by the class.*
- T \* [release](#) ()  
*release() method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool [set](#) (T \*ptr)  
*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- T \* [\\_Ptr](#)

#### 7.131.1 Detailed Description

```
template<class T> class OsciExclusivePtr< T >
```

The OsciExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsciExclusivePtr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an OsciExclusivePtr object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The OsciExclusivePtr is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

## 7.131.2 Constructor & Destructor Documentation

**7.131.2.1** `template<class T> OsciExclusivePtr< T >::OsciExclusivePtr (T * inPtr = 0)`  
[inline, explicit]

Default constructor Initializes the pointer and takes ownership.

**7.131.2.2** `template<class T> OsciExclusivePtr< T >::OsciExclusivePtr (OsciExclusivePtr< T > &_Y)` [inline]

Copy constructor.

Initializes the pointer and takes ownership from another OsciExclusivePtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

**7.131.2.3** `template<class T> virtual OsciExclusivePtr< T >::~~OsciExclusivePtr ()` [inline, virtual]

Destructor.

The pointer is deleted in case this class still has ownership

## 7.131.3 Member Function Documentation

**7.131.3.1** `template<class T> T* OsciExclusivePtr< T >::get () const` [inline]

[get\(\)](#) method returns the pointer, currently owned by the class.

**7.131.3.2** `template<class T> T& OsciExclusivePtr< T >::operator * () const` [inline]

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsciExclusivePtr can be used like the regular pointer that it was initialized with.

**7.131.3.3** `template<class T> T* OsciExclusivePtr< T >::operator -> () const` [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsciExclusivePtr can be used like the regular pointer that it was initialized with.

**7.131.3.4** `template<class T> OsciExclusivePtr<T>& OsciExclusivePtr< T >::operator=(OsciExclusivePtr< T > & _Y) [inline]`

Assignment operator from an another OsciExclusivePtr.

**Parameters:**

`_Y` The value parameter should be another OsciExclusivePtr

**Returns:**

Returns a reference to this OsciExclusivePtr instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the OsciExclusivePtr given as the input parameter. The ownership of the pointer is transferred.

**7.131.3.5** `template<class T> T* OsciExclusivePtr< T >::release () [inline]`

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**7.131.3.6** `template<class T> bool OsciExclusivePtr< T >::set (T * ptr) [inline]`

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

## 7.131.4 Field Documentation

**7.131.4.1** `template<class T> T* OsciExclusivePtr< T >::_Ptr [protected]`

The documentation for this class was generated from the following file:

- [osci\\_exclusive\\_ptr.h](#)

## 7.132 OsciExclusivePtrA< T, Alloc > Class Template Reference

The OsciExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or indirectly) through Alloc. When the OsciExclusivePtrA expires, Alloc is used to free the memory.

```
#include <osci_exclusive_ptr.h>
```

### Public Methods

- [OsciExclusivePtrA](#) (T \*inPtr=0)  
*Default constructor Initializes the pointer and takes ownership.*
- [OsciExclusivePtrA](#) (OsciExclusivePtrA< T, Alloc > &\_Y)  
*Copy constructor.*
- OsciExclusivePtrA< T, Alloc > & [operator=](#) (OsciExclusivePtrA< T, Alloc > &\_Y)  
*Assignment operator from an another [OsciExclusiveArrayPtr](#).*
- virtual [~OsciExclusivePtrA](#) ()  
*Destructor.*
- T & [operator \\*](#) () const  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- T \* [operator →](#) () const  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- T \* [get](#) () const  
*[get\(\)](#) method returns the pointer, currently owned by the class.*
- T \* [release](#) ()  
*[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool [set](#) (T \*ptr)  
*[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- T \* [\\_Ptr](#)

#### 7.132.1 Detailed Description

```
template<class T, class Alloc> class OsciExclusivePtrA< T, Alloc >
```

The OsciExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or indirectly) through Alloc. When the OsciExclusivePtrA expires, Alloc is used to free the memory.



The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an [OsciExclusivePtr](#) object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The [OsciExclusivePtr](#) is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

## 7.132.2 Constructor & Destructor Documentation

**7.132.2.1** `template<class T, class Alloc> OsciExclusivePtrA< T, Alloc >::OsciExclusivePtrA (T * inPtr = 0) [inline, explicit]`

Default constructor Initializes the pointer and takes ownership.

**7.132.2.2** `template<class T, class Alloc> OsciExclusivePtrA< T, Alloc >::OsciExclusivePtrA (OsciExclusivePtrA< T, Alloc > & _Y) [inline]`

Copy constructor.

Initializes the pointer and takes ownership from another [OsciExclusiveArrayPtr](#). Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

**7.132.2.3** `template<class T, class Alloc> virtual OsciExclusivePtrA< T, Alloc >::~~OsciExclusivePtrA () [inline, virtual]`

Destructor.

The pointer is deleted in case this class still has ownership

## 7.132.3 Member Function Documentation

**7.132.3.1** `template<class T, class Alloc> T* OsciExclusivePtrA< T, Alloc >::get () const [inline]`

[get\(\)](#) method returns the pointer, currently owned by the class.

**7.132.3.2** `template<class T, class Alloc> T& OsciExclusivePtrA< T, Alloc >::operator * () const [inline]`

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the [OsciExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

**7.132.3.3** `template<class T, class Alloc> T* OsciExclusivePtrA< T, Alloc >::operator -> () const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the [OsciExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

**7.132.3.4** `template<class T, class Alloc> OsciExclusivePtrA<T, Alloc> & OsciExclusivePtrA< T, Alloc >::operator= (OsciExclusivePtrA< T, Alloc > & _Y) [inline]`

Assignment operator from an another [OsciExclusiveArrayPtr](#).

**Parameters:**

`_Y` The value parameter should be another [OsciExclusiveArrayPtr](#)

**Returns:**

Returns a reference to this [OsciExclusiveArrayPtr](#) instance with pointer initialized.

**Precondition:**

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the [OsciExclusiveArrayPtr](#) given as the input parameter. The ownership of the pointer is transferred.

**7.132.3.5** `template<class T, class Alloc> T* OsciExclusivePtrA< T, Alloc >::release () [inline]`

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**7.132.3.6** `template<class T, class Alloc> bool OsciExclusivePtrA< T, Alloc >::set (T * ptr) [inline]`

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

## 7.132.4 Field Documentation

**7.132.4.1** `template<class T, class Alloc> T* OsciExclusivePtrA< T, Alloc >::_Ptr [protected]`

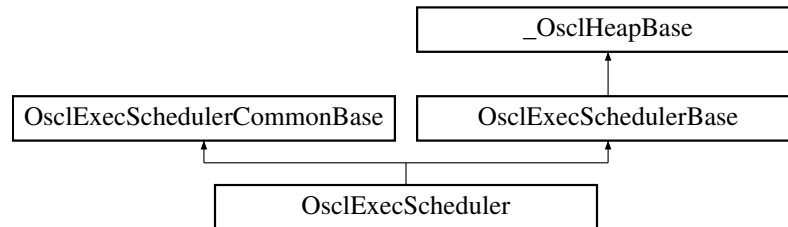
The documentation for this class was generated from the following file:

- [osci\\_exclusive\\_ptr.h](#)

## 7.133 OsciExecScheduler Class Reference

```
#include <osci_scheduler.h>
```

Inheritance diagram for OsciExecScheduler::



### Public Methods

- OSCI\_IMPORT\_REF void [RunSchedulerNonBlocking](#) (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)
- OSCI\_IMPORT\_REF void [RegisterForCallback](#) ([OsciSchedulerObserver](#) \*aCallback, [OsciAny](#) \*a-CallbackContext)

### Static Public Methods

- OSCI\_IMPORT\_REF [OsciExecScheduler](#) \* [Current](#) ()

### Friends

- class [OsciScheduler](#)

### 7.133.1 Member Function Documentation

#### 7.133.1.1 OSCI\_IMPORT\_REF [OsciExecScheduler](#)\* [OsciExecScheduler::Current](#) () [static]

Get currently installed scheduler for calling thread, or NULL if no scheduler is installed.

#### 7.133.1.2 OSCI\_IMPORT\_REF void [OsciExecScheduler::RegisterForCallback](#) ([OsciSchedulerObserver](#) \* aCallback, [OsciAny](#) \* aCallbackContext)

Register for a notification when non-blocking scheduler needs to run again.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

#### 7.133.1.3 OSCI\_IMPORT\_REF void [OsciExecScheduler::RunSchedulerNonBlocking](#) (int32 aTargetCount, int32 & aReady, uint32 & aDelayMsec)

Run PV scheduler in non-blocking mode. This call returns when the desired number of Run calls have been made, or when there are no more active objects that are ready to run.

**Parameters:**

***aTargetCount:*** (input param) the maximum number of Run calls to make.

***aReady:*** (output param) tells the number of active objects that are currently ready to run.

***aDelayMsec:*** (output param) If no active objects are ready to run, but one or more active objects are waiting on timers, this parameter will tell the time interval from the current time until the first of the pending timer objects will be ready to run, in milliseconds.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

## 7.133.2 Friends And Related Function Documentation

### 7.133.2.1 friend class OslScheduler [friend]

Reimplemented from [OslExecSchedulerCommonBase](#).

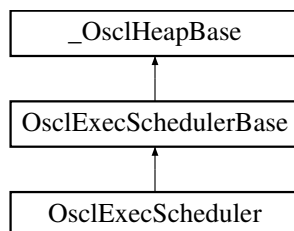
The documentation for this class was generated from the following file:

- [oscl\\_scheduler.h](#)

## 7.134 OsciExecSchedulerBase Class Reference

```
#include <osci_scheduler_types.h>
```

Inheritance diagram for OsciExecSchedulerBase::



### Friends

- class [OsciExecScheduler](#)
- class [OsciCoeActiveScheduler](#)
- class [PVActiveBase](#)

### 7.134.1 Detailed Description

OsciActiveSchedulerBase is the base for [OsciExecScheduler](#). The non-Symbian OsciActiveSchedulerBase class is functionally similar to a subset of Symbian CActiveScheduler class.

### 7.134.2 Friends And Related Function Documentation

**7.134.2.1 friend class OsciCoeActiveScheduler** [friend]

**7.134.2.2 friend class OsciExecScheduler** [friend]

**7.134.2.3 friend class PVActiveBase** [friend]

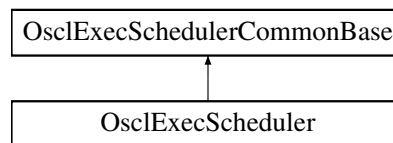
The documentation for this class was generated from the following file:

- [osci\\_scheduler\\_types.h](#)

## 7.135 OsciExecSchedulerCommonBase Class Reference

```
#include <osci_scheduler.h>
```

Inheritance diagram for OsciExecSchedulerCommonBase::



### Public Methods

- OSCL\_IMPORT\_REF void [StartScheduler](#) (OsciSemaphore \*sem=NULL)
- OSCL\_IMPORT\_REF void [StopScheduler](#) ()
- OSCL\_IMPORT\_REF void [SuspendScheduler](#) ()
- OSCL\_IMPORT\_REF void [ResumeScheduler](#) ()
- OSCL\_IMPORT\_REF void [StartNativeScheduler](#) ()

### Static Public Methods

- OSCL\_IMPORT\_REF [OsciNameString](#)< PVSCHEDNAMELEN > \* [GetName](#) ()
- OSCL\_IMPORT\_REF uint32 [GetId](#) ()

### Protected Types

- enum [TOtherExecStats](#) { [EOtherExecStats\\_WaitTime](#), [EOtherExecStats\\_QueueTime](#), [EOtherExecStats\\_NativeOS](#), [EOtherExecStats\\_ReleaseTime](#), [EOtherExecStats\\_Last](#) }

### Protected Methods

- virtual [~OsciExecSchedulerCommonBase](#) ()
- void [InstallScheduler](#) ()
- void [UninstallScheduler](#) ()
- void [Error](#) (int32 anError) const
- [OsciExecSchedulerCommonBase](#) ([Osci\\_DefAlloc](#) \*)
- virtual void [ConstructL](#) (const char \*name, int)
- void [BeginScheduling](#) (bool blocking, bool native)
- void [EndScheduling](#) ()
- void [BlockingLoopL](#) ()
- bool [IsStarted](#) ()
- bool [IsInstalled](#) ()
- void [AddToExecTimerQ](#) (PActiveBase \*active, uint32)
- void [PendComplete](#) (PActiveBase \*, int32 aReason, [TPVThreadContext](#) aContext)
- void [RequestCanceled](#) (PActiveBase \*)
- PActiveBase \* [UpdateTimers](#) (uint32 &aDelay)
- PActiveBase \* [UpdateTimersMsec](#) (uint32 &aDelay)
- PActiveBase \* [WaitForReadyAO](#) ()

- void [CallRunExec](#) (PActiveBase \*)
- void [ConstructStatQ](#) ()
- void [BeginStats](#) ()
- void [EndStats](#) ()
- void [CleanupStatQ](#) ()
- PActiveBase \* [FindPVBase](#) (PActiveBase \*active, OsciDoubleList< PActiveBase > &)
- void [CleanupExecQ](#) ()
- void [InitExecQ](#) (int)
- void [ResetLogPerf](#) ()
- void [IncLogPerf](#) (uint32)

## Static Protected Methods

- OsciExecSchedulerCommonBase \* [GetScheduler](#) ()
- OsciExecSchedulerCommonBase \* [SetScheduler](#) (OsciExecSchedulerCommonBase \*)
- void [ShowStats](#) (PActiveStats \*active)
- void [ShowSummaryStats](#) (PActiveStats \*active, PVLogger \*, int64, int64 &, float &)

## Protected Attributes

- bool [iBlockingMode](#)
- bool [iNativeMode](#)
- PVSchedulerStopper \* [iStopper](#)
- OsciNoYieldMutex [iStopperCrit](#)
- PVThreadContext [iThreadContext](#)
- OsciNameString< PVSCHEDNAMELEN > [iName](#)
- bool [iDoStop](#)
- bool [iDoSuspend](#)
- bool [iSuspended](#)
- OsciSemaphore [iResumeSem](#)
- OsciErrorTrapImp \* [iErrorTrapImp](#)
- OsciReadyQ [iReadyQ](#)
- OsciTimerQ [iExecTimerQ](#)
- uint32 [iNumAOAdded](#)
- OsciDoubleList< PActiveStats > [iPVStatQ](#)
- PActiveStats \* [iOtherExecStats](#) [EOtherExecStats\_Last]
- uint8 \* [iTotalTicksTemp](#)
- int64 [iGrandTotalTicks](#)
- float [iTotalPercent](#)
- uint32 [iTime](#)
- int32 [iDelta](#)
- PActiveStats \* [iPVStats](#)
- PVLogger \* [iLogger](#)
- PVLogger \* [iDebugLogger](#)
- char \* [iLogPerfIndentStr](#)
- int32 [iLogPerfIndentStrLen](#)
- uint32 [iLogPerfTotal](#)
- Osci\_DefAlloc \* [iAlloc](#)
- OsciMemAllocator [iDefAlloc](#)

## Static Protected Attributes

- const uint32 [iTimeCompareThreshold](#)

## Friends

- class [OsciScheduler](#)
- class [PVThreadContext](#)
- class [OsciCoeActiveScheduler](#)
- class [OsciTimerCompare](#)
- class [OsciReadyQ](#)
- class [OsciError](#)
- class [PVActiveStats](#)
- class [OsciActiveObject](#)
- class [OsciTimerObject](#)
- class [PVActiveBase](#)
- class [PVSchedulerStopper](#)
- class [OsciExecScheduler](#)

## 7.135.1 Member Enumeration Documentation

### 7.135.1.1 enum OsciExecSchedulerCommonBase::TOtherExecStats [protected]

Enumeration values:

**EOtherExecStats\_WaitTime**  
**EOtherExecStats\_QueueTime**  
**EOtherExecStats\_NativeOS**  
**EOtherExecStats\_ReleaseTime**  
**EOtherExecStats\_Last**



## 7.135.2 Constructor & Destructor Documentation

- 7.135.2.1 `virtual OsciExecSchedulerCommonBase::~~OsciExecSchedulerCommonBase ()`  
[protected, virtual]
- 7.135.2.2 `OsciExecSchedulerCommonBase::OsciExecSchedulerCommonBase (Osci_DefAlloc *)`  
[protected]

## 7.135.3 Member Function Documentation

- 7.135.3.1 `void OsciExecSchedulerCommonBase::AddToExecTimerQ (PActiveBase * active, uint32)` [protected]
- 7.135.3.2 `void OsciExecSchedulerCommonBase::BeginScheduling (bool blocking, bool native)`  
[protected]
- 7.135.3.3 `void OsciExecSchedulerCommonBase::BeginStats ()` [protected]
- 7.135.3.4 `void OsciExecSchedulerCommonBase::BlockingLoopL ()` [protected]
- 7.135.3.5 `void OsciExecSchedulerCommonBase::CallRunExec (PActiveBase *)` [protected]
- 7.135.3.6 `void OsciExecSchedulerCommonBase::CleanupExecQ ()` [protected]
- 7.135.3.7 `void OsciExecSchedulerCommonBase::CleanupStatQ ()` [protected]
- 7.135.3.8 `virtual void OsciExecSchedulerCommonBase::ConstructL (const char * name, int)`  
[protected, virtual]
- 7.135.3.9 `void OsciExecSchedulerCommonBase::ConstructStatQ ()` [protected]
- 7.135.3.10 `void OsciExecSchedulerCommonBase::EndScheduling ()` [protected]
- 7.135.3.11 `void OsciExecSchedulerCommonBase::EndStats ()` [protected]
- 7.135.3.12 `void OsciExecSchedulerCommonBase::Error (int32 anError) const` [protected]
- 7.135.3.13 `PActiveBase* OsciExecSchedulerCommonBase::FindPVBase (PActiveBase * active, OsciDoubleList< PActiveBase > &)` [protected]
- 7.135.3.14 `OSCL_IMPORT_REF uint32 OsciExecSchedulerCommonBase::GetId ()` [static]

Get numeric ID of current thread.

- 7.135.3.15 `OSCL_IMPORT_REF OsciNameString<PVSCHEDNAMELEN>*`  
`OsciExecSchedulerCommonBase::GetName ()` [static]

Get name of scheduler for current thread.

- 7.135.3.16 **OsciExecSchedulerCommonBase\* OsciExecSchedulerCommonBase::GetScheduler ()**  
[static, protected]
- 7.135.3.17 **void OsciExecSchedulerCommonBase::IncLogPerf (uint32)** [protected]
- 7.135.3.18 **void OsciExecSchedulerCommonBase::InitExecQ (int)** [protected]
- 7.135.3.19 **void OsciExecSchedulerCommonBase::InstallScheduler ()** [protected]
- 7.135.3.20 **bool OsciExecSchedulerCommonBase::IsInstalled ()** [inline, protected]
- 7.135.3.21 **bool OsciExecSchedulerCommonBase::IsStarted ()** [protected]
- 7.135.3.22 **void OsciExecSchedulerCommonBase::PendComplete (PVAActiveBase \*, int32 aReason, TPVThreadContext aContext)** [protected]
- 7.135.3.23 **void OsciExecSchedulerCommonBase::RequestCanceled (PVAActiveBase \*)**  
[protected]
- 7.135.3.24 **void OsciExecSchedulerCommonBase::ResetLogPerf ()** [protected]
- 7.135.3.25 **OSCL\_IMPORT\_REF void OsciExecSchedulerCommonBase::ResumeScheduler ()**

Resume scheduling immediately. This API only applies to a blocking loop scheduler.

- 7.135.3.26 **OsciExecSchedulerCommonBase\* OsciExecSchedulerCommonBase::SetScheduler (OsciExecSchedulerCommonBase \*)** [static, protected]
- 7.135.3.27 **void OsciExecSchedulerCommonBase::ShowStats (PVAActiveStats \* active)** [static, protected]
- 7.135.3.28 **void OsciExecSchedulerCommonBase::ShowSummaryStats (PVAActiveStats \* active, PVLogger \*, int64, int64 &, float &)** [static, protected]
- 7.135.3.29 **OSCL\_IMPORT\_REF void OsciExecSchedulerCommonBase::StartNativeScheduler ()**

Start the OS native scheduling loop. This is an alternative to the PV scheduling loop. To stop the native scheduler, use the StopScheduler API.

- 7.135.3.30 **OSCL\_IMPORT\_REF void OsciExecSchedulerCommonBase::StartScheduler (OsciSemaphore \* sem = NULL)**

Start scheduling. This call blocks until scheduler is stopped or an error occurs.

#### Parameters:

**sem:** optional startup semaphore. If provided, the scheduler will signal this semaphore when the startup has progressed to the point that it's safe to call StopScheduler or SuspendScheduler from another thread.

### 7.135.3.31 OSCL\_IMPORT\_REF void OsciExecSchedulerCommonBase::StopScheduler ()

Stop scheduling. This API may be called from the scheduling thread or some other thread.

### 7.135.3.32 OSCL\_IMPORT\_REF void OsciExecSchedulerCommonBase::SuspendScheduler ()

Suspend scheduling when the current Run is complete. This API only applies to a blocking loop scheduler.

### 7.135.3.33 void OsciExecSchedulerCommonBase::UninstallScheduler () [protected]

### 7.135.3.34 PVActiveBase\* OsciExecSchedulerCommonBase::UpdateTimers (uint32 & *aDelay*) [protected]

### 7.135.3.35 PVActiveBase\* OsciExecSchedulerCommonBase::UpdateTimersMsec (uint32 & *aDelay*) [protected]

### 7.135.3.36 PVActiveBase\* OsciExecSchedulerCommonBase::WaitForReadyAO () [protected]

## 7.135.4 Friends And Related Function Documentation

### 7.135.4.1 friend class OsciActiveObject [friend]

### 7.135.4.2 friend class OsciCoeActiveScheduler [friend]

### 7.135.4.3 friend class OsciError [friend]

### 7.135.4.4 friend class OsciExecScheduler [friend]

### 7.135.4.5 friend class OsciReadyQ [friend]

### 7.135.4.6 friend class OsciScheduler [friend]

Reimplemented in [OsciExecScheduler](#).



7.135.4.7 friend class OsciTimerCompare [friend]

7.135.4.8 friend class OsciTimerObject [friend]

7.135.4.9 friend class PVActiveBase [friend]

7.135.4.10 friend class PVActiveStats [friend]

7.135.4.11 friend class PVSchedulerStopper [friend]

7.135.4.12 friend class PVThreadContext [friend]

## 7.135.5 Field Documentation

7.135.5.1 [Osci\\_DefAlloc\\*](#) OsciExecSchedulerCommonBase::iAlloc [protected]

7.135.5.2 bool OsciExecSchedulerCommonBase::iBlockingMode [protected]

7.135.5.3 [PVLogger\\*](#) OsciExecSchedulerCommonBase::iDebugLogger [protected]

7.135.5.4 [OsciMemAllocator](#) OsciExecSchedulerCommonBase::iDefAlloc [protected]

7.135.5.5 int32 OsciExecSchedulerCommonBase::iDelta [protected]

7.135.5.6 bool OsciExecSchedulerCommonBase::iDoStop [protected]

7.135.5.7 bool OsciExecSchedulerCommonBase::iDoSuspend [protected]

7.135.5.8 [OsciErrorTrapImp\\*](#) OsciExecSchedulerCommonBase::iErrorTrapImp  
[protected]

7.135.5.9 [OsciTimerQ](#) OsciExecSchedulerCommonBase::iExecTimerQ [protected]

7.135.5.10 [int64](#) OsciExecSchedulerCommonBase::iGrandTotalTicks [protected]

7.135.5.11 [PVLogger\\*](#) OsciExecSchedulerCommonBase::iLogger [protected]

7.135.5.12 char\* OsciExecSchedulerCommonBase::iLogPerfIndentStr [protected]

7.135.5.13 int32 OsciExecSchedulerCommonBase::iLogPerfIndentStrLen [protected]

7.135.5.14 uint32 OsciExecSchedulerCommonBase::iLogPerfTotal [protected]

7.135.5.15 [OsciNameString](#)<PVSCHEDNAMELEN> OsciExecSchedulerCommonBase::iName  
[protected]

7.135.5.16 bool OsciExecSchedulerCommonBase::iNativeMode [protected]

7.135.5.17 uint32 OsciExecSchedulerCommonBase::iNumAOAdded [protected]

7.135.5.18 [PVActiveStats\\*](#) OsciExecSchedulerCommonBase::iOtherExecStats[EOtherExecStats\_  
Last] [protected]

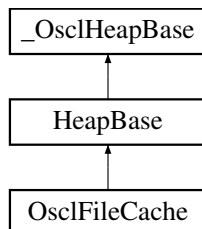
7.135.5.19 [OsciDoubleList](#)<[PVActiveStats](#)> OsciExecSchedulerCommonBase::iPVStatQ  
[protected]

- [osci\\_scheduler.h](#)

## 7.136 OslFileCache Class Reference

```
#include <osl_file_cache.h>
```

Inheritance diagram for OslFileCache::



### Public Methods

- [OslFileCache](#) ([Osl\\_File](#) &aContainer)
- [~OslFileCache](#) ()
- [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](#))
- [TOslFileOffset FileSize](#) ()
- [int32 Seek](#) ([TOslFileOffset offset](#), [Osl\\_File::seek\\_type origin](#))
- [TOslFileOffset Tell](#) ()
- [int32 Flush](#) ()
- [int32 EndOfFile](#) ()
- [OSCL\\_IMPORT\\_REF OslFileCacheBuffer \\* AddFixedCache](#) ([const Osl\\_File::OslFixedCacheParam &](#))

### Data Fields

- [OslFileCacheBuffer \\_movableCache](#)
- [Osl\\_Vector< OslFileCacheBuffer, OslMemAllocator > \\_fixedCaches](#)

### Friends

- [class OslFileCacheBuffer](#)

## 7.136.1 Constructor & Destructor Documentation

7.136.1.1 OsciFileCache::OsciFileCache ([Osci\\_File](#) & *aContainer*)

7.136.1.2 OsciFileCache::~~OsciFileCache ()

## 7.136.2 Member Function Documentation

7.136.2.1 OSCL\_IMPORT\_REF [OsciFileCacheBuffer](#)\* OsciFileCache::AddFixedCache (const [Osci\\_File::OsciFixedCacheParam](#) &)

7.136.2.2 void OsciFileCache::Close ()

7.136.2.3 int32 OsciFileCache::EndOfFile () [inline]

7.136.2.4 [TOsciFileOffset](#) OsciFileCache::FileSize () [inline]

7.136.2.5 int32 OsciFileCache::Flush ()

7.136.2.6 int32 OsciFileCache::Open (uint32 *mode*, uint32 *cache\_size*)

7.136.2.7 uint32 OsciFileCache::Read (void \* *outputBuffer*, uint32 *size*, uint32 *numelements*)

7.136.2.8 int32 OsciFileCache::Seek ([TOsciFileOffset](#) *offset*, [Osci\\_File::seek\\_type](#) *origin*)

7.136.2.9 [TOsciFileOffset](#) OsciFileCache::Tell () [inline]

7.136.2.10 uint32 OsciFileCache::Write (const void \* *inputBuffer*, uint32 *size*, uint32 *numelements*)

## 7.136.3 Friends And Related Function Documentation

7.136.3.1 friend class OsciFileCacheBuffer [friend]

## 7.136.4 Field Documentation

7.136.4.1 [Osci\\_Vector](#)<[OsciFileCacheBuffer](#), [OsciMemAllocator](#)> OsciFileCache::\_fixedCaches

7.136.4.2 [OsciFileCacheBuffer](#) OsciFileCache::\_movableCache

The documentation for this class was generated from the following file:

- [osci\\_file\\_cache.h](#)



## 7.137 OsciFileCacheBuffer Class Reference

```
#include <osci_file_cache.h>
```

### Public Methods

- [OsciFileCacheBuffer \(\)](#)
- [int32 SetPosition \(TOsciFileOffset pos\)](#)
- [int32 PrepRead \(\)](#)
- [int32 PrepWrite \(\)](#)
- [int32 WriteUpdatesToFile \(\)](#)
- [int32 FillFromFile \(uint32, uint32\)](#)
- [bool IsUpdated \(\)](#)
- [bool Contains \(TOsciFileOffset pos\)](#)
- [bool Precedes \(TOsciFileOffset pos\)](#)

### Data Fields

- [OsciFileCache \\* iContainer](#)
- [bool isFixed](#)
- [uint32 capacity](#)
- [uint32 usableSize](#)
- [uint8 \\* pBuffer](#)
- [TOsciFileOffset filePosition](#)
- [uint32 currentPos](#)
- [uint32 endPos](#)
- [uint32 updateStart](#)
- [uint32 updateEnd](#)

## 7.137.1 Constructor & Destructor Documentation

7.137.1.1 OsciFileCacheBuffer::OsciFileCacheBuffer () [inline]

## 7.137.2 Member Function Documentation

7.137.2.1 bool OsciFileCacheBuffer::Contains ([TOsciFileOffset](#) *pos*) [inline]

7.137.2.2 int32 OsciFileCacheBuffer::FillFromFile (uint32, uint32)

7.137.2.3 bool OsciFileCacheBuffer::IsUpdated () [inline]

7.137.2.4 bool OsciFileCacheBuffer::Preceeds ([TOsciFileOffset](#) *pos*) [inline]

7.137.2.5 int32 OsciFileCacheBuffer::PrepRead ()

7.137.2.6 int32 OsciFileCacheBuffer::PrepWrite ()

7.137.2.7 int32 OsciFileCacheBuffer::SetPosition ([TOsciFileOffset](#) *pos*)

7.137.2.8 int32 OsciFileCacheBuffer::WriteUpdatesToFile ()

## 7.137.3 Field Documentation

7.137.3.1 uint32 OsciFileCacheBuffer::capacity

7.137.3.2 uint32 OsciFileCacheBuffer::currentPos

7.137.3.3 uint32 OsciFileCacheBuffer::endPos

7.137.3.4 [TOsciFileOffset](#) OsciFileCacheBuffer::filePosition

7.137.3.5 [OsciFileCache\\*](#) OsciFileCacheBuffer::iContainer

7.137.3.6 bool OsciFileCacheBuffer::isFixed

7.137.3.7 uint8\* OsciFileCacheBuffer::pBuffer

7.137.3.8 uint32 OsciFileCacheBuffer::updateEnd

7.137.3.9 uint32 OsciFileCacheBuffer::updateStart

7.137.3.10 uint32 OsciFileCacheBuffer::usableSize

The documentation for this class was generated from the following file:

- [osci\\_file\\_cache.h](#)

## 7.138 OsciFileHandle Class Reference

```
#include <osci_file_handle.h>
```

### Public Methods

- [OsciFileHandle](#) ([TOsciFileHandle](#) aHandle)
- [OsciFileHandle](#) (const [OsciFileHandle](#) &aHandle)
- [TOsciFileHandle Handle](#) () const

### Friends

- class [Osci\\_File](#)

### 7.138.1 Detailed Description

OsciFileHandle is a container for a handle to a previously-opened file.

### 7.138.2 Constructor & Destructor Documentation

**7.138.2.1** [OsciFileHandle::OsciFileHandle](#) ([TOsciFileHandle](#) *aHandle*) [inline]

**7.138.2.2** [OsciFileHandle::OsciFileHandle](#) (const [OsciFileHandle](#) &*aHandle*) [inline]

### 7.138.3 Member Function Documentation

**7.138.3.1** [TOsciFileHandle](#) [OsciFileHandle::Handle](#) () const [inline]

### 7.138.4 Friends And Related Function Documentation

**7.138.4.1** [friend class Osci\\_File](#) [friend]

The documentation for this class was generated from the following file:

- [osci\\_file\\_handle.h](#)

## 7.139 OslFileManager Class Reference

```
#include <oscl_file_manager.h>
```

### Public Types

- enum [OSCL\\_FILE\\_ATTRIBUTE\\_TYPE](#) { [OSCL\\_FILE\\_ATTRIBUTE\\_READONLY](#) = 0x00000001, [OSCL\\_FILE\\_ATTRIBUTE\\_HIDDEN](#) = 0x00000002, [OSCL\\_FILE\\_ATTRIBUTE\\_SYSTEM](#) = 0x00000004, [OSCL\\_FILE\\_ATTRIBUTE\\_DIRECTORY](#) = 0x00000010, [OSCL\\_FILE\\_ATTRIBUTE\\_ARCHIVE](#) = 0x00000020, [OSCL\\_FILE\\_ATTRIBUTE\\_NORMAL](#) = 0x00000080 }

### Static Public Methods

- OSCL\_IMPORT\_REF bool [OslGetFileSize](#) (const [oscl\\_wchar](#) \*aFileName, [uint64](#) &aFileSize)
- OSCL\_IMPORT\_REF bool [OslGetFileSize](#) (const char \*aFileName, [uint64](#) &aFileSize)
- OSCL\_IMPORT\_REF bool [OslGetFileCreationTime](#) (const [oscl\\_wchar](#) \*aFileName, [uint64](#) &aFileCreationTime)
- OSCL\_IMPORT\_REF bool [OslGetFileCreationTime](#) (const char \*aFileName, [uint64](#) &aFileCreationTime)
- OSCL\_IMPORT\_REF bool [OslGetFileLastAccessTime](#) (const [oscl\\_wchar](#) \*aFileName, [uint64](#) &aFileLastAccessTime)
- OSCL\_IMPORT\_REF bool [OslGetFileLastAccessTime](#) (const char \*aFileName, [uint64](#) &aFileLastAccessTime)
- OSCL\_IMPORT\_REF bool [OslGetFileLastWriteTime](#) (const [oscl\\_wchar](#) \*aFileName, [uint64](#) &aFileLastWriteTime)
- OSCL\_IMPORT\_REF bool [OslGetFileLastWriteTime](#) (const char \*aFileName, [uint64](#) &aFileLastWriteTime)
- OSCL\_IMPORT\_REF bool [OslGetFileAttributes](#) (const [oscl\\_wchar](#) \*aFileName, [uint32](#) &aFileAttributes)
- OSCL\_IMPORT\_REF bool [OslGetFileAttributes](#) (const char \*aFileName, [uint32](#) &aFileAttributes)
- OSCL\_IMPORT\_REF void [OslExtractFilenameFromFullpath](#) (const char \*aPath, char \*&aFileName)
- OSCL\_IMPORT\_REF void [OslExtractFilenameFromFullpath](#) (const [oscl\\_wchar](#) \*aPath, [oscl\\_wchar](#) \*&aFileName)

### 7.139.1 Member Enumeration Documentation

#### 7.139.1.1 enum OslFileManager::OSCL\_FILE\_ATTRIBUTE\_TYPE

Enumeration values:

**OSCL\_FILE\_ATTRIBUTE\_READONLY**  
**OSCL\_FILE\_ATTRIBUTE\_HIDDEN**  
**OSCL\_FILE\_ATTRIBUTE\_SYSTEM**  
**OSCL\_FILE\_ATTRIBUTE\_DIRECTORY**  
**OSCL\_FILE\_ATTRIBUTE\_ARCHIVE**  
**OSCL\_FILE\_ATTRIBUTE\_NORMAL**

## 7.139.2 Member Function Documentation

**7.139.2.1** `OSCL_IMPORT_REF void OslFileManager::OslExtractFilenameFromFullpath (const oscl\_wchar * aPath, oscl\_wchar *& aFileName) [static]`

**7.139.2.2** `OSCL_IMPORT_REF void OslFileManager::OslExtractFilenameFromFullpath (const char * aPath, char *& aFileName) [static]`

OslExtractFilenameFromFullpath 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 OslFileManager::OslGetFileAttributes (const char * aFileName, uint32 & aFileAttributes) [static]`

OslGetFileAttributes 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 OslFileManager::OslGetFileAttributes (const oscl\_wchar * aFileName, uint32 & aFileAttributes) [static]`

OslGetFileAttributes utility function provides the various attributes of file (or directory) like if it is hidden, read only etc. The uint32 value is to be interpreted as per the enum OSCL\_FILE\_ATTRIBUTE\_TYPE defined in [oscl\\_file\\_manager.h](#)

### Parameters:

*in* ] wide character path; the full path of the file or directory

*out* ] file attributes.

### Returns:

true if successful, otherwise false.

**7.139.2.5 OSCL\_IMPORT\_REF bool OslFileManager::OslGetFileCreationTime (const char \* *aFileName*, [uint64](#) & *aFileCreationTime*) [static]**

OslGetFileCreationTime utility function provides the file (or directory) creation time

**Note:**

On symbian platform, this api returns last modified time.

**Parameters:**

*in* ] character path; the full path of the file or directory

*out* ] creation time in microseconds.

**Returns:**

true if successful, otherwise false.

**7.139.2.6 OSCL\_IMPORT\_REF bool OslFileManager::OslGetFileCreationTime (const [oscl\\_wchar](#) \* *aFileName*, [uint64](#) & *aFileCreationTime*) [static]**

OslGetFileCreationTime utility function provides the file (or directory) creation time

**Note:**

On symbian platform, this api returns last modified time.

**Parameters:**

*in* ] wide character path; the full path of the file or directory

*out* ] creation time in microseconds

**Returns:**

true if successful, otherwise false.

**7.139.2.7 OSCL\_IMPORT\_REF bool OslFileManager::OslGetFileLastAccessTime (const char \* *aFileName*, [uint64](#) & *aFileLastAccessTime*) [static]**

OslGetFileLastAccessTime utility function provides the file (or directory) last access time, which might be different from last modified time.

**Note:**

On symbian platform, this api returns last modified time.

**Parameters:**

*in* ] character path; the full path of the file or directory

*out* ] Last access time in microseconds.

**Returns:**

true if successful, otherwise false.

### 7.139.2.8 OSCL\_IMPORT\_REF bool OsciFileManager::OsciGetFileLastAccessTime (const osci\_wchar \* aFileName, uint64 & aFileLastAccessTime) [static]

OsciGetFileLastAccessTime utility function provides the file (or directory) last access time, which might be different from last modified time.

#### Note:

On symbian platform, this api returns last modified time.

#### Parameters:

*in* ] wide character path; the full path of the file or directory  
*out* ] Last access time in microseconds

#### Returns:

true if successful, otherwise false.

### 7.139.2.9 OSCL\_IMPORT\_REF bool OsciFileManager::OsciGetFileLastWriteTime (const char \* aFileName, uint64 & aFileLastWriteTime) [static]

OsciGetFileLastWriteTime utility function provides the file (or directory) last modified time.

#### Parameters:

*in* ] character path; the full path of the file or directory  
*out* ] last modified time in microseconds

#### Returns:

true if successful, otherwise false.

### 7.139.2.10 OSCL\_IMPORT\_REF bool OsciFileManager::OsciGetFileLastWriteTime (const osci\_wchar \* aFileName, uint64 & aFileLastWriteTime) [static]

OsciGetFileLastWriteTime utility function provides the file (or directory) last modified time.

#### Parameters:

*in* ] wide character path; the full path of the file or directory  
*out* ] last modified time in microseconds

#### Returns:

true if successful, otherwise false.

### 7.139.2.11 OSCL\_IMPORT\_REF bool OsciFileManager::OsciGetFileSize (const char \* aFileName, uint64 & aFileSize) [static]

OsciGetFileSize utility function provides the file size. For directory, this value is undefined.

#### Parameters:

*in* ] character path; the full path of the file or directory  
*out* ] file size in bytes.

#### Returns:

true if successful, otherwise false.

**7.139.2.12 OSCL\_IMPORT\_REF bool OsciFileManager::OsciGetFileSize (const [osci\\_wchar](#) \* *aFileName*, [uint64](#) & *aFileSize*) [static]**

OsciGetFileSize 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:

- [osci\\_file\\_manager.h](#)



## 7.140 OsciFileStats Class Reference

```
#include <osci_file_stats.h>
```

### Public Methods

- [OsciFileStats](#) ([Osci\\_File](#) \*c)
- void [Start](#) (uint32 &aTicks)
- void [End](#) ([TOsciFileOp](#) aOp, uint32 aStart, uint32 aParam=0, [TOsciFileOffset](#) aParam2=0)
- void [Log](#) ([TOsciFileOp](#), [PVLogger](#) \*, uint32)
- void [LogAll](#) ([PVLogger](#) \*, uint32)

### 7.140.1 Constructor & Destructor Documentation

7.140.1.1 [OsciFileStats::OsciFileStats](#) ([Osci\\_File](#) \* c)

### 7.140.2 Member Function Documentation

7.140.2.1 void [OsciFileStats::End](#) ([TOsciFileOp](#) aOp, uint32 aStart, uint32 aParam = 0, [TOsciFileOffset](#) aParam2 = 0)

7.140.2.2 void [OsciFileStats::Log](#) ([TOsciFileOp](#), [PVLogger](#) \*, uint32)

7.140.2.3 void [OsciFileStats::LogAll](#) ([PVLogger](#) \*, uint32)

7.140.2.4 void [OsciFileStats::Start](#) (uint32 & aTicks)

The documentation for this class was generated from the following file:

- [osci\\_file\\_stats.h](#)

## 7.141 OsciFileStatsItem Class Reference

```
#include <osci_file_stats.h>
```

### Data Fields

- uint32 [iOpCount](#)
- [uint64](#) [iParam](#)
- [TOsciFileOffset](#) [iParam2](#)
- uint32 [iStartTick](#)
- uint32 [iTotalTicks](#)

### 7.141.1 Field Documentation

7.141.1.1 uint32 OsciFileStatsItem::iOpCount

7.141.1.2 [uint64](#) OsciFileStatsItem::iParam

7.141.1.3 [TOsciFileOffset](#) OsciFileStatsItem::iParam2

7.141.1.4 uint32 OsciFileStatsItem::iStartTick

7.141.1.5 uint32 OsciFileStatsItem::iTotalTicks

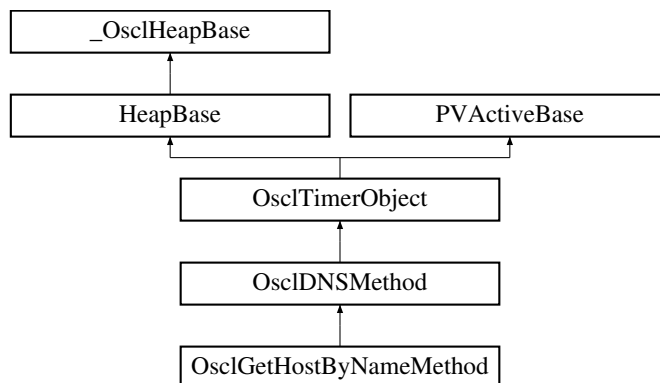
The documentation for this class was generated from the following file:

- [osci\\_file\\_stats.h](#)

## 7.142 OsciGetHostByNameMethod Class Reference

```
#include <osci_dns_gethostbyname.h>
```

Inheritance diagram for OsciGetHostByNameMethod::



### Public Methods

- [~OsciGetHostByNameMethod \(\)](#)
- [TPVDNSEvent GetHostByName \(char \\*name, OsciNetworkAddress \\*addr, int32 aTimeout, Osci\\_Vector< OsciNetworkAddress, OsciMemAllocator > \\*aAddressList\)](#)

### Static Public Methods

- [OsciGetHostByNameMethod \\* NewL \(Osci\\_DefAlloc &a, OsciIDNSI \\*aDNS, OsciDNSObserver \\*aObserver, uint32 aId\)](#)

### 7.142.1 Constructor & Destructor Documentation

#### 7.142.1.1 OsciGetHostByNameMethod::~~OsciGetHostByNameMethod ()

### 7.142.2 Member Function Documentation

#### 7.142.2.1 [TPVDNSEvent OsciGetHostByNameMethod::GetHostByName \(char \\* name, OsciNetworkAddress \\* addr, int32 aTimeout, Osci\\_Vector< OsciNetworkAddress, OsciMemAllocator > \\* aAddressList\)](#)

#### 7.142.2.2 [OsciGetHostByNameMethod\\* OsciGetHostByNameMethod::NewL \(Osci\\_DefAlloc &a, OsciIDNSI \\* aDNS, OsciDNSObserver \\* aObserver, uint32 aId\) \[static\]](#)

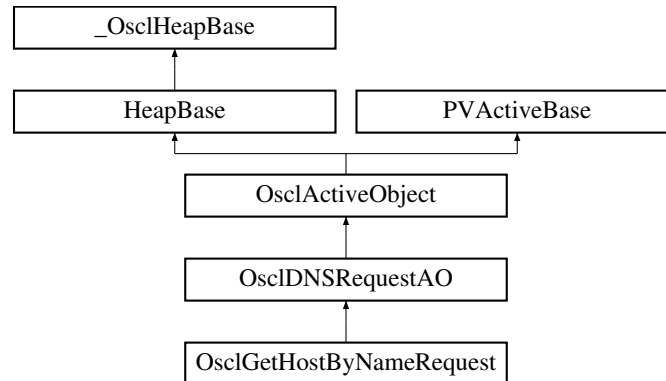
The documentation for this class was generated from the following file:

- [osci\\_dns\\_gethostbyname.h](#)

## 7.143 OsciGetHostByNameRequest Class Reference

```
#include <osci_dns_gethostbyname.h>
```

Inheritance diagram for OsciGetHostByNameRequest::



### Friends

- class [OsciGetHostByNameMethod](#)

### 7.143.1 Friends And Related Function Documentation

#### 7.143.1.1 friend class OsciGetHostByNameMethod [friend]

The documentation for this class was generated from the following file:

- [osci\\_dns\\_gethostbyname.h](#)

## 7.144 OslInit Class Reference

```
#include <osl_init.h>
```

### Static Public Methods

- OSL\_IMPORT\_REF void [Init](#) (int32 &aError, const [OslSelect](#) \*aSelect=NULL)
- OSL\_IMPORT\_REF void [Cleanup](#) (int32 &aError, const [OslSelect](#) \*aSelect=NULL)

### 7.144.1 Detailed Description

Per-thread osl initialization and cleanup.

### 7.144.2 Member Function Documentation

**7.144.2.1 OSL\_IMPORT\_REF void OslInit::Cleanup (int32 &aError, const [OslSelect](#) \*aSelect = NULL) [static]**

This routine cleans up the Osl 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 OSL\_IMPORT\_REF void OslInit::Init (int32 &aError, const [OslSelect](#) \*aSelect = NULL) [static]**

This routine initializes the Osl 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:

- [osl\\_init.h](#)

## 7.145 OsciInteger64Transport Struct Reference

```
#include <osci_int64_utils.h>
```

### Data Fields

- uint32 [iHigh](#)
- uint32 [iLow](#)

### 7.145.1 Detailed Description

OsciInteger64Transport Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

### 7.145.2 Field Documentation

#### 7.145.2.1 uint32 OsciInteger64Transport::iHigh

#### 7.145.2.2 uint32 OsciInteger64Transport::iLow

The documentation for this struct was generated from the following file:

- [osci\\_int64\\_utils.h](#)

## 7.146 OsclIpMReq Class Reference

```
#include <oscl_socket_types.h>
```

### Public Methods

- [OsclIpMReq](#) (const char \*intrfcAddr, const char \*multcstAddr)

### Data Fields

- [OsclNameString](#)< PVNETWORKADDRESS\_LEN > [interfaceAddr](#)
- [OsclNameString](#)< PVNETWORKADDRESS\_LEN > [multicastAddr](#)

### 7.146.1 Constructor & Destructor Documentation

**7.146.1.1** [OsclIpMReq::OsclIpMReq](#) (const char \* *intrfcAddr*, const char \* *multcstAddr*)  
[inline]

### 7.146.2 Field Documentation

**7.146.2.1** [OsclNameString](#)<PVNETWORKADDRESS\_LEN> [OsclIpMReq::interfaceAddr](#)

**7.146.2.2** [OsclNameString](#)<PVNETWORKADDRESS\_LEN> [OsclIpMReq::multicastAddr](#)

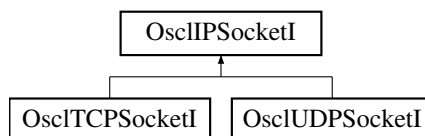
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_types.h](#)

## 7.147 OsciIPSocketI Class Reference

```
#include <osci_ip_socket.h>
```

Inheritance diagram for OsciIPSocketI::



### Public Methods

- int32 [Bind](#) ([OsciNetworkAddress](#) &aAddress)
- int32 [Join](#) ([OsciNetworkAddress](#) &aAddress)
- int32 [SetRecvBufferSize](#) (uint32 size)
- int32 [SetOptionToReuseAddress](#) ()
- int32 [SetTOS](#) (const [OsciSocketTOS](#) &aTOS)
- int32 [GetPeerName](#) ([OsciNetworkAddress](#) &peerName)
- virtual int32 [Close](#) ()=0
- virtual uint8 \* [GetRecvData](#) (int32 \*aLength)=0
- virtual uint8 \* [GetSendData](#) (int32 \*aLength)=0
- virtual ~[OsciIPSocketI](#) ()
- void [ThreadLogoff](#) ()
- void [ThreadLogon](#) ([OsciSocketObserver](#) \*aObs, [OsciSocketServI](#) \*aServ)
- [OsciSocketServI](#) \* [SocketServ](#) ()
- [Osci\\_DefAlloc](#) & [Alloc](#) ()

### Protected Methods

- [OsciIPSocketI](#) ([Osci\\_DefAlloc](#) &a)
- void [ConstructL](#) ([OsciSocketObserver](#) \*aObs, [OsciSocketI](#) \*aSock, [OsciSocketServI](#) \*aServ, uint32 aId)

### Protected Attributes

- [Osci\\_DefAlloc](#) & iAlloc
- [OsciNetworkAddress](#) iAddress
- uint32 iId
- [OsciSocketObserver](#) \* iObserver
- [OsciSocketI](#) \* iSocket
- [OsciSocketServI](#) \* iSocketServ
- [PVLogger](#) \* iLogger

### Friends

- class [OsciSocketRequestAO](#)
- class [OsciSocketMethod](#)



## 7.147.1 Constructor & Destructor Documentation

7.147.1.1 virtual OsciPSSocketI::~~OsciPSSocketI () [inline, virtual]

7.147.1.2 OsciPSSocketI::OsciPSSocketI (Osci\_DefAlloc & a) [inline, protected]

## 7.147.2 Member Function Documentation

7.147.2.1 Osci\_DefAlloc& OsciPSSocketI::Alloc () [inline]

7.147.2.2 int32 OsciPSSocketI::Bind (OsciNetworkAddress & aAddress)

7.147.2.3 virtual int32 OsciPSSocketI::Close () [pure virtual]

Implemented in OsciTCPSSocketI, and OsciUDPSocketI.

7.147.2.4 void OsciPSSocketI::ConstructL (OsciSocketObserver \* aObs, OsciSocketI \* aSock, OsciSocketServI \* aServ, uint32 aId) [protected]

7.147.2.5 int32 OsciPSSocketI::GetPeerName (OsciNetworkAddress & peerName)

7.147.2.6 virtual uint8\* OsciPSSocketI::GetRecvData (int32 \* aLength) [pure virtual]

Implemented in OsciTCPSSocketI, and OsciUDPSocketI.

7.147.2.7 virtual uint8\* OsciPSSocketI::GetSendData (int32 \* aLength) [pure virtual]

Implemented in OsciTCPSSocketI, and OsciUDPSocketI.

7.147.2.8 int32 OsciPSSocketI::Join (OsciNetworkAddress & aAddress)

7.147.2.9 int32 OsciPSSocketI::SetOptionToReuseAddress ()

7.147.2.10 int32 OsciPSSocketI::SetRecvBufferSize (uint32 size)

7.147.2.11 int32 OsciPSSocketI::SetTOS (const OsciSocketTOS & aTOS)

7.147.2.12 OsciSocketServI\* OsciPSSocketI::SocketServ () [inline]

7.147.2.13 void OsciPSSocketI::ThreadLogoff ()

Reimplemented in OsciTCPSSocketI, and OsciUDPSocketI.

**7.147.2.14** void OsciPSocketI::ThreadLogon ([OsciSocketObserver](#) \* *aObs*, [OsciSocketServI](#) \* *aServ*)

### 7.147.3 Friends And Related Function Documentation

**7.147.3.1** friend class OsciSocketMethod [friend]

**7.147.3.2** friend class OsciSocketRequestAO [friend]

### 7.147.4 Field Documentation

**7.147.4.1** [OsciNetworkAddress](#) OsciPSocketI::iAddress [protected]

**7.147.4.2** [Osci\\_DefAlloc&](#) OsciPSocketI::iAlloc [protected]

**7.147.4.3** uint32 OsciPSocketI::iId [protected]

**7.147.4.4** [PVLogger](#)\* OsciPSocketI::iLogger [protected]

**7.147.4.5** [OsciSocketObserver](#)\* OsciPSocketI::iObserver [protected]

**7.147.4.6** [OsciSocketI](#)\* OsciPSocketI::iSocket [protected]

**7.147.4.7** [OsciSocketServI](#)\* OsciPSocketI::iSocketServ [protected]

The documentation for this class was generated from the following file:

- [osci\\_ip\\_socket.h](#)

## 7.148 OslJump Class Reference

```
#include <osl_error_imp_jumps.h>
```

### Public Methods

- void [Jump](#) (int a)
- jmp\_buf \* [Top](#) ()
- [~OslJump](#) ()

### Static Public Methods

- OSCL\_IMPORT\_REF void [StaticJump](#) (int a)

### Friends

- class [OslErrorTrapImp](#)

### 7.148.1 Constructor & Destructor Documentation

**7.148.1.1** [OslJump::~~OslJump](#) () [inline]

### 7.148.2 Member Function Documentation

**7.148.2.1** void [OslJump::Jump](#) (int *a*) [inline]

**7.148.2.2** OSCL\_IMPORT\_REF void [OslJump::StaticJump](#) (int *a*) [static]

**7.148.2.3** jmp\_buf\* [OslJump::Top](#) () [inline]

### 7.148.3 Friends And Related Function Documentation

**7.148.3.1** friend class [OslErrorTrapImp](#) [friend]

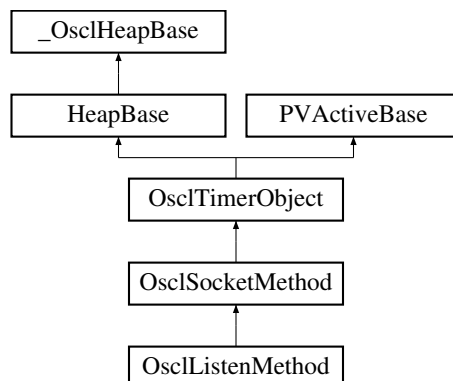
The documentation for this class was generated from the following file:

- [osl\\_error\\_imp\\_jumps.h](#)

## 7.149 OsciListenMethod Class Reference

```
#include <osci_socket_listen.h>
```

Inheritance diagram for OsciListenMethod::



### Public Methods

- [~OsciListenMethod \(\)](#)
- [TPVSocketEvent Listen](#) (uint32 qsize, int32 aTimeout)
- [OsciListenRequest \\* ListenRequest \(\)](#)

### Static Public Methods

- [OsciListenMethod \\* NewL](#) ([OsciIPSocketI](#) &c)

### 7.149.1 Constructor & Destructor Documentation

#### 7.149.1.1 OsciListenMethod::~~OsciListenMethod ()

### 7.149.2 Member Function Documentation

#### 7.149.2.1 [TPVSocketEvent](#) OsciListenMethod::Listen (uint32 qsize, int32 aTimeout)

#### 7.149.2.2 [OsciListenRequest\\*](#) OsciListenMethod::ListenRequest () [inline]

#### 7.149.2.3 [OsciListenMethod\\*](#) OsciListenMethod::NewL ([OsciIPSocketI](#) & c) [static]

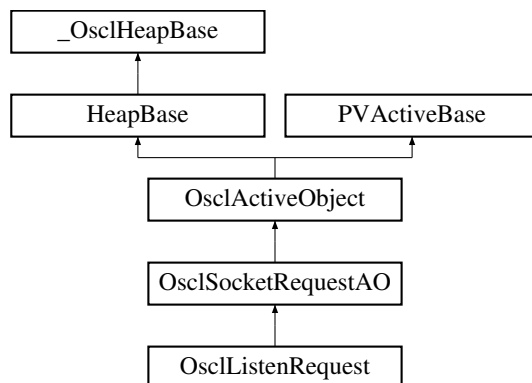
The documentation for this class was generated from the following file:

- [osci\\_socket\\_listen.h](#)

## 7.150 OsciListenRequest Class Reference

```
#include <osci_socket_listen.h>
```

Inheritance diagram for OsciListenRequest::



### Public Methods

- [OsciListenRequest](#) ([OsciSocketMethod](#) &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** [OsciListenRequest::OsciListenRequest](#) ([OsciSocketMethod](#) &c) [inline]

### 7.150.3 Member Function Documentation

**7.150.3.1** void [OsciListenRequest::Listen](#) (uint32 qsize)

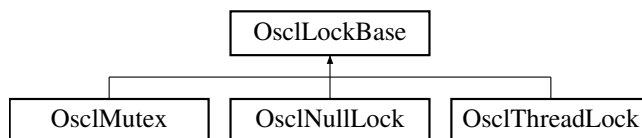
The documentation for this class was generated from the following file:

- [osci\\_socket\\_listen.h](#)

## 7.151 OsciLockBase Class Reference

```
#include <osci_lock_base.h>
```

Inheritance diagram for OsciLockBase::



### Public Methods

- virtual void [Lock](#) ()=0
- virtual void [Unlock](#) ()=0
- virtual [~OsciLockBase](#) ()

### 7.151.1 Constructor & Destructor Documentation

**7.151.1.1** virtual OsciLockBase::~~OsciLockBase () [inline, virtual]

### 7.151.2 Member Function Documentation

**7.151.2.1** virtual void OsciLockBase::Lock () [pure virtual]

Implemented in [OsciNullLock](#), [OsciMutex](#), and [OsciThreadLock](#).

**7.151.2.2** virtual void OsciLockBase::Unlock () [pure virtual]

Implemented in [OsciNullLock](#), [OsciMutex](#), and [OsciThreadLock](#).

The documentation for this class was generated from the following file:

- [osci\\_lock\\_base.h](#)

## 7.152 OslMem Class Reference

```
#include <osl_mem.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [Init](#) ()
- OSCL\_IMPORT\_REF void [Cleanup](#) ()

### 7.152.1 Member Function Documentation

#### 7.152.1.1 OSCL\_IMPORT\_REF void OslMem::Cleanup () [static]

Per-thread cleanup of Osl Memory @exception: Leaves on error;

#### 7.152.1.2 OSCL\_IMPORT\_REF void OslMem::Init () [static]

Per-thread initialization of Osl Memory

#### Parameters:

**lock:** A lock class for use with multi-threaded applications. The lock is needed in use cases where memory may be allocated in one thread and freed in another. In this case, there must be a single lock object, and its pointer must be passed to the [OslMem::Init](#) call in each thread. If no lock is provided, the memory manager will not be thread-safe. @exception: Leaves on error

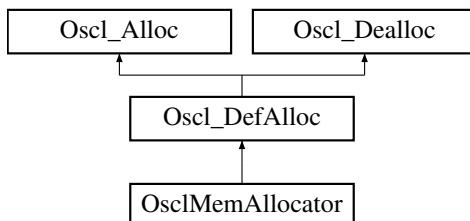
The documentation for this class was generated from the following file:

- [osl\\_mem.h](#)

## 7.153 OsciMemAllocator Class Reference

```
#include <osci_mem.h>
```

Inheritance diagram for OsciMemAllocator::



### Public Methods

- [OsciAny \\* allocate](#) (const uint32 n)
- [OsciAny \\* allocate\\_fl](#) (const uint32 n, const char \*file\_name, const int line\_num)
- void [dealloc](#) ([OsciAny \\*p](#))

### 7.153.1 Detailed Description

A simple allocator class. Configurable as to whether this goes through the memory manager or not.

### 7.153.2 Member Function Documentation

#### 7.153.2.1 [OsciAny\\* OsciMemAllocator::allocate](#) (const uint32 n) [inline, virtual]

This API throws an exception when malloc returns NULL. n must be greater than 0.

#### Returns:

pointer (or Leave with OsciErrNoMemory )

Implements [Osci\\_DefAlloc](#).

#### 7.153.2.2 [OsciAny\\* OsciMemAllocator::allocate\\_fl](#) (const uint32 n, const char \*file\_name, const int line\_num) [inline, virtual]

Reimplemented from [Osci\\_DefAlloc](#).

#### 7.153.2.3 void [OsciMemAllocator::dealloc](#) ([OsciAny \\*p](#)) [inline, virtual]

Implements [Osci\\_DefAlloc](#).

The documentation for this class was generated from the following file:

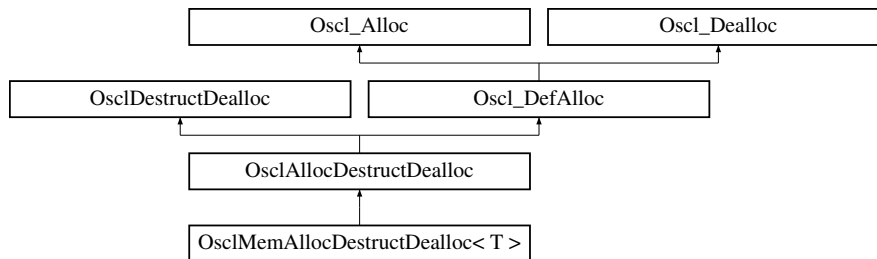
- [osci\\_mem.h](#)



## 7.154 OsciMemAllocDestructDealloc< T > Class Template Reference

```
#include <osci_mem.h>
```

Inheritance diagram for OsciMemAllocDestructDealloc< T >::



### Public Methods

- [OsciAny](#) \* [allocate\\_fl](#) (const uint32 size, const char \*file\_name, const int line\_num)
- [OsciAny](#) \* [allocate](#) (const uint32 size)
- void [dealloc](#) ([OsciAny](#) \*p)
- void [destruct\\_and\\_dealloc](#) ([OsciAny](#) \*p)

### 7.154.1 Detailed Description

**template<class T> class OsciMemAllocDestructDealloc< T >**

An [OsciAllocDestructDealloc](#) class that uses [OsciMemAllocator](#).

### 7.154.2 Member Function Documentation

**7.154.2.1** **template<class T> [OsciAny](#)\* OsciMemAllocDestructDealloc< T >::allocate** (const uint32 size) [inline, virtual]

Implements [Osci\\_DefAlloc](#).

**7.154.2.2** **template<class T> [OsciAny](#)\* OsciMemAllocDestructDealloc< T >::allocate\_fl** (const uint32 size, const char \*file\_name, const int line\_num) [inline, virtual]

Reimplemented from [Osci\\_DefAlloc](#).

**7.154.2.3** **template<class T> void OsciMemAllocDestructDealloc< T >::dealloc** ([OsciAny](#) \*p) [inline, virtual]

Implements [Osci\\_DefAlloc](#).

**7.154.2.4** `template<class T> void OsciMemAllocDestructDealloc< T >::destruct_and_dealloc  
(OsciAny *p) [inline, virtual]`

Implements [OsciDestructDealloc](#).

The documentation for this class was generated from the following file:

- [osci\\_mem.h](#)

## 7.155 OslMemAudit Class Reference

```
#include <osl_mem_audit.h>
```

### Public Methods

- [OslMemAudit \(\)](#)
- [~OslMemAudit \(\)](#)
- void \* [MM\\_allocate](#) (const [OslMemStatsNode](#) \*statsNode, uint32 sizeIn, const char \*pFileName, uint32 lineNumber, bool allocNodeTracking=false)
- bool [MM\\_deallocate](#) (void \*pMemBlockIn)
- [MM\\_Stats\\_t](#) \* [MM\\_GetStats](#) (const char \*const tagIn)
- uint32 [MM\\_GetStatsInDepth](#) (const char \*tagIn, [MM\\_Stats\\_CB](#) \*array\_ptr, uint32 max\_nodes)
- uint32 [MM\\_GetTreeNodees](#) (const char \*tagIn)
- bool [MM\\_AddTag](#) (const char \*tagIn)
- const [OslMemStatsNode](#) \* [MM\\_GetTagNode](#) (const char \*tagIn)
- const [OslMemStatsNode](#) \* [MM\\_GetExistingTag](#) (const char \*tagIn)
- const [OslMemStatsNode](#) \* [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](#) ()
- [OslLockBase](#) \* [GetLock](#) ()

### Friends

- class [OslMemGlobalAuditObject](#)

### 7.155.1 Constructor & Destructor Documentation

#### 7.155.1.1 OslMemAudit::OslMemAudit () [inline]

Constructor, create the root node in statistics table

### 7.155.1.2 OsciMemAudit::~~OsciMemAudit () [inline]

A destructor, remove all the nodes in allocation and statistics table

## 7.155.2 Member Function Documentation

### 7.155.2.1 OsciLockBase\* OsciMemAudit::GetLock () [inline]

API to obtain mem lock ptr

### 7.155.2.2 bool OsciMemAudit::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\* OsciMemAudit::MM\_allocate (const OsciMemStatsNode \* *statsNode*, uint32 *sizeIn*, const char \* *pFileName*, uint32 *lineNumber*, bool *allocNodeTracking* = false) [inline]

The following are APIs t \_\_nothrow\_/ const \_\_nothrow\_

#### Returns:

the memory pointer if operation succeeds.

### 7.155.2.4 MM\_AllocQueryInfo\* OsciMemAudit::MM\_CreateAllocNodeInfo (uint32 *max\_array\_size*) [inline]

### 7.155.2.5 bool OsciMemAudit::MM\_deallocate (void \* *pMemBlockIn*) [inline]

#### Returns:

true if operation succeeds;

### 7.155.2.6 uint32 OsciMemAudit::MM\_GetAllocNo (void) [inline]

API to get the current allocation number

#### Returns:

the current allocation number

**7.155.2.7** `uint32 OsciMemAudit::MM_GetAllocNodeInfo (MM_AllocQueryInfo * output_array, uint32 max_array_size, uint32 offset) [inline]`

API to query the list of alloc nodes. It copies the information into the provided output array.

**Parameters:**

*output\_array* the array where the data will be written

*max\_array\_size* the max number of output array elements

*offset* the offset into the alloc node list from which the data should begin.

**Returns:**

the number of valid nodes in the output array

**7.155.2.8** `const OsciMemStatsNode* OsciMemAudit::MM_GetExistingTag (const char * tagIn) [inline]`

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

true if operation succeeds;

**7.155.2.9** `uint32 OsciMemAudit::MM_GetMode (void) [inline]`

API to get the operating mode of the mm\_audit class.

**7.155.2.10** `uint32 OsciMemAudit::MM_GetNumAllocNodes () [inline]`

API to get the number of allocation nodes (records) for allocations that are being tracked individually.

**7.155.2.11** `void OsciMemAudit::MM_GetOverheadStats (MM_AuditOverheadStats & stats) [inline]`

API to get the overhead statistics for the memory used by the mm\_audit class.

**7.155.2.12** `uint32 OsciMemAudit::MM_GetPostfillPattern (void) [inline]`

API to get the postfill pattern. The pattern is used to fill the memory before freeing it.

**7.155.2.13** `uint8 OsciMemAudit::MM_GetPrefillPattern (void) [inline]`

API to get the prefill pattern. The pattern is used to fill the memory before returning it to the caller.

**7.155.2.14** `int32 OsciMemAudit::MM_GetRefCount () [inline]`

**7.155.2.15** `const OsciMemStatsNode* OsciMemAudit::MM_GetRootNode () [inline]`

**7.155.2.16** `MM_Stats_t* OsciMemAudit::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 OsciMemAudit::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 OsciMemStatsNode* OsciMemAudit::MM_GetTagNode (const char * tagIn) [inline]`

API to add a node and zero out its counters; Note that this tag should be re-used

**Parameters:**

*tagIn* input tag

**Returns:**

pointer to [OsciMemStatsNode](#) which should be passed to MM\_allocate

**7.155.2.19** `uint32 OsciMemAudit::MM_GetTreeNodees (const char * tagIn) [inline]`

API to get the number of tree nodes including the tag node and its subtree

**Parameters:**

*tagIn* input tag

**Returns:**

the number of tree nodes ; 0 means no tag node

**7.155.2.20** `void OsciMemAudit::MM_ReleaseAllocNodeInfo (MM_AllocQueryInfo * info) [inline]`

**7.155.2.21** `bool OsciMemAudit::MM_SetFailurePoint (const char * tagIn, uint32 alloc_number) [inline]`

API to insert allocation failure deterministically according to allocation number associated with tag

**Parameters:**

*tagIn* input tag  
*alloc\_number* allocation number associated with tag

**Returns:**

true if operation succeeds;

**7.155.2.22 void OsciMemAudit::MM\_SetMode (uint32 *inMode*) [inline]**

API to set the operating mode of the mm\_audit class.

**7.155.2.23 void OsciMemAudit::MM\_SetPostfillPattern (uint8 *pattern*) [inline]**

API to set the postfill pattern.

**7.155.2.24 void OsciMemAudit::MM\_SetPrefillPattern (uint8 *pattern*) [inline]**

API to set the prefill pattern.

**7.155.2.25 void OsciMemAudit::MM\_SetTagLevel (uint32 *level*) [inline]**

API to set the maximum tag level,i.e. tag level for a.b.c.d = 4

**Parameters:**

*level* input tag level to be set

**7.155.2.26 void OsciMemAudit::MM\_UnsetFailurePoint (const char \* *tagIn*) [inline]**

API to cancel the allocation failure point associated with tag

**Parameters:**

*tagIn* input tag

**7.155.2.27 bool OsciMemAudit::MM\_Validate (const void \* *ptrIn*) [inline]**

API to check the input pointer is a valid pointer to a chunk of memory

**Parameters:**

*ptrIn* input pointer to be validated

**Returns:**

true if operation succeeds;

### 7.155.3 Friends And Related Function Documentation

#### 7.155.3.1 friend class OslMemGlobalAuditObject [friend]

The documentation for this class was generated from the following file:

- [osl\\_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 = Osl\_TAlloc<T, OslMemAllocator>>> class OSCLMemAutoPtr< T, \_Allocator >**

The `oscl_auto_ptr` class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by `new`. When the `oscl_auto_ptr` expires, its destructor uses `delete` to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by `new` to an `oscl_auto_ptr` object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The `oscl_auto_ptr` is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### 7.156.2 Constructor & Destructor Documentation

**7.156.2.1** **template<class T, class \_Allocator = Osl\_TAlloc<T, OslMemAllocator>>>**  
**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 = Osl\_TAlloc<T, OslMemAllocator>>>**  
**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 = Osl\_TAlloc<T, OslMemAllocator>>>**  
**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 = Osl_TAlloc<T, OslMemAllocator>> void OSCLMemAutoPtr< T, _Allocator >::allocate (oscl_memsize_t size) [inline]`

**7.156.3.2** `template<class T, class _Allocator = Osl_TAlloc<T, OslMemAllocator>> void OSCLMemAutoPtr< T, _Allocator >::deallocate (T *ptr) [inline, static]`

**7.156.3.3** `template<class T, class _Allocator = Osl_TAlloc<T, OslMemAllocator>> 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 = Osl_TAlloc<T, OslMemAllocator>> 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 = Osl_TAlloc<T, OslMemAllocator>> 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 = Osl_TAlloc<T, OslMemAllocator>> 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 = Osl_TAlloc<T, OslMemAllocator>> 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 = Osl_TAlloc<T, OslMemAllocator>> 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 = Osl_TAlloc<T, OslMemAllocator>> 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 = Osl_TAlloc<T, OslMemAllocator>> bool  
OSCLMemAutoPtr< T, _Allocator >::_Ownership`

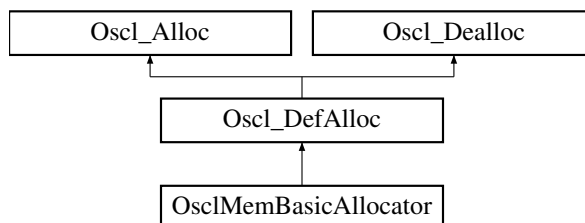
The documentation for this class was generated from the following file:

- [oscl\\_mem\\_auto\\_ptr.h](#)

## 7.157 OsciMemBasicAllocator Class Reference

```
#include <osci_mem.h>
```

Inheritance diagram for OsciMemBasicAllocator::



### Public Methods

- [OsciAny \\* allocate](#) (const uint32 n)
- void [deallocate](#) ([OsciAny \\*p](#))

### 7.157.1 Detailed Description

A simple allocator class that does not use the memory management.

Note: this allocator is for internal use by Osci only. Higher level code should use [OsciMemAllocator](#).

### 7.157.2 Member Function Documentation

#### 7.157.2.1 [OsciAny\\*](#) OsciMemBasicAllocator::allocate (const uint32 n) [inline, virtual]

This API throws an exception when malloc returns NULL. n must be greater than 0.

#### Returns:

pointer (or Leave with OsciErrNoMemory )

Implements [Osci\\_DefAlloc](#).

#### 7.157.2.2 void OsciMemBasicAllocator::deallocate ([OsciAny \\*p](#)) [inline, virtual]

Implements [Osci\\_DefAlloc](#).

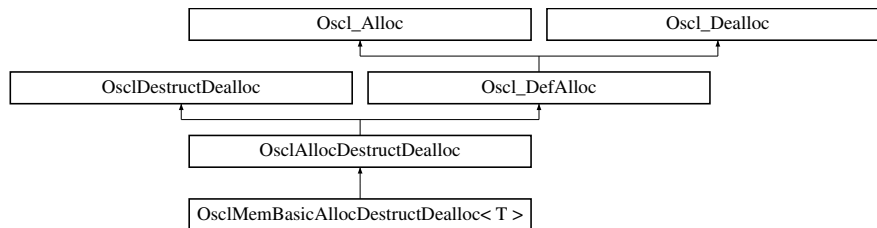
The documentation for this class was generated from the following file:

- [osci\\_mem.h](#)

## 7.158 OslMemBasicAllocDestructDealloc< T > Class Template Reference

```
#include <osl_mem.h>
```

Inheritance diagram for OslMemBasicAllocDestructDealloc< T >::



### Public Methods

- [OslAny \\* allocate](#) (const uint32 size)
- void [deallocate](#) ([OslAny \\*p](#))
- void [destruct\\_and\\_dealloc](#) ([OslAny \\*p](#))

### 7.158.1 Detailed Description

**template<class T> class OslMemBasicAllocDestructDealloc< T >**

An [OslAllocDestructDealloc](#) class that uses [OslMemBasicAllocator](#).

### 7.158.2 Member Function Documentation

**7.158.2.1** **template<class T> [OslAny\\*](#) OslMemBasicAllocDestructDealloc< T >::allocate** (const uint32 size) [inline, virtual]

Implements [Osl\\_DefAlloc](#).

**7.158.2.2** **template<class T> void OslMemBasicAllocDestructDealloc< T >::deallocate** ([OslAny \\*p](#)) [inline, virtual]

Implements [Osl\\_DefAlloc](#).

**7.158.2.3** **template<class T> void OslMemBasicAllocDestructDealloc< T >::destruct\_and\_dealloc** ([OslAny \\*p](#)) [inline, virtual]

Implements [OslDestructDealloc](#).

The documentation for this class was generated from the following file:

- [osl\\_mem.h](#)

## 7.159 OslMemGlobalAuditObject Class Reference

```
#include <osl_mem.h>
```

### Public Types

- typedef [OslMemAudit](#) [audit\\_type](#)

### Static Public Methods

- `OSCL_IMPORT_REF audit\_type * getGlobalMemAuditObject ()`

### Friends

- class [OslMem](#)

### 7.159.1 Member Typedef Documentation

**7.159.1.1** typedef [OslMemAudit](#) `OslMemGlobalAuditObject::audit_type`

### 7.159.2 Member Function Documentation

**7.159.2.1** `OSCL_IMPORT_REF audit\_type* OslMemGlobalAuditObject::getGlobalMemAuditObject () [static]`

returns the global audit object. For use in macros only– not a public API.

### 7.159.3 Friends And Related Function Documentation

**7.159.3.1** friend class `OslMem` [friend]

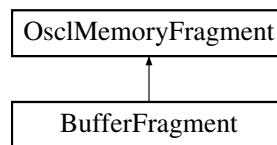
The documentation for this class was generated from the following file:

- [osl\\_mem.h](#)

## 7.160 OsciMemoryFragment Struct Reference

```
#include <osci_types.h>
```

Inheritance diagram for OsciMemoryFragment::



### Data Fields

- uint32 [len](#)
- void \* [ptr](#)

### 7.160.1 Field Documentation

#### 7.160.1.1 uint32 OsciMemoryFragment::len

#### 7.160.1.2 void\* OsciMemoryFragment::ptr

The documentation for this struct was generated from the following file:

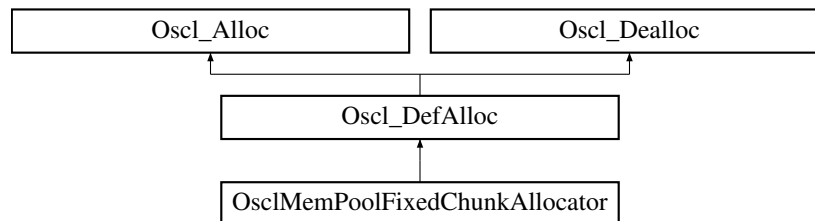
- [osci\\_types.h](#)



## 7.161 OsciMemPoolFixedChunkAllocator Class Reference

```
#include <osci_mem_mempool.h>
```

Inheritance diagram for OsciMemPoolFixedChunkAllocator::



### Public Methods

- OSCL\_IMPORT\_REF [OsciMemPoolFixedChunkAllocator](#) (const uint32 numchunk=1, const uint32 chunksize=0, [Osci\\_DefAlloc](#) \*gen\_alloc=NULL, const uint32 chunkalignment=0)
- virtual OSCL\_IMPORT\_REF void [enablenullpointerreturn](#) ()
- virtual [~OsciMemPoolFixedChunkAllocator](#) ()
- virtual OSCL\_IMPORT\_REF [OsciAny](#) \* [allocate](#) (const uint32 n)
- virtual OSCL\_IMPORT\_REF void [deallocate](#) ([OsciAny](#) \*p)
- virtual OSCL\_IMPORT\_REF void [notifyfreechunkavailable](#) ([OsciMemPoolFixedChunkAllocator-Observer](#) &obs, [OsciAny](#) \*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](#)
- [Osci\\_DefAlloc](#) \* [iMemPoolAllocator](#)
- [OsciAny](#) \* [iMemPool](#)
- [OsciAny](#) \* [iMemPoolAligned](#)
- [Osci\\_Vector](#)< [OsciAny](#) \*, [OsciMemAllocator](#) > [iFreeMemChunkList](#)
- bool [iCheckNextAvailableFreeChunk](#)
- [OsciMemPoolFixedChunkAllocatorObserver](#) \* [iObserver](#)
- [OsciAny](#) \* [iNextAvailableContextData](#)
- int32 [iRefCount](#)
- bool [iEnableNullPtrReturn](#)

## 7.161.1 Constructor & Destructor Documentation

**7.161.1.1** `OSCL_IMPORT_REF OslMemPoolFixedChunkAllocator::OslMemPoolFixedChunkAllocator (const uint32 numchunk = 1, const uint32 chunksize = 0, Osl\_DefAlloc * gen_alloc = NULL, const uint32 chunkalignment = 0)`

This API throws an exception when the memory allocation for pool fails. If numchunk and chunksize parameters are not set, memory pool of 1 chunk will be created in the first call to allocate. The chunk size will be set to the n passed in for [allocate\(\)](#). If numchunk parameter is set to 0, the memory pool will use 1 for numchunk. If chunkalignment is set to 0, memory pool will use default allocator alignment (8-byte). If chunkalignment is > 0, memory pool will align all buffers in the mempool to the specified alignment. Alignment should be a power of 2.

**Returns:**

void

**7.161.1.2** `virtual OslMemPoolFixedChunkAllocator::~OslMemPoolFixedChunkAllocator ()`  
[inline, virtual]

The destructor for the memory pool.

## 7.161.2 Member Function Documentation

**7.161.2.1** `OSCL_IMPORT_REF void OslMemPoolFixedChunkAllocator::addRef ()`

Increments the reference count for this memory pool allocator.

**Returns:**

void

**7.161.2.2** `virtual OSCL_IMPORT_REF OslAny* OslMemPoolFixedChunkAllocator::allocate (const uint32 n)` [virtual]

This API throws an exception when n is greater than the fixed chunk size or there are no free chunks available in the pool, if "enablenullpointerreturn" has not been called. If the memory pool hasn't been created yet, the pool will be created with chunk size equal to n so n must be greater than 0. Exception will be thrown if memory allocation for the memory pool fails.

**Returns:**

pointer to available chunk from memory pool

Implements [Osl\\_DefAlloc](#).

**7.161.2.3** `virtual OSCL_IMPORT_REF void OslMemPoolFixedChunkAllocator::CancelFreeChunkAvailableCallback ()` [virtual]

This API will cancel any past callback requests..

**Returns:**

void

**7.161.2.4** `virtual OSCL_IMPORT_REF void OsciMemPoolFixedChunkAllocator::createmempool ()` [protected, virtual]

**7.161.2.5** `virtual OSCL_IMPORT_REF void OsciMemPoolFixedChunkAllocator::deallocate (OsciAny *p)` [virtual]

This API throws an exception when the pointer p passed in is not part of the memory pool. Exception will be thrown if the memory pool is not set up yet.

**Returns:**

void

Implements [OsciDefAlloc](#).

**7.161.2.6** `virtual OSCL_IMPORT_REF void OsciMemPoolFixedChunk-Allocator::destroymempool ()` [protected, virtual]

**7.161.2.7** `virtual OSCL_IMPORT_REF void OsciMemPoolFixedChunk-Allocator::enablenullpointerreturn ()` [virtual]

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL\\_LEAVE\(OsciErrNoResources\)](#)" allocate API will return NULL.

**Returns:**

void

**7.161.2.8** `virtual OSCL_IMPORT_REF void OsciMemPoolFixedChunk-Allocator::notifyfreechunkavailable (OsciMemPoolFixedChunkAllocatorObserver &obs, OsciAny *aContextData = NULL)` [virtual]

This API will set the flag to send a callback via specified observer object when the next memory chunk is deallocated by [deallocate\(\)](#) call..

**Returns:**

void

**7.161.2.9** `OSCL_IMPORT_REF void OsciMemPoolFixedChunkAllocator::removeRef ()`

Decrements the reference count for this memory pool allocator When the reference count goes to 0, this instance of the memory pool object is deleted

**Returns:**

void

### 7.161.3 Field Documentation

- 7.161.3.1 **bool** OsciMemPoolFixedChunkAllocator::iCheckNextAvailableFreeChunk  
[protected]
- 7.161.3.2 **uint32** OsciMemPoolFixedChunkAllocator::iChunkAlignment [protected]
- 7.161.3.3 **uint32** OsciMemPoolFixedChunkAllocator::iChunkSize [protected]
- 7.161.3.4 **uint32** OsciMemPoolFixedChunkAllocator::iChunkSizeMemAligned [protected]
- 7.161.3.5 **bool** OsciMemPoolFixedChunkAllocator::iEnableNullPtrReturn [protected]
- 7.161.3.6 **Osci\_Vector**<**OsciAny**\*, **OsciMemAllocator**> OsciMemPoolFixedChunkAllocator::iFreeMemChunkList [protected]
- 7.161.3.7 **OsciAny**\* OsciMemPoolFixedChunkAllocator::iMemPool [protected]
- 7.161.3.8 **OsciAny**\* OsciMemPoolFixedChunkAllocator::iMemPoolAligned [protected]
- 7.161.3.9 **Osci\_DefAlloc**\* OsciMemPoolFixedChunkAllocator::iMemPoolAllocator  
[protected]
- 7.161.3.10 **OsciAny**\* OsciMemPoolFixedChunkAllocator::iNextAvailableContextData  
[protected]
- 7.161.3.11 **uint32** OsciMemPoolFixedChunkAllocator::iNumChunk [protected]
- 7.161.3.12 **OsciMemPoolFixedChunkAllocatorObserver**\* OsciMemPoolFixedChunkAllocator::iObserver [protected]
- 7.161.3.13 **int32** OsciMemPoolFixedChunkAllocator::iRefCount [protected]

The documentation for this class was generated from the following file:

- [osci\\_mem\\_mempool.h](#)

## 7.162 OsciMemPoolFixedChunkAllocatorObserver Class Reference

```
#include <osci_mem_mempool.h>
```

### Public Methods

- virtual void [freechunkavailable](#) ([OsciAny](#) \*aContextData)=0
- virtual [~OsciMemPoolFixedChunkAllocatorObserver](#) ()

### 7.162.1 Constructor & Destructor Documentation

**7.162.1.1** virtual [OsciMemPoolFixedChunkAllocatorObserver::~OsciMemPoolFixedChunkAllocatorObserver](#) () [inline, virtual]

### 7.162.2 Member Function Documentation

**7.162.2.1** virtual void [OsciMemPoolFixedChunkAllocatorObserver::freechunkavailable](#) ([OsciAny](#) \* *aContextData*) [pure virtual]

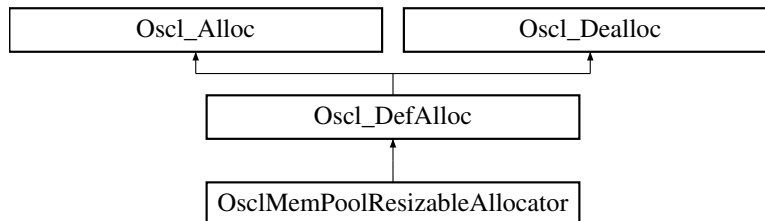
The documentation for this class was generated from the following file:

- [osci\\_mem\\_mempool.h](#)

## 7.163 OslMemPoolResizableAllocator Class Reference

```
#include <osl_mem_mempool.h>
```

Inheritance diagram for OslMemPoolResizableAllocator::



### Public Methods

- OSCL\_IMPORT\_REF [OslMemPoolResizableAllocator](#) (uint32 aMemPoolBufferSize, uint32 aMemPoolBufferNumLimit=0, uint32 aExpectedNumBlocksPerBuffer=0, [Osl\\_DefAlloc](#) \*gen\_alloc=NULL)
- virtual OSCL\_IMPORT\_REF void [enablenullpointerreturn](#) ()
- virtual OSCL\_IMPORT\_REF [OslAny](#) \* [allocate](#) (const uint32 aNumBytes)
- virtual OSCL\_IMPORT\_REF void [deallocate](#) ([OslAny](#) \*aPtr)
- virtual OSCL\_IMPORT\_REF bool [trim](#) ([OslAny](#) \*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 aMaxNewMemPool-BufferSz)
- virtual OSCL\_IMPORT\_REF void [notifyfreeblockavailable](#) ([OslMemPoolResizableAllocator-Observer](#) &aObserver, uint32 aRequestedSize=0, [OslAny](#) \*aContextData=NULL)
- virtual OSCL\_IMPORT\_REF void [CancelFreeChunkAvailableCallback](#) ()
- virtual OSCL\_IMPORT\_REF void [notifyfreememoryavailable](#) ([OslMemPoolResizableAllocator-MemoryObserver](#) &aObserver, uint32 aRequestedSize=0, [OslAny](#) \*aContextData=NULL)
- OSCL\_IMPORT\_REF void [CancelFreeMemoryAvailableCallback](#) ()
- OSCL\_IMPORT\_REF void [addRef](#) ()
- OSCL\_IMPORT\_REF void [removeRef](#) ()

### Protected Methods

- virtual [~OslMemPoolResizableAllocator](#) ()
- [MemPoolBufferInfo](#) \* [addnewmempoolbuffer](#) (uint32 aBufferSize)
- void [destroyallmempoolbuffers](#) ()
- [MemPoolBlockInfo](#) \* [findfreeblock](#) (uint32 aBlockSize)
- [OslAny](#) \* [allocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr, uint32 aNumBytes)
- void [deallocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr)
- bool [validateblock](#) ([OslAny](#) \*aBlockBufPtr)
- uint32 [getMemPoolBufferSize](#) ([MemPoolBufferInfo](#) \*aBufferInfo) const
- uint32 [getMemPoolBufferAllocatedSize](#) ([MemPoolBufferInfo](#) \*aBufferInfo) const
- uint32 [memoryPoolBufferMgmtOverhead](#) () const

## Protected Attributes

- uint32 [iMemPoolBufferSize](#)
- uint32 [iMemPoolBufferNumLimit](#)
- uint32 [iExpectedNumBlocksPerBuffer](#)
- uint32 [iMaxNewMemPoolBufferSz](#)
- [Osl\\_DefAlloc](#) \* [iMemPoolBufferAllocator](#)
- [Osl\\_Vector](#)< [MemPoolBufferInfo](#) \*, [OslMemAllocator](#) > [iMemPoolBufferList](#)
- uint32 [iBufferInfoAlignedSize](#)
- uint32 [iBlockInfoAlignedSize](#)
- bool [iCheckNextAvailable](#)
- uint32 [iRequestedNextAvailableSize](#)
- [OslAny](#) \* [iNextAvailableContextData](#)
- [OslMemPoolResizableAllocatorObserver](#) \* [iObserver](#)
- bool [iCheckFreeMemoryAvailable](#)
- uint32 [iRequestedAvailableFreeMemSize](#)
- [OslAny](#) \* [iFreeMemContextData](#)
- [OslMemPoolResizableAllocatorMemoryObserver](#) \* [iFreeMemPoolObserver](#)
- int32 [iRefCount](#)
- bool [iEnableNullPtrReturn](#)

## 7.163.1 Constructor & Destructor Documentation

### 7.163.1.1 `OSCL_IMPORT_REF OslMemPoolResizableAllocator::OslMemPoolResizableAllocator (uint32 aMemPoolBufferSize, uint32 aMemPoolBufferNumLimit = 0, uint32 aExpectedNumBlocksPerBuffer = 0, Osl\_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 OslMemPoolResizableAllocator::~~OslMemPoolResizableAllocator ()` [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* OslMemPoolResizableAllocator::addnewmempoolbuffer (uint32 aBufferSize)` [protected]

### 7.163.2.2 `OSCL_IMPORT_REF void OslMemPoolResizableAllocator::addRef ()`

Increments the reference count for this memory pool allocator

**Returns:**

void

**7.163.2.3** virtual OSCL\_IMPORT\_REF [OsciAny](#)\* OsciMemPoolResizableAllocator::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 [Osci\\_DefAlloc](#).

**7.163.2.4** [OsciAny](#)\* OsciMemPoolResizableAllocator::allocateblock ([MemPoolBlockInfo](#) & *aBlockPtr*, uint32 *aNumBytes*) [protected]

**7.163.2.5** virtual OSCL\_IMPORT\_REF void OsciMemPoolResizableAllocator::CancelFreeChunkAvailableCallback () [virtual]

This API will cancel any past callback requests..

**Returns:**

void

**7.163.2.6** OSCL\_IMPORT\_REF void OsciMemPoolResizableAllocator::CancelFreeMemoryAvailableCallback ()

**7.163.2.7** virtual OSCL\_IMPORT\_REF void OsciMemPoolResizableAllocator::deallocate ([OsciAny](#) \* *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 [Osci\\_DefAlloc](#).

**7.163.2.8** void OsciMemPoolResizableAllocator::deallocateblock ([MemPoolBlockInfo](#) & *aBlockPtr*) [protected]

**7.163.2.9** void OsciMemPoolResizableAllocator::destroyallmempoolbuffers () [protected]

**7.163.2.10** virtual OSCL\_IMPORT\_REF void OsciMemPoolResizableAllocator::enablenullpointerreturn () [virtual]

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL\\_LEAVE\(OsciErrNoResources\)](#)" allocate API will return NULL.



**Returns:**

void

**7.163.2.11** [MemPoolBlockInfo\\*](#) OsciMemPoolResizableAllocator::findfreeblock (uint32 *aBlockSize*) [protected]

**7.163.2.12** virtual OSCL\_IMPORT\_REF uint32 OsciMemPoolResizableAllocator::getAllocated-Size () [virtual]

Returns the number of bytes allocated from the buffer<including the overhead bytes that may be allocated by the allocator to keep track of the chunks allocated>

**7.163.2.13** virtual OSCL\_IMPORT\_REF uint32 OsciMemPoolResizableAllocator::getAvailable-Size () [virtual]

Returns the number of bytes available with the buffer

**7.163.2.14** OSCL\_IMPORT\_REF uint32 OsciMemPoolResizableAllocator::getBufferSize ()

Returns the size of the buffer <including the overhead bytes that may be allocated by the allocator>

**7.163.2.15** virtual OSCL\_IMPORT\_REF uint32 OsciMemPoolResizableAllocator::getLargest-ContiguousFreeBlockSize () [virtual]

Returns the size of the largest available chunk in the memory.

**7.163.2.16** uint32 OsciMemPoolResizableAllocator::getMemPoolBufferAllocatedSize ([MemPoolBufferInfo](#) \* *aBufferInfo*) const [protected]

**7.163.2.17** uint32 OsciMemPoolResizableAllocator::getMemPoolBufferSize ([MemPoolBufferInfo](#) \* *aBufferInfo*) const [protected]

**7.163.2.18** uint32 OsciMemPoolResizableAllocator::memoryPoolBufferMgmtOverhead () [protected]

**7.163.2.19** virtual OSCL\_IMPORT\_REF void OsciMemPoolResizable-Allocator::notifyfreeblockavailable ([OsciMemPoolResizableAllocatorObserver](#) & *aObserver*, uint32 *aRequestedSize* = 0, [OsciAny](#) \* *aContextData* = NULL) [virtual]

This API will set the flag to send a callback via specified observer object when the next memory block is deallocated by [deallocate\(\)](#) call. If the optional requested size parameter is set, the callback is sent when a free memory space of requested size becomes available. The optional context data is returned with the callback and can be used by the user to differentiate between different instances of memory pool objects. This memory pool only allows one notify to be queued. Another call to this function will just overwrite the previous call.

**Returns:**

void

**7.163.2.20** `virtual OSCL_IMPORT_REF void OslMemPoolResizableAllocator::notifyfreememoryavailable (OslMemPoolResizableAllocatorMemoryObserver & aObserver, uint32 aRequestedSize = 0, OslAny * aContextData = NULL)` [virtual]

**7.163.2.21** `OSCL_IMPORT_REF void OslMemPoolResizableAllocator::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 OslMemPoolResizableAllocator::setMaxSzForNewMemPoolBuffer (uint32 aMaxNewMemPoolBufferSz)` [virtual]

**7.163.2.23** `virtual OSCL_IMPORT_REF bool OslMemPoolResizableAllocator::trim (OslAny * aPtr, uint32 aBytesToFree)` [virtual]

Returns a tail segment of a previously allocated memory block back to the memory pool. The passed-in pointer to the memory buffer is still valid after the call completes but the buffer size is smaller by the specified amount that was freed. This function allows the user to allocate a larger size block initially when the amount needed is unknown and then return the unused portion of the block when the amount becomes known. This API throws an exception if the pointer passed in is not part of the memory pool or the size to return is bigger than the size of the passed-in block. Exception will be thrown if the memory pool is not set up yet.

**Returns:**

bool True if trim operation successful. False if the block wasn't trimmed

**7.163.2.24** `bool OsciMemPoolResizableAllocator::validateblock (OsciAny * aBlockBufPtr)`  
[protected]

### 7.163.3 Field Documentation

**7.163.3.1** `uint32 OsciMemPoolResizableAllocator::iBlockInfoAlignedSize` [protected]

**7.163.3.2** `uint32 OsciMemPoolResizableAllocator::iBufferInfoAlignedSize` [protected]

**7.163.3.3** `bool OsciMemPoolResizableAllocator::iCheckFreeMemoryAvailable` [protected]

**7.163.3.4** `bool OsciMemPoolResizableAllocator::iCheckNextAvailable` [protected]

**7.163.3.5** `bool OsciMemPoolResizableAllocator::iEnableNullPtrReturn` [protected]

**7.163.3.6** `uint32 OsciMemPoolResizableAllocator::iExpectedNumBlocksPerBuffer`  
[protected]

**7.163.3.7** `OsciAny* OsciMemPoolResizableAllocator::iFreeMemContextData` [protected]

**7.163.3.8** `OsciMemPoolResizableAllocatorMemoryObserver* OsciMemPoolResizable-  
Allocator::iFreeMemPoolObserver` [protected]

**7.163.3.9** `uint32 OsciMemPoolResizableAllocator::iMaxNewMemPoolBufferSz` [protected]

**7.163.3.10** `Osci\_DefAlloc* OsciMemPoolResizableAllocator::iMemPoolBufferAllocator`  
[protected]

**7.163.3.11** `Osci\_Vector<MemPoolBufferInfo*, OsciMemAllocator>  
OsciMemPoolResizableAllocator::iMemPoolBufferList` [protected]

**7.163.3.12** `uint32 OsciMemPoolResizableAllocator::iMemPoolBufferNumLimit` [protected]

**7.163.3.13** `uint32 OsciMemPoolResizableAllocator::iMemPoolBufferSize` [protected]

**7.163.3.14** `OsciAny* OsciMemPoolResizableAllocator::iNextAvailableContextData`  
[protected]

**7.163.3.15** `OsciMemPoolResizableAllocatorObserver* OsciMemPoolResizableAllocator::i-  
Observer` [protected]

**7.163.3.16** `int32 OsciMemPoolResizableAllocator::iRefCount` [protected]

**7.163.3.17** `uint32 OsciMemPoolResizableAllocator::iRequestedAvailableFreeMemSize`  
[protected]

**7.163.3.18** `uint32 OsciMemPoolResizableAllocator::iRequestedNextAvailableSize`  
[protected]

The documentation for this class was generated from the following file:

- [osci\\_mem\\_mempool.h](#)

## 7.164 OsciMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference

```
#include <osci_mem_mempool.h>
```

### Data Fields

- uint32 [iBlockPreFence](#)
- MemPoolBlockInfo \* [iNextFreeBlock](#)
- MemPoolBlockInfo \* [iPrevFreeBlock](#)
- uint32 [iBlockSize](#)
- uint8 \* [iBlockBuffer](#)
- [MemPoolBufferInfo](#) \* [iParentBuffer](#)
- uint32 [iBlockPostFence](#)

### 7.164.1 Field Documentation

**7.164.1.1** uint8\* OsciMemPoolResizableAllocator::MemPoolBlockInfo::iBlockBuffer

**7.164.1.2** uint32 OsciMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPostFence

**7.164.1.3** uint32 OsciMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPreFence

**7.164.1.4** uint32 OsciMemPoolResizableAllocator::MemPoolBlockInfo::iBlockSize

**7.164.1.5** MemPoolBlockInfo\* OsciMemPoolResizableAllocator::MemPoolBlockInfo::iNextFree-Block

**7.164.1.6** [MemPoolBufferInfo](#)\* OsciMemPoolResizableAllocator::MemPoolBlockInfo::iParent-Buffer

**7.164.1.7** MemPoolBlockInfo\* OsciMemPoolResizableAllocator::MemPoolBlockInfo::iPrevFree-Block

The documentation for this struct was generated from the following file:

- [osci\\_mem\\_mempool.h](#)

## 7.165 OsciMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference

```
#include <osci_mem_mempool.h>
```

### Data Fields

- uint32 iBufferPreFence
- OsciAny \* iStartAddr
- OsciAny \* iEndAddr
- uint32 iBufferSize
- uint32 iNumOutstanding
- MemPoolBlockInfo \* iNextFreeBlock
- uint32 iAllocatedSz
- uint32 iBufferPostFence

### 7.165.1 Field Documentation

**7.165.1.1** uint32 OsciMemPoolResizableAllocator::MemPoolBufferInfo::iAllocatedSz

**7.165.1.2** uint32 OsciMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPostFence

**7.165.1.3** uint32 OsciMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPreFence

**7.165.1.4** uint32 OsciMemPoolResizableAllocator::MemPoolBufferInfo::iBufferSize

**7.165.1.5** OsciAny\* OsciMemPoolResizableAllocator::MemPoolBufferInfo::iEndAddr

**7.165.1.6** MemPoolBlockInfo\* OsciMemPoolResizableAllocator::MemPoolBufferInfo::iNextFree-Block

**7.165.1.7** uint32 OsciMemPoolResizableAllocator::MemPoolBufferInfo::iNumOutstanding

**7.165.1.8** OsciAny\* OsciMemPoolResizableAllocator::MemPoolBufferInfo::iStartAddr

The documentation for this struct was generated from the following file:

- [osci\\_mem\\_mempool.h](#)

## 7.166 OsciMemPoolResizableAllocatorMemoryObserver Class Reference

```
#include <osci_mem_mempool.h>
```

### Public Methods

- virtual void [freememoryavailable](#) ([OsciAny](#) \*aContextData)=0
- virtual [~OsciMemPoolResizableAllocatorMemoryObserver](#) ()

### 7.166.1 Constructor & Destructor Documentation

**7.166.1.1** virtual [OsciMemPoolResizableAllocatorMemoryObserver::~OsciMemPoolResizableAllocatorMemoryObserver](#) () [inline, virtual]

### 7.166.2 Member Function Documentation

**7.166.2.1** virtual void [OsciMemPoolResizableAllocatorMemoryObserver::freememoryavailable](#) ([OsciAny](#) \*aContextData) [pure virtual]

The documentation for this class was generated from the following file:

- [osci\\_mem\\_mempool.h](#)

## 7.167 OsciMemPoolResizableAllocatorObserver Class Reference

```
#include <osci_mem_mempool.h>
```

### Public Methods

- virtual void [freeblockavailable](#) ([OsciAny](#) \*aContextData)=0
- virtual [~OsciMemPoolResizableAllocatorObserver](#) ()

### 7.167.1 Constructor & Destructor Documentation

**7.167.1.1** virtual [OsciMemPoolResizableAllocatorObserver::~~OsciMemPoolResizableAllocatorObserver](#) () [inline, virtual]

### 7.167.2 Member Function Documentation

**7.167.2.1** virtual void [OsciMemPoolResizableAllocatorObserver::freeblockavailable](#) ([OsciAny](#) \**aContextData*) [pure virtual]

The documentation for this class was generated from the following file:

- [osci\\_mem\\_mempool.h](#)

## 7.168 OsciMemStatsNode Class Reference

```
#include <osci_mem_audit.h>
```

### Public Methods

- [OsciMemStatsNode \(\)](#)
- void [reset \(\)](#)
- [~OsciMemStatsNode \(\)](#)
- void \* [operator new \(osci\\_memsize\\_t size\)](#)
- void \* [operator new \(osci\\_memsize\\_t size, OsciMemStatsNode \\*ptr\)](#)
- void [operator delete](#) (void \*ptr) throw ()

### Data Fields

- [MM\\_Stats\\_t](#) \* [pMMStats](#)
- [MM\\_FailInsertParam](#) \* [pMMFIPParam](#)
- char \* [tag](#)

### 7.168.1 Constructor & Destructor Documentation

**7.168.1.1** [OsciMemStatsNode::OsciMemStatsNode \(\)](#) [inline]

**7.168.1.2** [OsciMemStatsNode::~~OsciMemStatsNode \(\)](#) [inline]

### 7.168.2 Member Function Documentation

**7.168.2.1** void [OsciMemStatsNode::operator delete \(void \\*ptr\) throw \(\)](#) [inline]

**7.168.2.2** void\* [OsciMemStatsNode::operator new \(osci\\_memsize\\_t size, OsciMemStatsNode \\*ptr\)](#) [inline]

**7.168.2.3** void\* [OsciMemStatsNode::operator new \(osci\\_memsize\\_t size\)](#) [inline]

**7.168.2.4** void [OsciMemStatsNode::reset \(\)](#) [inline]

### 7.168.3 Field Documentation

**7.168.3.1** [MM\\_FailInsertParam](#)\* [OsciMemStatsNode::pMMFIPParam](#)

**7.168.3.2** [MM\\_Stats\\_t](#)\* [OsciMemStatsNode::pMMStats](#)

**7.168.3.3** char\* [OsciMemStatsNode::tag](#)

The documentation for this class was generated from the following file:

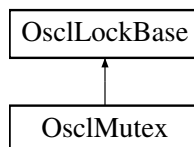
- [osci\\_mem\\_audit.h](#)



## 7.169 OsciMutex Class Reference

```
#include <osci_mutex.h>
```

Inheritance diagram for OsciMutex::



### Public Methods

- OSCL\_IMPORT\_REF [OsciMutex](#) ()
- virtual OSCL\_IMPORT\_REF [~OsciMutex](#) ()
- OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) [Create](#) (void)
- OSCL\_IMPORT\_REF void [Lock](#) ()
- OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) [TryLock](#) ()
- OSCL\_IMPORT\_REF void [Unlock](#) ()
- OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) [Close](#) (void)

### 7.169.1 Detailed Description

Class OsciMutex

### 7.169.2 Constructor & Destructor Documentation

#### 7.169.2.1 OSCL\_IMPORT\_REF OsciMutex::OsciMutex ()

Class constructor

#### 7.169.2.2 virtual OSCL\_IMPORT\_REF OsciMutex::~~OsciMutex () [virtual]

Class destructor

### 7.169.3 Member Function Documentation

#### 7.169.3.1 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciMutex::Close (void)

Closes the Mutex

##### Parameters:

*It* wont take any prameters

##### Returns:

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

**7.169.3.2 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciMutex::Create (void)**

Creates the Mutex

**Parameters:**

*No* input arguments

**Returns:**

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

**7.169.3.3 OSCL\_IMPORT\_REF void OsciMutex::Lock () [virtual]**

Locks the Mutex

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns nothing

Implements [OsciLockBase](#).

**7.169.3.4 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciMutex::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 OsciMutex::Unlock () [virtual]**

Releases the Mutex

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns nothing

Implements [OsciLockBase](#).

The documentation for this class was generated from the following file:

- [osci\\_mutex.h](#)

## 7.170 OsciNameString< \_\_len > Class Template Reference

```
#include <osci_namestring.h>
```

### Public Methods

- [OsciNameString](#) ()
- [OsciNameString](#) (const char a[ ])
- [OsciNameString](#) (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 OsciNameString< __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> OsciNameString< __len >::OsciNameString () [inline]`

**7.170.2.2** `template<int __len> OsciNameString< __len >::OsciNameString (const char a[ ]) [inline]`

**7.170.2.3** `template<int __len> OsciNameString< __len >::OsciNameString (uint8 * a) [inline]`

### 7.170.3 Member Function Documentation

**7.170.3.1** `template<int __len> int32 OsciNameString< __len >::MaxLen () const [inline]`

**7.170.3.2** `template<int __len> void OsciNameString< __len >::Set (const char a[ ]) [inline]`

**7.170.3.3** `template<int __len> void OsciNameString< __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* OsciNameString< __len >::Str () const [inline]`

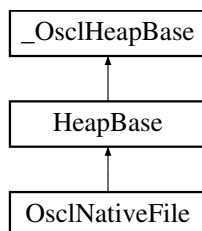
The documentation for this class was generated from the following file:

- [osci\\_namestring.h](#)

## 7.171 OsciNativeFile Class Reference

```
#include <osci_file_native.h>
```

Inheritance diagram for OsciNativeFile::



### Public Methods

- [OsciNativeFile \(\)](#)
- [~OsciNativeFile \(\)](#)
- [int32 Open \(const \[OsciFileHandle\]\(#\) &, uint32 mode, const \[OsciNativeFileParams\]\(#\) &params, \[Osci\\\_FileServer\]\(#\) &fileserv\)](#)
- [int32 Open \(const \[osci\\\_wchar\]\(#\) \\*filename, uint32 mode, const \[OsciNativeFileParams\]\(#\) &params, \[Osci\\\_FileServer\]\(#\) &fileserv\)](#)
- [int32 Open \(const char \\*filename, uint32 mode, const \[OsciNativeFileParams\]\(#\) &params, \[Osci\\\_FileServer\]\(#\) &fileserv\)](#)
- [uint32 Read \(\[OsciAny\]\(#\) \\*buffer, uint32 size, uint32 numelements\)](#)
- [uint32 Write \(const \[OsciAny\]\(#\) \\*buffer, uint32 size, uint32 numelements\)](#)
- [int32 Seek \(\[TOsciFileOffset\]\(#\) offset, \[Osci\\\_File::seek\\\_type\]\(#\) origin\)](#)
- [TOsciFileOffset Tell \(\)](#)
- [int32 Flush \(\)](#)
- [int32 EndOfFile \(\)](#)
- [TOsciFileOffset Size \(\)](#)
- [int32 Close \(\)](#)
- [int32 SetSize \(uint32 size\)](#)
- [uint32 Mode \(\)](#)
- [int32 GetError \(\)](#)
- [int32 ReadAsync \(\[OsciAny\]\(#\) \\*buffer, uint32 size, uint32 numelements, \[OsciAOSStatus\]\(#\) &status\)](#)
- [uint32 GetReadAsyncNumElements \(\)](#)
- [bool HasAsyncRead \(\)](#)
- [void ReadAsyncCancel \(\)](#)

## 7.171.1 Constructor & Destructor Documentation

7.171.1.1 OsciNativeFile::OsciNativeFile ()

7.171.1.2 OsciNativeFile::~~OsciNativeFile ()

## 7.171.2 Member Function Documentation

7.171.2.1 int32 OsciNativeFile::Close ()

7.171.2.2 int32 OsciNativeFile::EndOfFile ()

7.171.2.3 int32 OsciNativeFile::Flush ()

7.171.2.4 int32 OsciNativeFile::GetError ()

7.171.2.5 uint32 OsciNativeFile::GetReadAsyncNumElements ()

Get the number of elements read in the last call to ReadAsync. @returns: number of elements read.

7.171.2.6 bool OsciNativeFile::HasAsyncRead ()

@returns: true if async read is supported natively.

7.171.2.7 uint32 OsciNativeFile::Mode () [inline]

7.171.2.8 int32 OsciNativeFile::Open (const char \* *filename*, uint32 *mode*, const OsciNativeFileParams & *params*, OsciFileServer & *fileserv*)

7.171.2.9 int32 OsciNativeFile::Open (const osci\_wchar \* *filename*, uint32 *mode*, const OsciNativeFileParams & *params*, OsciFileServer & *fileserv*)

7.171.2.10 int32 OsciNativeFile::Open (const OsciFileHandle &, uint32 *mode*, const OsciNativeFileParams & *params*, OsciFileServer & *fileserv*)

7.171.2.11 uint32 OsciNativeFile::Read (OsciAny \* *buffer*, uint32 *size*, uint32 *numelements*)

7.171.2.12 int32 OsciNativeFile::ReadAsync (OsciAny \* *buffer*, uint32 *size*, uint32 *numelements*, OsciAOSStatus & *status*)

Asynchronous read.

### Parameters:

*buffer*: data buffer, must be at least size\*numelements bytes

*size*: size of elements

*numelements*: number of elements to read

*status*: Request status for asynchronous completion @returns: 0 for success.

**7.171.2.13 void OsciNativeFile::ReadAsyncCancel ()**

Cancel any pending async read.

**7.171.2.14 int32 OsciNativeFile::Seek ([TOsciFileOffset](#) *offset*, [Osci\\_File::seek\\_type](#) *origin*)****7.171.2.15 int32 OsciNativeFile::SetSize (uint32 *size*)****7.171.2.16 [TOsciFileOffset](#) OsciNativeFile::Size ()****7.171.2.17 [TOsciFileOffset](#) OsciNativeFile::Tell ()****7.171.2.18 uint32 OsciNativeFile::Write (const [OsciAny](#) \* *buffer*, uint32 *size*, uint32 *numelements*)**

The documentation for this class was generated from the following file:

- [osci\\_file\\_native.h](#)

## 7.172 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 OsciNetworkAddress Class Reference

```
#include <osci_socket_types.h>
```

### Public Methods

- [OsciNetworkAddress](#) ()
- [OsciNetworkAddress](#) (const char \*addr, int p)
- bool [operator==](#) (const OsciNetworkAddress &rhs) const

### Data Fields

- [OsciNameString](#)< PVNETWORKADDRESS\_LEN > [ipAddr](#)
- int [port](#)

### 7.173.1 Constructor & Destructor Documentation

7.173.1.1 [OsciNetworkAddress::OsciNetworkAddress](#) () [inline]

7.173.1.2 [OsciNetworkAddress::OsciNetworkAddress](#) (const char \* *addr*, int *p*) [inline]

### 7.173.2 Member Function Documentation

7.173.2.1 bool [OsciNetworkAddress::operator==](#) (const OsciNetworkAddress & *rhs*) const  
[inline]

### 7.173.3 Field Documentation

7.173.3.1 [OsciNameString](#)<PVNETWORKADDRESS\_LEN> [OsciNetworkAddress::ipAddr](#)

7.173.3.2 int [OsciNetworkAddress::port](#)

The documentation for this class was generated from the following file:

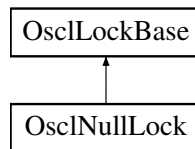
- [osci\\_socket\\_types.h](#)



## 7.174 OsciNullLock Class Reference

```
#include <osci_lock_base.h>
```

Inheritance diagram for OsciNullLock::



### Public Methods

- virtual void [Lock](#) ()
- virtual void [Unlock](#) ()
- virtual [~OsciNullLock](#) ()

### 7.174.1 Constructor & Destructor Documentation

**7.174.1.1** virtual OsciNullLock::~~OsciNullLock () [inline, virtual]

### 7.174.2 Member Function Documentation

**7.174.2.1** virtual void OsciNullLock::Lock () [inline, virtual]

Implements [OsciLockBase](#).

**7.174.2.2** virtual void OsciNullLock::Unlock () [inline, virtual]

Implements [OsciLockBase](#).

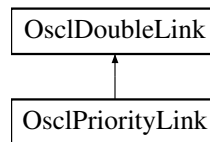
The documentation for this class was generated from the following file:

- [osci\\_lock\\_base.h](#)

## 7.175 OsciPriorityLink Class Reference

```
#include <osci_double_list.h>
```

Inheritance diagram for OsciPriorityLink::



### Data Fields

- int32 [iPriority](#)

### 7.175.1 Field Documentation

#### 7.175.1.1 int32 OsciPriorityLink::iPriority

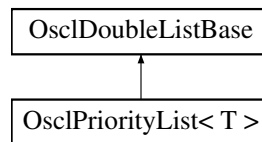
The documentation for this class was generated from the following file:

- [osci\\_double\\_list.h](#)

## 7.176 OsciPriorityList< T > Class Template Reference

```
#include <osci_double_list.h>
```

Inheritance diagram for OsciPriorityList< T >::



### Public Methods

- OSCL\_INLINE [OsciPriorityList](#) ()
- OSCL\_INLINE [OsciPriorityList](#) (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 OsciPriorityList< T >
```

### 7.176.1 Constructor & Destructor Documentation

**7.176.1.1** template<class T> OSCL\_INLINE OsciPriorityList< T >::OsciPriorityList ()

**7.176.1.2** template<class T> OSCL\_INLINE OsciPriorityList< T >::OsciPriorityList (int32 *anOffset*)

### 7.176.2 Member Function Documentation

**7.176.2.1** template<class T> OSCL\_INLINE T\* OsciPriorityList< T >::Head ()

**7.176.2.2** template<class T> OSCL\_INLINE void OsciPriorityList< T >::Insert (T &*aRef*)

**7.176.2.3** template<class T> OSCL\_INLINE bool OsciPriorityList< T >::IsHead (const T \* *aPtr*) const

**7.176.2.4** template<class T> OSCL\_INLINE bool OsciPriorityList< T >::IsTail (const T \* *aPtr*) const

**7.176.2.5** template<class T> OSCL\_INLINE T\* OsciPriorityList< T >::Tail ()

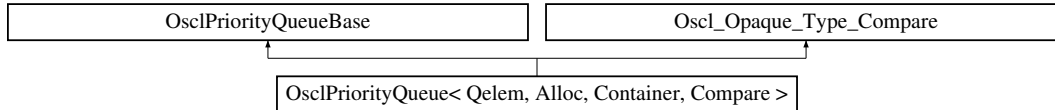
The documentation for this class was generated from the following file:

- [osci\\_double\\_list.h](#)

## 7.177 OsciPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference

```
#include <osci_priorityqueue.h>
```

Inheritance diagram for OsciPriorityQueue< Qelem, Alloc, Container, Compare >::



### Public Types

- typedef Container::value\_type [value\\_type](#)
- typedef Container [container\\_type](#)
- typedef Container::iterator [iterator](#)
- typedef Container::const\_reference [const\\_reference](#)

### Public Methods

- bool [empty](#) () const
- uint32 [size](#) () const
- void [reserve](#) (uint32 n)
- [const\\_reference top](#) () const
- const Container & [vec](#) ()
- void [push](#) (const [value\\_type](#) &input)
- void [pop](#) ()
- int [remove](#) (const [value\\_type](#) &input)
- [OsciPriorityQueue](#) ()
- virtual [~OsciPriorityQueue](#) ()

### Protected Methods

- void [push\\_heap](#) (iterator first, iterator last)
- void [pop\\_heap](#) (iterator first, iterator last)
- iterator [find\\_heap](#) (const [value\\_type](#) &input, iterator first, iterator last)
- int [validate](#) ()
- void [swap](#) (OsciAny \*dest, const OsciAny \*src)
- int [compare\\_LT](#) (OsciAny \*a, OsciAny \*b) const
- int [compare\\_EQ](#) (const OsciAny \*a, const OsciAny \*b) const

### Protected Attributes

- Container [c](#)
- Compare [comp](#)

## Friends

- class [oscl\\_priqueue\\_test](#)

```
template<class Qelem, class Alloc, class Container = 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 [OsciOpaqueTypeCompare](#).

**7.177.3.3** `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> bool OsciPriorityQueue< Qelem, Alloc, Container, Compare >::empty () const [inline]`

**7.177.3.4** `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> iterator OsciPriorityQueue< 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 = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> void OsciPriorityQueue< Qelem, Alloc, Container, Compare >::pop () [inline]`

**7.177.3.6** `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> void OsciPriorityQueue< Qelem, Alloc, Container, Compare >::pop_heap (iterator first, iterator last) [inline, protected]`

**7.177.3.7** `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> void OsciPriorityQueue< Qelem, Alloc, Container, Compare >::push (const value_type & input) [inline]`

**7.177.3.8** `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> void OsciPriorityQueue< Qelem, Alloc, Container, Compare >::push_heap (iterator first, iterator last) [inline, protected]`

**7.177.3.9** `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> int OsciPriorityQueue< Qelem, Alloc, Container, Compare >::remove (const value_type & input) [inline]`

**7.177.3.10** `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> void OsciPriorityQueue< Qelem, Alloc, Container, Compare >::reserve (uint32 n) [inline]`

**7.177.3.11** `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> uint32 OsciPriorityQueue< Qelem, Alloc, Container, Compare >::size () const [inline]`

**7.177.3.12** `template<class Qelem, class Alloc, class Container = Osci_Vector<Qelem, Alloc>, class Compare = OsciCompareLess<Qelem>> void OsciPriorityQueue< Qelem, Alloc, Container, Compare >::swap (OsciAny * dest, const OsciAny * src) [inline, protected, virtual]`

Swap element at "a" with element at "b". Both pointers must be non-NULL.

Implements [OsciOpaqueTypeCompare](#).

- 7.177.3.13** `template<class Qelem, class Alloc, class Container = Osl_Vector<Qelem, Alloc>, class Compare = OslCompareLess<Qelem>> const\_reference OslPriorityQueue<Qelem, Alloc, Container, Compare >::top () const [inline]`
- 7.177.3.14** `template<class Qelem, class Alloc, class Container = Osl_Vector<Qelem, Alloc>, class Compare = OslCompareLess<Qelem>> int OslPriorityQueue< Qelem, Alloc, Container, Compare >::validate () [inline, protected]`
- 7.177.3.15** `template<class Qelem, class Alloc, class Container = Osl_Vector<Qelem, Alloc>, class Compare = OslCompareLess<Qelem>> const Container& OslPriorityQueue<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 = Osl_Vector<Qelem, Alloc>, class Compare = OslCompareLess<Qelem>> friend class oscl_priqueue_test [friend]`

### 7.177.5 Field Documentation

- 7.177.5.1** `template<class Qelem, class Alloc, class Container = Osl_Vector<Qelem, Alloc>, class Compare = OslCompareLess<Qelem>> Container OslPriorityQueue< Qelem, Alloc, Container, Compare >::c [protected]`
- 7.177.5.2** `template<class Qelem, class Alloc, class Container = Osl_Vector<Qelem, Alloc>, class Compare = OslCompareLess<Qelem>> Compare OslPriorityQueue< Qelem, Alloc, Container, Compare >::comp [protected]`

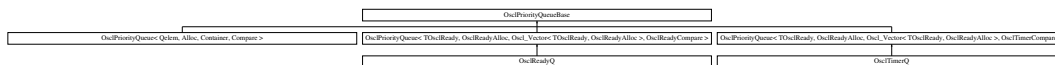
The documentation for this class was generated from the following file:

- [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-templated base class for [OsclPriorityQueue](#). The purpose of this base class is to avoid large inline routines in the [OsclPriorityQueue](#) implementation. This class is not intended for direct instantiation except by [OsclPriorityQueue](#).

### 7.178.2 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 OsciProcStatus Class Reference

```
#include <osci_procstatus.h>
```

### Public Types

- enum `eOsciProcError` { `SUCCESS_ERROR` = 0, `OTHER_ERROR`, `TOO_MANY_THREADS_ERROR`, `BAD_THREADID_ADDR_ERROR`, `MAX_THRDS_REACHED_ERROR`, `INVALID_THREAD_ID_ERROR`, `NOT_ENOUGH_MEMORY_ERROR`, `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_SIGNED_ERROR`, `PSHARED_ATTRIBUTE_SETTING_ERROR`, `NOT_IMPLEMENTED` }

### 7.179.1 Detailed Description

Class `OsciProcStatus`

### 7.179.2 Member Enumeration Documentation

#### 7.179.2.1 enum `OsciProcStatus::eOsciProcError`

List of enums which contain error codes

Enumeration values:

**SUCCESS\_ERROR**  
**OTHER\_ERROR**  
**TOO\_MANY\_THREADS\_ERROR**  
**BAD\_THREADID\_ADDR\_ERROR**  
**MAX\_THRDS\_REACHED\_ERROR**  
**INVALID\_THREAD\_ID\_ERROR**  
**NOT\_ENOUGH\_MEMORY\_ERROR**  
**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**

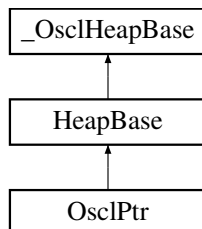
The documentation for this class was generated from the following file:

- [osci\\_procstatus.h](#)

## 7.180 OslPtr Class Reference

```
#include <osl_file_async_read.h>
```

Inheritance diagram for OslPtr::



### Public Methods

- [OslPtr](#) (uint8 \*ptr, int32 &len, int32 max)
- [OslPtr](#) (const OslPtr &d)
- uint8 \* [Ptr](#) ()
- void [SetLength](#) (int32 l)
- int32 [Length](#) ()
- void [Zero](#) ()
- void [Set](#) (OslPtr &v)
- void [Set](#) (uint8 \*ptr, int32 len, int32 max)
- void [Append](#) ([OslPtrC](#) &v)

### 7.180.1 Constructor & Destructor Documentation

**7.180.1.1** OslPtr::OslPtr (uint8 \* *ptr*, int32 & *len*, int32 *max*) [inline]

**7.180.1.2** OslPtr::OslPtr (const OslPtr & *d*) [inline]

### 7.180.2 Member Function Documentation

**7.180.2.1** void OslPtr::Append ([OslPtrC](#) & *v*) [inline]

**7.180.2.2** int32 OslPtr::Length () [inline]

**7.180.2.3** uint8\* OslPtr::Ptr () [inline]

**7.180.2.4** void OslPtr::Set (uint8 \* *ptr*, int32 *len*, int32 *max*) [inline]

**7.180.2.5** void OslPtr::Set (OslPtr & *v*) [inline]

**7.180.2.6** void OslPtr::SetLength (int32 *l*) [inline]

**7.180.2.7** void OslPtr::Zero () [inline]

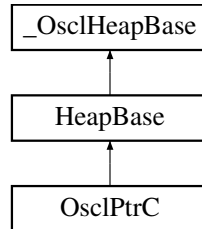
The documentation for this class was generated from the following file:

- [osci\\_file\\_async\\_read.h](#)

## 7.181 OslPtrC Class Reference

```
#include <osl_file_async_read.h>
```

Inheritance diagram for OslPtrC::



### Public Methods

- [OslPtrC](#) (const uint8 \*ptr, int32 len, int32 max)
- [OslPtrC](#) (const OslPtrC &d)
- const uint8 \* [Ptr](#) ()
- void [SetLength](#) (int32 l)
- int32 [Length](#) ()
- void [Zero](#) ()
- void [Set](#) (OslPtrC \*v)
- void [Set](#) (uint8 \*ptr, int32 len, int32 max)
- OslPtrC [Right](#) (int32 size)
- OslPtrC [Left](#) (int32 size)

## 7.181.1 Constructor & Destructor Documentation

7.181.1.1 `OslPtrC::OslPtrC (const uint8 * ptr, int32 len, int32 max)` [inline]

7.181.1.2 `OslPtrC::OslPtrC (const OslPtrC & d)` [inline]

## 7.181.2 Member Function Documentation

7.181.2.1 `OslPtrC OslPtrC::Left (int32 size)` [inline]

7.181.2.2 `int32 OslPtrC::Length ()` [inline]

7.181.2.3 `const uint8* OslPtrC::Ptr ()` [inline]

7.181.2.4 `OslPtrC OslPtrC::Right (int32 size)` [inline]

7.181.2.5 `void OslPtrC::Set (uint8 * ptr, int32 len, int32 max)` [inline]

7.181.2.6 `void OslPtrC::Set (OslPtrC * v)` [inline]

7.181.2.7 `void OslPtrC::SetLength (int32 l)` [inline]

7.181.2.8 `void OslPtrC::Zero ()` [inline]

The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 7.182 OslRand Class Reference

```
#include <osl_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 OslRand::Rand ()

**7.182.1.2** OSCL\_COND\_IMPORT\_REF void OslRand::Seed (int32 *seed*)

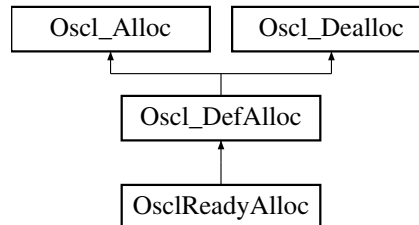
The documentation for this class was generated from the following file:

- [osl\\_rand.h](#)

## 7.183 OsciReadyAlloc Class Reference

```
#include <osci_scheduler_readyq.h>
```

Inheritance diagram for OsciReadyAlloc::



### Public Methods

- [OsciAny](#) \* [allocate](#) (const uint32 size)
- [OsciAny](#) \* [allocate\\_fl](#) (const uint32 size, const char \*file\_name, const int line\_num)
- void [dealloc](#) ([OsciAny](#) \*p)

### 7.183.1 Member Function Documentation

**7.183.1.1** [OsciAny](#)\* [OsciReadyAlloc::allocate](#) (const uint32 *size*) [virtual]

Implements [Osci\\_DefAlloc](#).

**7.183.1.2** [OsciAny](#)\* [OsciReadyAlloc::allocate\\_fl](#) (const uint32 *size*, const char \**file\_name*, const int *line\_num*) [virtual]

Reimplemented from [Osci\\_DefAlloc](#).

**7.183.1.3** void [OsciReadyAlloc::dealloc](#) ([OsciAny](#) \**p*) [virtual]

Implements [Osci\\_DefAlloc](#).

The documentation for this class was generated from the following file:

- [osci\\_scheduler\\_readyq.h](#)



## 7.184 OsciReadyCompare Class Reference

```
#include <osci_scheduler_readyq.h>
```

### Static Public Methods

- `int compare (TOsciReady &a, TOsciReady &b)`

### 7.184.1 Member Function Documentation

**7.184.1.1** `int OsciReadyCompare::compare (TOsciReady &a, TOsciReady &b)` [static]

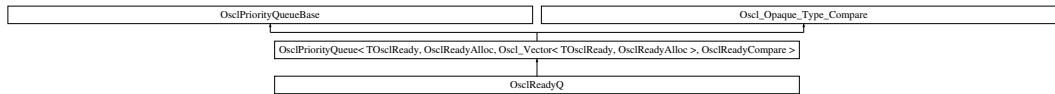
The documentation for this class was generated from the following file:

- [osci\\_scheduler\\_readyq.h](#)

## 7.185 OsclReadyQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyQ::



### Public Methods

- void [Construct](#) (int)
- void [ThreadLogon](#) ()
- void [ThreadLogoff](#) ()
- void [Remove](#) (TOsclReady)
- bool [IsIn](#) (TOsclReady)
- uint32 [Depth](#) ()
- TOsclReady [PopTop](#) ()
- TOsclReady [Top](#) ()
- TOsclReady [WaitAndPopTop](#) ()
- TOsclReady [WaitAndPopTop](#) (uint32)
- int32 [PendComplete](#) (PActiveBase \*pvbase, int32 aReason)
- int32 [WaitForRequestComplete](#) (PActiveBase \*)
- void [RegisterForCallback](#) (OsclSchedulerObserver \*aCallback, OsclAny \*aCallbackContext)
- void [TimerCallback](#) (uint32 aDelayMicrosec)
- OsclSchedulerObserver \* [Callback](#) ()

## 7.185.1 Member Function Documentation

- 7.185.1.1 [OsciSchedulerObserver](#)\* OsciReadyQ::Callback () [inline]
- 7.185.1.2 void OsciReadyQ::Construct (int)
- 7.185.1.3 uint32 OsciReadyQ::Depth () [inline]
- 7.185.1.4 bool OsciReadyQ::IsIn ([TOsciReady](#))
- 7.185.1.5 int32 OsciReadyQ::PendComplete ([PVActiveBase](#) \* *pvbase*, int32 *aReason*)
- 7.185.1.6 [TOsciReady](#) OsciReadyQ::PopTop ()
- 7.185.1.7 void OsciReadyQ::RegisterForCallback ([OsciSchedulerObserver](#) \* *aCallback*, [OsciAny](#) \* *aCallbackContext*)
- 7.185.1.8 void OsciReadyQ::Remove ([TOsciReady](#))
- 7.185.1.9 void OsciReadyQ::ThreadLogoff ()
- 7.185.1.10 void OsciReadyQ::ThreadLogon ()
- 7.185.1.11 void OsciReadyQ::TimerCallback (uint32 *aDelayMicrosec*)
- 7.185.1.12 [TOsciReady](#) OsciReadyQ::Top ()
- 7.185.1.13 [TOsciReady](#) OsciReadyQ::WaitAndPopTop (uint32)
- 7.185.1.14 [TOsciReady](#) OsciReadyQ::WaitAndPopTop ()
- 7.185.1.15 int32 OsciReadyQ::WaitForRequestComplete ([PVActiveBase](#) \*)

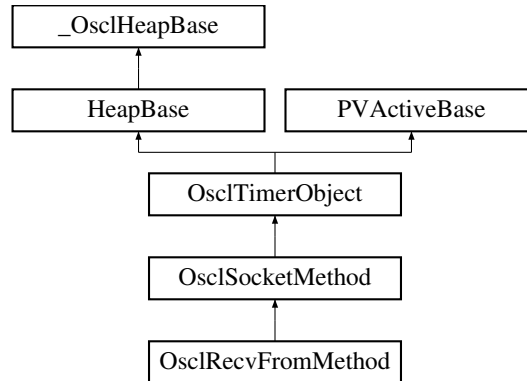
The documentation for this class was generated from the following file:

- [osci\\_scheduler\\_readyq.h](#)

## 7.186 OsciRecvFromMethod Class Reference

```
#include <osci_socket_recv_from.h>
```

Inheritance diagram for OsciRecvFromMethod::



### Public Methods

- [~OsciRecvFromMethod \(\)](#)
- [TPVSocketEvent RecvFrom](#) (uint8 \*&aPtr, uint32 aMaxLen, [OsciNetworkAddress](#) &aAddress, int32 aTimeout, uint32 aMultiMax, [Osci\\_Vector](#)< uint32, [OsciMemAllocator](#) > \*aPacketLen, [Osci\\_Vector](#)< [OsciNetworkAddress](#), [OsciMemAllocator](#) > \*aPacketSource)
- uint8 \* [GetRecvData](#) (int32 \*aLength)
- [OsciRecvFromRequest](#) \* [RecvFromRequest](#) ()

### Static Public Methods

- OsciRecvFromMethod \* [NewL](#) ([OsciIPSocketI](#) &c)

### 7.186.1 Constructor & Destructor Documentation

#### 7.186.1.1 OsciRecvFromMethod::~~OsciRecvFromMethod ()

### 7.186.2 Member Function Documentation

#### 7.186.2.1 uint8\* OsciRecvFromMethod::GetRecvData (int32 \*aLength)

#### 7.186.2.2 OsciRecvFromMethod\* OsciRecvFromMethod::NewL ([OsciIPSocketI](#) &c) [static]

#### 7.186.2.3 [TPVSocketEvent](#) OsciRecvFromMethod::RecvFrom (uint8 \*&aPtr, uint32 aMaxLen, [OsciNetworkAddress](#) &aAddress, int32 aTimeout, uint32 aMultiMax, [Osci\\_Vector](#)< uint32, [OsciMemAllocator](#) > \*aPacketLen, [Osci\\_Vector](#)< [OsciNetworkAddress](#), [OsciMemAllocator](#) > \*aPacketSource)

#### 7.186.2.4 [OsciRecvFromRequest](#)\* OsciRecvFromMethod::RecvFromRequest () [inline]

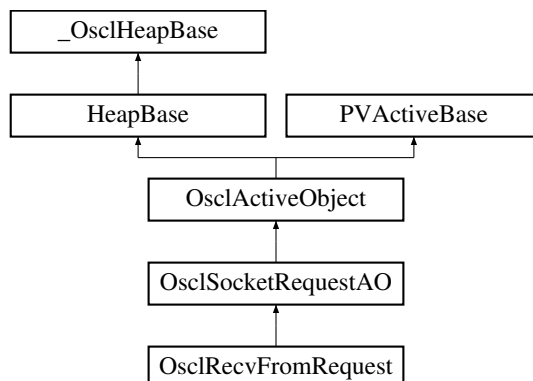
The documentation for this class was generated from the following file:

- [osci\\_socket\\_recv\\_from.h](#)

## 7.187 OsciRecvFromRequest Class Reference

```
#include <osci_socket_recv_from.h>
```

Inheritance diagram for OsciRecvFromRequest::



### Public Methods

- `uint8 * GetRecvData (int32 *aLength)`
- `OsciRecvFromRequest (OsciSocketMethod &c)`
- `void RecvFrom (uint8 *&aPtr, uint32 aMaxLen, OsciNetworkAddress &aAddress, uint32 aMultiMax, Osci_Vector< uint32, OsciMemAllocator > *aPacketLen, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > *aPacketSource)`
- `void Success ()`

### 7.187.1 Detailed Description

This is the AO that interacts with the socket server

### 7.187.2 Constructor & Destructor Documentation

**7.187.2.1** `OsciRecvFromRequest::OsciRecvFromRequest (OsciSocketMethod &c) [inline]`

### 7.187.3 Member Function Documentation

**7.187.3.1** `uint8* OsciRecvFromRequest::GetRecvData (int32 *aLength)`

**7.187.3.2** `void OsciRecvFromRequest::RecvFrom (uint8 *&aPtr, uint32 aMaxLen, OsciNetworkAddress &aAddress, uint32 aMultiMax, Osci_Vector< uint32, OsciMemAllocator > *aPacketLen, Osci_Vector< OsciNetworkAddress, OsciMemAllocator > *aPacketSource)`

**7.187.3.3** `void OsciRecvFromRequest::Success () [virtual]`

Reimplemented from `OsciSocketRequestAO`.

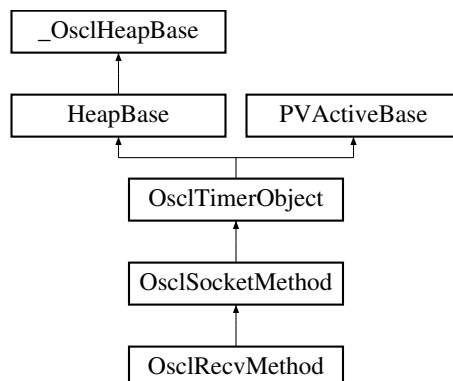
The documentation for this class was generated from the following file:

- [osci\\_socket\\_recv\\_from.h](#)

## 7.188 OsciRecvMethod Class Reference

```
#include <osci_socket_recv.h>
```

Inheritance diagram for OsciRecvMethod::



### Public Methods

- [~OsciRecvMethod \(\)](#)
- [TPVSocketEvent Recv](#) (uint8 \*&aPtr, uint32 aMaxLen, int32 aTimeout)
- uint8 \* [GetRecvData](#) (int32 \*aLength)
- [OsciRecvRequest](#) \* [RecvRequest](#) ()

### Static Public Methods

- OsciRecvMethod \* [NewL](#) ([OsciIPSocketI](#) &c)

### 7.188.1 Constructor & Destructor Documentation

#### 7.188.1.1 OsciRecvMethod::~~OsciRecvMethod ()

### 7.188.2 Member Function Documentation

#### 7.188.2.1 uint8\* OsciRecvMethod::GetRecvData (int32 \*aLength)

#### 7.188.2.2 OsciRecvMethod\* OsciRecvMethod::NewL ([OsciIPSocketI](#) &c) [static]

#### 7.188.2.3 [TPVSocketEvent](#) OsciRecvMethod::Recv (uint8 \*&aPtr, uint32 aMaxLen, int32 aTimeout)

#### 7.188.2.4 [OsciRecvRequest](#)\* OsciRecvMethod::RecvRequest () [inline]

The documentation for this class was generated from the following file:

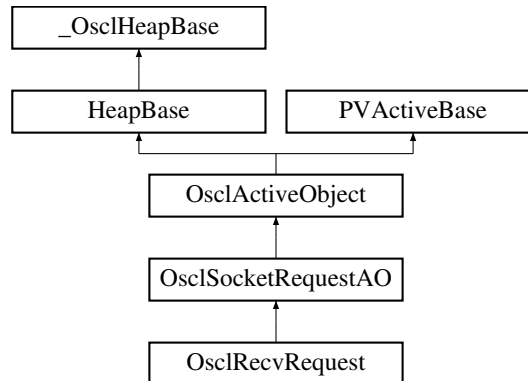
- [osci\\_socket\\_recv.h](#)



## 7.189 OsciRecvRequest Class Reference

```
#include <osci_socket_recv.h>
```

Inheritance diagram for OsciRecvRequest::



### Public Methods

- uint8 \* [GetRecvData](#) (int32 \*aLength)
- [OsciRecvRequest](#) ([OsciSocketMethod](#) &c)
- void [Recv](#) (uint8 \*&aPtr, uint32 aMaxLen)
- void [Success](#) ()

### 7.189.1 Detailed Description

This is the AO that interacts with the socket server

### 7.189.2 Constructor & Destructor Documentation

**7.189.2.1** [OsciRecvRequest::OsciRecvRequest](#) ([OsciSocketMethod](#) &c) [inline]

### 7.189.3 Member Function Documentation

**7.189.3.1** uint8\* [OsciRecvRequest::GetRecvData](#) (int32 \*aLength)

**7.189.3.2** void [OsciRecvRequest::Recv](#) (uint8 \*&aPtr, uint32 aMaxLen)

**7.189.3.3** void [OsciRecvRequest::Success](#) () [virtual]

Reimplemented from [OsciSocketRequestAO](#).

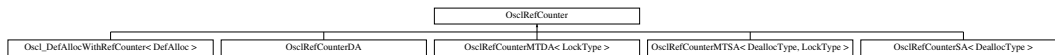
The documentation for this class was generated from the following file:

- [osci\\_socket\\_recv.h](#)

## 7.190 OsciRefCounter Class Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for OsciRefCounter::



### Public Methods

- virtual void [addRef](#) ()=0
- virtual void [removeRef](#) ()=0
- virtual uint32 [getCount](#) ()=0
- virtual [~OsciRefCounter](#) ()

### 7.190.1 Detailed Description

Interface class for OsciRefCounter implementations

### 7.190.2 Constructor & Destructor Documentation

**7.190.2.1** virtual OsciRefCounter::~~OsciRefCounter () [inline, virtual]

### 7.190.3 Member Function Documentation

**7.190.3.1** virtual void OsciRefCounter::addRef () [pure virtual]

Add to the reference count

Implemented in [OsciRefCounterDA](#), [OsciRefCounterSA< DeallocType >](#), [OsciRefCounterMTDA< LockType >](#), [OsciRefCounterMTSA< DeallocType, LockType >](#), and [Osci\\_DefAllocWithRefCounter< DefAlloc >](#).

**7.190.3.2** virtual uint32 OsciRefCounter::getCount () [pure virtual]

Gets the current number of references

Implemented in [OsciRefCounterDA](#), [OsciRefCounterSA< DeallocType >](#), [OsciRefCounterMTDA< LockType >](#), [OsciRefCounterMTSA< DeallocType, LockType >](#), and [Osci\\_DefAllocWithRefCounter< DefAlloc >](#).

**7.190.3.3** virtual void OsciRefCounter::removeRef () [pure virtual]

Delete from reference count

Implemented in [OsciRefCounterDA](#), [OsciRefCounterSA< DeallocType >](#), [OsciRefCounterMTDA< LockType >](#), [OsciRefCounterMTSA< DeallocType, LockType >](#), and [Osci\\_DefAllocWithRefCounter< DefAlloc >](#).

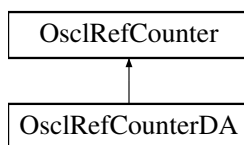
The documentation for this class was generated from the following file:

- [osci\\_refcounter.h](#)

## 7.191 OsciRefCounterDA Class Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for OsciRefCounterDA::



### Public Methods

- [OsciRefCounterDA](#) ([OsciAny](#) \*p, [OsciDestructDealloc](#) \*dealloc)
- virtual [~OsciRefCounterDA](#) ()
- void [addRef](#) ()
- void [removeRef](#) ()
- uint32 [getCount](#) ()

### 7.191.1 Detailed Description

Implementation of an [OsciRefCounter](#) that uses a dynamically created deallocator.

### 7.191.2 Constructor & Destructor Documentation

**7.191.2.1** [OsciRefCounterDA::OsciRefCounterDA](#) ([OsciAny](#) \* p, [OsciDestructDealloc](#) \* dealloc) [inline]

Constructor Takes a pointer to the buffer to track, and a pointer to the deallocator object which will be used to delete the buffer.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsciRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsciRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to delete().

#### Parameters:

- p* pointer to the buffer to track
- dealloc* pointer to the deallocator to use when deleting the buffer

**7.191.2.2** [virtual OsciRefCounterDA::~~OsciRefCounterDA](#) () [inline, virtual]

Destructor empty

### 7.191.3 Member Function Documentation

#### 7.191.3.1 void OsciRefCounterDA::addRef() [inline, virtual]

Add to the reference count

Implements [OsciRefCounter](#).

#### 7.191.3.2 uint32 OsciRefCounterDA::getCount() [inline, virtual]

Gets the current number of references

Implements [OsciRefCounter](#).

#### 7.191.3.3 void OsciRefCounterDA::removeRef() [inline, virtual]

Remove from the reference count

Implements [OsciRefCounter](#).

The documentation for this class was generated from the following file:

- [osci\\_refcounter.h](#)

## 7.192 OsciRefCounterMemFrag Class Reference

```
#include <osci_refcounter_memfrag.h>
```

### Public Methods

- [OsciRefCounterMemFrag](#) ([OsciMemoryFragment](#) &m, [OsciRefCounter](#) \*r, uint32 in\_capacity)
- [OsciRefCounterMemFrag](#) (const [OsciRefCounterMemFrag](#) &x)
- [OsciRefCounterMemFrag](#) ()
- [OsciRefCounterMemFrag](#) & [operator=](#) (const [OsciRefCounterMemFrag](#) &x)
- [~OsciRefCounterMemFrag](#) ()
- [OsciRefCounter](#) \* [getRefCounter](#) ()
- [OsciMemoryFragment](#) & [getMemFrag](#) ()
- [OsciAny](#) \* [getMemFragPtr](#) ()
- uint32 [getMemFragSize](#) ()
- uint32 [getCapacity](#) ()
- uint32 [getCount](#) ()

### 7.192.1 Detailed Description

Class to contain a memory fragment with it's associated reference counter.

### 7.192.2 Constructor & Destructor Documentation

#### 7.192.2.1 [OsciRefCounterMemFrag::OsciRefCounterMemFrag](#) ([OsciMemoryFragment](#) & m, [OsciRefCounter](#) \* r, uint32 in\_capacity) [inline]

Constructor. A valid memory fragment and reference counter are required as input. The memory fragment structure will be copied locally.

#### Parameters:

- m* reference to memory fragment
- r* pointer to the reference counter associated with the memory fragment.

#### 7.192.2.2 [OsciRefCounterMemFrag::OsciRefCounterMemFrag](#) (const [OsciRefCounterMemFrag](#) & x) [inline]

Copy constructor.

#### 7.192.2.3 [OsciRefCounterMemFrag::OsciRefCounterMemFrag](#) () [inline]

Default constructor.

#### 7.192.2.4 [OsciRefCounterMemFrag::~~OsciRefCounterMemFrag](#) () [inline]

Destructor. Removes this object's reference from the reference counter. The reference counter will not be deleted. The reference counter is designed to self-delete when it's reference count reaches 0.

## 7.192.3 Member Function Documentation

### 7.192.3.1 `uint32 OsciRefCounterMemFrag::getCapacity ()` [inline]

Returns the capacity of the memory fragment

**Returns:**

### 7.192.3.2 `uint32 OsciRefCounterMemFrag::getCount ()` [inline]

Returns the reference counter's current count.

### 7.192.3.3 `OsciMemoryFragment& OsciRefCounterMemFrag::getMemFrag ()` [inline]

Returns a reference to the contained memory fragment structure.

### 7.192.3.4 `OsciAny* OsciRefCounterMemFrag::getMemFragPtr ()` [inline]

Returns a pointer to the memory fragment data.

### 7.192.3.5 `uint32 OsciRefCounterMemFrag::getMemFragSize ()` [inline]

Returns the size of the memory fragment data which equals its filled size.

**Returns:**

### 7.192.3.6 `OsciRefCounter* OsciRefCounterMemFrag::getRefCounter ()` [inline]

Returns a pointer to the contained reference counter object

### 7.192.3.7 `OsciRefCounterMemFrag& OsciRefCounterMemFrag::operator= (const OsciRefCounterMemFrag &x)` [inline]

Assignment Operator

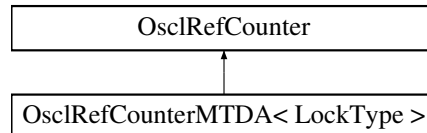
The documentation for this class was generated from the following file:

- [osci\\_refcounter\\_memfrag.h](#)

## 7.193 OsciRefCounterMTDA< LockType > Class Template Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for OsciRefCounterMTDA< LockType >::



### Public Methods

- [OsciRefCounterMTDA](#) ([OsciAny](#) \*p, [OsciDestructDealloc](#) \*dealloc)
- virtual [~OsciRefCounterMTDA](#) ()
- void [addRef](#) ()
- void [removeRef](#) ()
- uint32 [getCount](#) ()

### 7.193.1 Detailed Description

**template<class LockType> class OsciRefCounterMTDA< LockType >**

Implementation of [OsciRefCounterDA](#) for multi-threaded use. A templated lock class must be specified.

### 7.193.2 Constructor & Destructor Documentation

**7.193.2.1 template<class LockType> OsciRefCounterMTDA< LockType >::OsciRefCounterMTDA ([OsciAny](#) \*p, [OsciDestructDealloc](#) \*dealloc) [inline]**

Constructor Takes a pointer to the buffer to track, and a pointer to the deallocator object which will be used to delete the buffer.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsciRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsciRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to delete().

#### Parameters:

*p* pointer to the buffer to track

*dealloc* pointer to the deallocator to use when deleting the buffer

**7.193.2.2 template<class LockType> virtual OsciRefCounterMTDA< LockType >::~~OsciRefCounterMTDA () [inline, virtual]**

Destructor empty



### 7.193.3 Member Function Documentation

**7.193.3.1** `template<class LockType> void OsciRefCounterMTDA< LockType >::addRef ()`  
[inline, virtual]

Add to the reference count

Implements [OsciRefCounter](#).

**7.193.3.2** `template<class LockType> uint32 OsciRefCounterMTDA< LockType >::getCount ()`  
[inline, virtual]

Gets the current number of references

Implements [OsciRefCounter](#).

**7.193.3.3** `template<class LockType> void OsciRefCounterMTDA< LockType >::removeRef ()`  
[inline, virtual]

Remove from the reference count

Implements [OsciRefCounter](#).

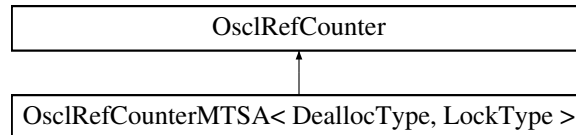
The documentation for this class was generated from the following file:

- [osci\\_refcounter.h](#)

## 7.194 OsciRefCounterMTSA< DeallocType, LockType > Class Template Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for OsciRefCounterMTSA< DeallocType, LockType >::



### Public Methods

- [OsciRefCounterMTSA](#) ([OsciAny](#) \*p)
- virtual [~OsciRefCounterMTSA](#) ()
- void [addRef](#) ()
- void [removeRef](#) ()
- uint32 [getCount](#) ()

### 7.194.1 Detailed Description

`template<class DeallocType, class LockType> class OsciRefCounterMTSA< DeallocType, LockType >`

Implementation of [OsciRefCounterSA](#) for multi-threaded use. A templated lock class must be specified.

### 7.194.2 Constructor & Destructor Documentation

**7.194.2.1** `template<class DeallocType, class LockType> OsciRefCounterMTSA< DeallocType, LockType >::OsciRefCounterMTSA (OsciAny *p) [inline]`

Constructor Takes a pointer to the buffer to track.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsciRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsciRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to delete().

#### Parameters:

*p* pointer to the buffer to track

**7.194.2.2** `template<class DeallocType, class LockType> virtual OsciRefCounterMTSA< DeallocType, LockType >::~~OsciRefCounterMTSA () [inline, virtual]`

Destructor empty

## 7.194.3 Member Function Documentation

**7.194.3.1** `template<class DeallocType, class LockType> void OsciRefCounterMTSA< DeallocType, LockType >::addRef () [inline, virtual]`

Add to the reference count

Implements [OsciRefCounter](#).

**7.194.3.2** `template<class DeallocType, class LockType> uint32 OsciRefCounterMTSA< DeallocType, LockType >::getCount () [inline, virtual]`

Gets the current number of references

Implements [OsciRefCounter](#).

**7.194.3.3** `template<class DeallocType, class LockType> void OsciRefCounterMTSA< DeallocType, LockType >::removeRef () [inline, virtual]`

Remove from the reference count

Implements [OsciRefCounter](#).

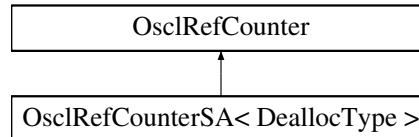
The documentation for this class was generated from the following file:

- [osci\\_refcounter.h](#)

## 7.195 OsciRefCounterSA< DeallocType > Class Template Reference

```
#include <osci_refcounter.h>
```

Inheritance diagram for OsciRefCounterSA< DeallocType >::



### Public Methods

- [OsciRefCounterSA](#) ([OsciAny](#) \*p)
- virtual [~OsciRefCounterSA](#) ()
- void [addRef](#) ()
- void [removeRef](#) ()
- uint32 [getCount](#) ()

### 7.195.1 Detailed Description

**template<class DeallocType> class OsciRefCounterSA< DeallocType >**

Implementation of an [OsciRefCounter](#) that uses a statically created deallocator.

### 7.195.2 Constructor & Destructor Documentation

**7.195.2.1 template<class DeallocType> OsciRefCounterSA< DeallocType >::OsciRefCounterSA**  
([OsciAny](#) \*p) [inline]

Constructor Takes a pointer to the buffer to track.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsciRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsciRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to delete().

#### Parameters:

*p* pointer to the buffer to track

**7.195.2.2 template<class DeallocType> virtual OsciRefCounterSA< DeallocType**  
**>::~~OsciRefCounterSA** () [inline, virtual]

Destructor empty

## 7.195.3 Member Function Documentation

**7.195.3.1** `template<class DeallocType> void OsciRefCounterSA< DeallocType >::addRef ()`  
[inline, virtual]

Add to the reference count

Implements [OsciRefCounter](#).

**7.195.3.2** `template<class DeallocType> uint32 OsciRefCounterSA< DeallocType >::getCount ()`  
[inline, virtual]

Gets the current number of references

Implements [OsciRefCounter](#).

**7.195.3.3** `template<class DeallocType> void OsciRefCounterSA< DeallocType >::removeRef ()`  
[inline, virtual]

Remove from the reference count

Implements [OsciRefCounter](#).

The documentation for this class was generated from the following file:

- [osci\\_refcounter.h](#)

## 7.196 OslRegistryAccessClient Class Reference

```
#include <osl_registry_access_client.h>
```

### Public Methods

- OSCL\_IMPORT\_REF [OslRegistryAccessClient](#) ()
- OSCL\_IMPORT\_REF [~OslRegistryAccessClient](#) ()
- OSCL\_IMPORT\_REF int32 [Connect](#) ()
- OSCL\_IMPORT\_REF [OslComponentFactory](#) [GetFactory](#) ([OSCL\\_String](#) &aComponent)
- OSCL\_IMPORT\_REF void [GetFactories](#) ([OSCL\\_String](#) &aRegistry, [Osl\\_Vector](#)< [OslRegistryAccessElement](#), [OslMemAllocator](#) > &aVec)
- OSCL\_IMPORT\_REF void [Close](#) ()

### 7.196.1 Constructor & Destructor Documentation

**7.196.1.1** OSCL\_IMPORT\_REF [OslRegistryAccessClient::OslRegistryAccessClient](#) ()

**7.196.1.2** OSCL\_IMPORT\_REF [OslRegistryAccessClient::~~OslRegistryAccessClient](#) ()

### 7.196.2 Member Function Documentation

**7.196.2.1** OSCL\_IMPORT\_REF void [OslRegistryAccessClient::Close](#) ()

Close and cleanup session.

**7.196.2.2** OSCL\_IMPORT\_REF int32 [OslRegistryAccessClient::Connect](#) ()

Create a session.

#### Returns:

OslErrNone on success.

**7.196.2.3** OSCL\_IMPORT\_REF void [OslRegistryAccessClient::GetFactories](#) ([OSCL\\_String](#) &*aRegistry*, [Osl\\_Vector](#)< [OslRegistryAccessElement](#), [OslMemAllocator](#) > &*aVec*)

Get all factories for a given registry type.

#### Parameters:

*aRegistry*: registry MIME type

*aVec*: output component factory + mimestring vector.

**7.196.2.4** OSCL\_IMPORT\_REF [OslComponentFactory](#) [OslRegistryAccessClient::GetFactory](#) ([OSCL\\_String](#) &*aComponent*)

Lookup a factory by registry and component mime type.

**Parameters:**

*aComponent:* registry+component MIME type

**Returns:**

Factory. Factory will be NULL if not found.

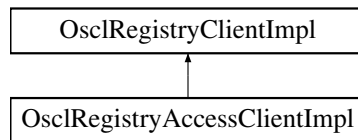
The documentation for this class was generated from the following file:

- [osci\\_registry\\_access\\_client.h](#)

## 7.197 OsciRegistryAccessClientImpl Class Reference

```
#include <osci_registry_client_impl.h>
```

Inheritance diagram for OsciRegistryAccessClientImpl::



The documentation for this class was generated from the following file:

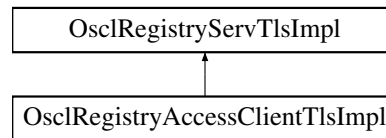
- [osci\\_registry\\_client\\_impl.h](#)



## 7.198 OsciRegistryAccessClientTlsImpl Class Reference

```
#include <osci_registry_client_impl.h>
```

Inheritance diagram for OsciRegistryAccessClientTlsImpl::



The documentation for this class was generated from the following file:

- [osci\\_registry\\_client\\_impl.h](#)

## 7.199 OslRegistryAccessElement Class Reference

```
#include <osl_registry_types.h>
```

### Data Fields

- [OslComponentFactory](#) iFactory
- [OSCL\\_HeapString< OslMemAllocator >](#) iMimeString

### 7.199.1 Detailed Description

A class used to access the registry data

### 7.199.2 Field Documentation

**7.199.2.1** [OslComponentFactory](#) OslRegistryAccessElement::iFactory

**7.199.2.2** [OSCL\\_HeapString<OslMemAllocator>](#) OslRegistryAccessElement::iMimeString

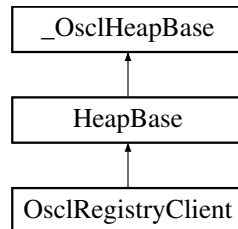
The documentation for this class was generated from the following file:

- [osl\\_registry\\_types.h](#)

## 7.200 OsciRegistryClient Class Reference

```
#include <osci_registry_client.h>
```

Inheritance diagram for OsciRegistryClient::



### Public Methods

- OSCL\_IMPORT\_REF [OsciRegistryClient](#) ()
- OSCL\_IMPORT\_REF [~OsciRegistryClient](#) ()
- OSCL\_IMPORT\_REF int32 [Connect](#) (bool aPerThread=false)
- OSCL\_IMPORT\_REF int32 [Register](#) (OSCL\_String &aComponentID, [OsciComponentFactory](#) 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 OsciRegistryClient::OsciRegistryClient ()

7.200.1.2 OSCL\_IMPORT\_REF OsciRegistryClient::~~OsciRegistryClient ()

### 7.200.2 Member Function Documentation

7.200.2.1 OSCL\_IMPORT\_REF void OsciRegistryClient::Close ()

Close and cleanup. All components registered in this session are automatically unregistered.

7.200.2.2 OSCL\_IMPORT\_REF int32 OsciRegistryClient::Connect (bool *aPerThread* = false)

Create a session.

#### Parameters:

***aPerThread***: Select per-thread registry instead of global registry.

#### Returns:

OsciErrNone on success.

**7.200.2.3 OSCL\_IMPORT\_REF int32 OsciRegistryClient::Register ([OSCL\\_String](#) & *aComponentID*, [OsciComponentFactory](#) *aFactory*)**

Register a component factory by registry ID and component ID.

**Parameters:**

*aComponentID*: registry + component mime-string.

*aFactory*: factory function pointer.

*aParam*: component Create param.

**Returns:**

OsciErrNone on success.

**7.200.2.4 OSCL\_IMPORT\_REF int32 OsciRegistryClient::UnRegister ([OSCL\\_String](#) & *aComponentID*)**

Unregister a previously registered component.

**Returns:**

OsciErrNone on success.

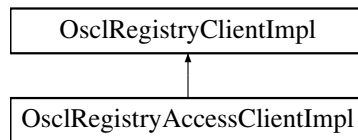
The documentation for this class was generated from the following file:

- [osci\\_registry\\_client.h](#)

## 7.201 OsciRegistryClientImpl Class Reference

```
#include <osci_registry_client_impl.h>
```

Inheritance diagram for OsciRegistryClientImpl::



### Protected Methods

- int32 [Connect](#) ()
- void [Close](#) ()
- int32 [Register](#) (OSCL\_String &, OsciComponentFactory)
- int32 [UnRegister](#) (OSCL\_String &)
- OsciComponentFactory [GetFactory](#) (OSCL\_String &)
- void [GetFactories](#) (OSCL\_String &, Osci\_Vector< OsciRegistryAccessElement, OsciMemAllocator > &)

### Friends

- class [OsciRegistryClient](#)
- class [OsciRegistryAccessClient](#)

## 7.201.1 Member Function Documentation

**7.201.1.1** void OslRegistryClientImpl::Close (void) [inline, protected]

**7.201.1.2** int32 OslRegistryClientImpl::Connect () [inline, protected]

**7.201.1.3** void OslRegistryClientImpl::GetFactories (OSCL\_String &, Osl\_Vector<  
OslRegistryAccessElement, OslMemAllocator > &) [inline, protected]

**7.201.1.4** OslComponentFactory OslRegistryClientImpl::GetFactory (OSCL\_String &)  
[inline, protected]

**7.201.1.5** int32 OslRegistryClientImpl::Register (OSCL\_String &, OslComponentFactory)  
[inline, protected]

**7.201.1.6** int32 OslRegistryClientImpl::UnRegister (OSCL\_String &) [inline,  
protected]

## 7.201.2 Friends And Related Function Documentation

**7.201.2.1** friend class OslRegistryAccessClient [friend]

**7.201.2.2** friend class OslRegistryClient [friend]

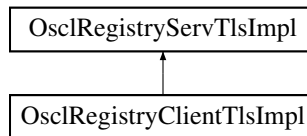
The documentation for this class was generated from the following file:

- [osl\\_registry\\_client\\_impl.h](#)

## 7.202 OsciRegistryClientTlsImpl Class Reference

```
#include <osci_registry_client_impl.h>
```

Inheritance diagram for OsciRegistryClientTlsImpl::



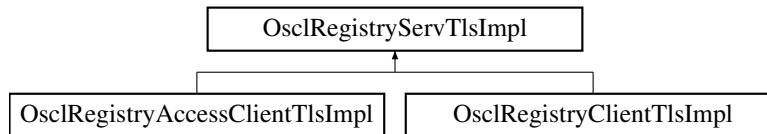
The documentation for this class was generated from the following file:

- [osci\\_registry\\_client\\_impl.h](#)

## 7.203 OsciRegistryServTlsImpl Class Reference

```
#include <osci_registry_serv_impl_tls.h>
```

Inheritance diagram for OsciRegistryServTlsImpl::



### Protected Methods

- [OsciRegistryServTlsImpl \(\)](#)
- virtual [~OsciRegistryServTlsImpl \(\)](#)
- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL\\_String &aComponentID, OsciComponentFactory aFactory\)](#)
- int32 [UnRegister \(OSCL\\_String &aComponentID\)](#)
- [OsciComponentFactory GetFactory \(OSCL\\_String &aComponent\)](#)
- void [GetFactories \(OSCL\\_String &aRegistry, Osci\\_Vector< OsciRegistryAccessElement, Osci-MemAllocator > &aVec\)](#)

### Friends

- class [OsciRegistryClient](#)
- class [OsciRegistryAccessClient](#)



## 7.203.1 Constructor & Destructor Documentation

7.203.1.1 **OsciRegistryServTlsImpl::OsciRegistryServTlsImpl ()** [protected]

7.203.1.2 **virtual OsciRegistryServTlsImpl::~~OsciRegistryServTlsImpl ()** [protected, virtual]

## 7.203.2 Member Function Documentation

7.203.2.1 **void OsciRegistryServTlsImpl::Close ()** [protected]

7.203.2.2 **int32 OsciRegistryServTlsImpl::Connect ()** [protected]

7.203.2.3 **void OsciRegistryServTlsImpl::GetFactories (OSCL\_String & aRegistry, Osci\_Vector< OsciRegistryAccessElement, OsciMemAllocator > & aVec)** [protected]

7.203.2.4 **OsciComponentFactory OsciRegistryServTlsImpl::GetFactory (OSCL\_String & aComponent)** [protected]

7.203.2.5 **int32 OsciRegistryServTlsImpl::Register (OSCL\_String & aComponentID, OsciComponentFactory aFactory)** [protected]

7.203.2.6 **int32 OsciRegistryServTlsImpl::UnRegister (OSCL\_String & aComponentID)** [protected]

## 7.203.3 Friends And Related Function Documentation

7.203.3.1 **friend class OsciRegistryAccessClient** [friend]

7.203.3.2 **friend class OsciRegistryClient** [friend]

The documentation for this class was generated from the following file:

- [osci\\_registry\\_serv\\_impl\\_tls.h](#)

## 7.204 OsciScheduler Class Reference

```
#include <osci_scheduler.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF void [Init](#) (const char \*name, [Osci\\_DefAlloc](#) \*alloc=NULL, int nreserve=20)
- OSCL\_IMPORT\_REF void [Cleanup](#) ()

### 7.204.1 Detailed Description

Per-thread scheduler initialization and cleanup.

### 7.204.2 Member Function Documentation

#### 7.204.2.1 OSCL\_IMPORT\_REF void OsciScheduler::Cleanup () [static]

This routine uninstalls and destroys Osci scheduler for the calling thread.

#### 7.204.2.2 OSCL\_IMPORT\_REF void OsciScheduler::Init (const char \* name, [Osci\\_DefAlloc](#) \* alloc = NULL, int nreserve = 20) [static]

This routine creates and installs a scheduler in the calling thread.

#### Parameters:

- name:** (input param) scheduler name.
- alloc:** (input param) optional allocator to use for the internal implementation.
- nreserve:** (input param) optional value for ready queue reserve size.

The documentation for this class was generated from the following file:

- [osci\\_scheduler.h](#)

## 7.205 OsciSchedulerObserver Class Reference

```
#include <osci_scheduler.h>
```

### Public Methods

- virtual void [OsciSchedulerTimerCallback](#) ([OsciAny](#) \*aContext, uint32 aDelayMsec)=0
- virtual void [OsciSchedulerReadyCallback](#) ([OsciAny](#) \*aContext)=0
- virtual [~OsciSchedulerObserver](#) ()

### 7.205.1 Detailed Description

OsciSchedulerObserver 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 [OsciSchedulerObserver::~OsciSchedulerObserver](#) () [inline, virtual]

### 7.205.3 Member Function Documentation

**7.205.3.1** virtual void [OsciSchedulerObserver::OsciSchedulerReadyCallback](#) ([OsciAny](#) \*aContext) [pure virtual]

OsciSchedulerReadyCallback 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 [OsciSchedulerObserver::OsciSchedulerTimerCallback](#) ([OsciAny](#) \*aContext, uint32 aDelayMsec) [pure virtual]

OsciSchedulerTimerCallback 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:

- [osci\\_scheduler.h](#)

## 7.206 OsciScopedLock< LockClass > Class Template Reference

The OsciScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsciScopedLock goes out of scope.

```
#include <osci_lock_base.h>
```

### Public Methods

- [OsciScopedLock](#) (LockClass &inLock)  
*Default constructor Initializes the pointer and takes ownership.*
- [~OsciScopedLock](#) ()  
*Destructor.*

### 7.206.1 Detailed Description

```
template<class LockClass> class OsciScopedLock< LockClass >
```

The OsciScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsciScopedLock goes out of scope.

The purpose of this class is to provide a way to prevent accidental resource leaks in a class or a method, due to "not remembering to unlock" variables which might lead to deadlock conditions.

### 7.206.2 Constructor & Destructor Documentation

**7.206.2.1** `template<class LockClass> OsciScopedLock< LockClass >::OsciScopedLock (LockClass &inLock) [inline, explicit]`

Default constructor Initializes the pointer and takes ownership.

**7.206.2.2** `template<class LockClass> OsciScopedLock< LockClass >::~~OsciScopedLock () [inline]`

Destructor.

The pointer is deleted in case this class still has ownership

The documentation for this class was generated from the following file:

- [osci\\_lock\\_base.h](#)

## 7.207 OslSelect Class Reference

```
#include <osl_init.h>
```

### Public Methods

- [OslSelect](#) ()
- [OslSelect](#) ([Osl\\_DefAlloc](#) \*erralloc, [Osl\\_DefAlloc](#) \*schedalloc, const char \*name, int32 reserve=10, bool heapcheck=false, FILE \*output=NULL)

### Data Fields

- bool [iOslBase](#)
- bool [iOslMemory](#)
- bool [iOslErrorTrap](#)
- bool [iOslLogger](#)
- bool [iOslScheduler](#)
- [Osl\\_DefAlloc](#) \* [iErrAlloc](#)
- [Osl\\_DefAlloc](#) \* [iSchedulerAlloc](#)
- const char \* [iSchedulerName](#)
- int32 [iSchedulerReserve](#)
- bool [iHeapCheck](#)
- FILE \* [iOutputFile](#)

### 7.207.1 Detailed Description

Osl Module selection and Init/Cleanup options.

## 7.207.2 Constructor & Destructor Documentation

7.207.2.1 `OsciSelect::OsciSelect ()` [inline]

7.207.2.2 `OsciSelect::OsciSelect (Osci_DefAlloc * erralloc, Osci_DefAlloc * schedalloc, const char * name, int32 reserve = 10, bool heapcheck = false, FILE * output = NULL)` [inline]

## 7.207.3 Field Documentation

7.207.3.1 `Osci_DefAlloc*` `OsciSelect::iErrAlloc`

7.207.3.2 `bool` `OsciSelect::iHeapCheck`

7.207.3.3 `bool` `OsciSelect::iOsciBase`

7.207.3.4 `bool` `OsciSelect::iOsciErrorTrap`

7.207.3.5 `bool` `OsciSelect::iOsciLogger`

7.207.3.6 `bool` `OsciSelect::iOsciMemory`

7.207.3.7 `bool` `OsciSelect::iOsciScheduler`

7.207.3.8 `FILE*` `OsciSelect::iOutputFile`

7.207.3.9 `Osci_DefAlloc*` `OsciSelect::iSchedulerAlloc`

7.207.3.10 `const char*` `OsciSelect::iSchedulerName`

7.207.3.11 `int32` `OsciSelect::iSchedulerReserve`

The documentation for this class was generated from the following file:

- [osci\\_init.h](#)

## 7.208 OsciSemaphore Class Reference

```
#include <osci_semaphore.h>
```

### Public Methods

- OSCL\_IMPORT\_REF [OsciSemaphore \(\)](#)
- OSCL\_IMPORT\_REF [~OsciSemaphore \(\)](#)
- OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError Create](#) (uint32 initVal=0)
- OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError Close](#) ()
- OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError Wait](#) ()
- OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError Wait](#) (uint32 timeout\_msec)
- OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError TryWait](#) ()
- OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError Signal](#) ()

### 7.208.1 Detailed Description

Class Semaphore

### 7.208.2 Constructor & Destructor Documentation

#### 7.208.2.1 OSCL\_IMPORT\_REF OsciSemaphore::OsciSemaphore ()

Class constructor

#### 7.208.2.2 OSCL\_IMPORT\_REF OsciSemaphore::~~OsciSemaphore ()

Class destructor

### 7.208.3 Member Function Documentation

#### 7.208.3.1 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciSemaphore::Close ()

Closes the Semaphore

##### Parameters:

*It* wont take any parameters

##### Returns:

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

#### 7.208.3.2 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciSemaphore::Create (uint32 initVal = 0)

Creates the Semaphore

##### Parameters:

*Intialcount*

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

**7.208.3.3 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciSemaphore::Signal ()**

Signals that the thread is finished with the Semaphore

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

**7.208.3.4 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciSemaphore::TryWait ()**

Try to acquire semaphore ,if the semaphore is already acquired by another thread, calling thread immediately returns with out blocking

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns SUCCESS\_ERROR if the semaphore was acquired, SEM\_LOCKED\_ERROR if the semaphore cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.208.3.5 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciSemaphore::Wait (uint32 *timeout\_msec*)**

Makes the thread to wait on the Semaphore, with a timeout.

**Parameters:**

*timeout* in milliseconds.

**Returns:**

Returns SUCCESS\_ERROR if the semaphore was aquired, WAIT\_TIMEOUT\_ERROR if the timeout expired without acquiring the semaphore, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.208.3.6 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciSemaphore::Wait ()**

Makes the thread to wait on the Semaphore

**Parameters:**

*It* wont take any parameters

**Returns:**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

The documentation for this class was generated from the following file:

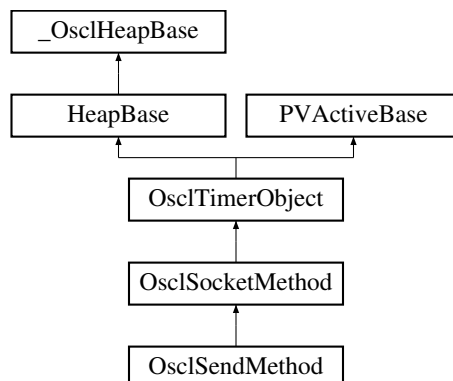
- [osci\\_semaphore.h](#)



## 7.209 OsciSendMethod Class Reference

```
#include <osci_socket_send.h>
```

Inheritance diagram for OsciSendMethod::



### Public Methods

- [~OsciSendMethod \(\)](#)
- [TPVSocketEvent Send](#) (const uint8 \*&aPtr, uint32 aLen, int32 aTimeout)
- uint8 \* [GetSendData](#) (int32 \*aLength)
- [OsciSendRequest](#) \* [SendRequest](#) ()

### Static Public Methods

- OsciSendMethod \* [NewL](#) ([OsciIPSocketI](#) &c)

### 7.209.1 Constructor & Destructor Documentation

#### 7.209.1.1 OsciSendMethod::~~OsciSendMethod ()

### 7.209.2 Member Function Documentation

#### 7.209.2.1 uint8\* OsciSendMethod::GetSendData (int32 \*aLength)

#### 7.209.2.2 OsciSendMethod\* OsciSendMethod::NewL ([OsciIPSocketI](#) &c) [static]

#### 7.209.2.3 [TPVSocketEvent](#) OsciSendMethod::Send (const uint8 \*&aPtr, uint32 aLen, int32 aTimeout)

#### 7.209.2.4 [OsciSendRequest](#)\* OsciSendMethod::SendRequest () [inline]

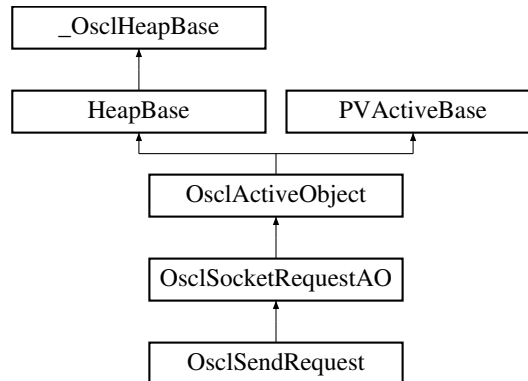
The documentation for this class was generated from the following file:

- [osci\\_socket\\_send.h](#)

## 7.210 OsciSendRequest Class Reference

```
#include <osci_socket_send.h>
```

Inheritance diagram for OsciSendRequest::



### Public Methods

- [OsciSendRequest](#) ([OsciSocketMethod](#) &c)
- void [Send](#) (const uint8 \*aPtr, uint32 aLen)
- void [Success](#) ()
- uint8 \* [GetSendData](#) (int32 \*aLength)

### 7.210.1 Constructor & Destructor Documentation

7.210.1.1 [OsciSendRequest::OsciSendRequest](#) ([OsciSocketMethod](#) & c) [inline]

### 7.210.2 Member Function Documentation

7.210.2.1 uint8\* [OsciSendRequest::GetSendData](#) (int32 \* aLength)

7.210.2.2 void [OsciSendRequest::Send](#) (const uint8 \*& aPtr, uint32 aLen)

7.210.2.3 void [OsciSendRequest::Success](#) () [virtual]

Reimplemented from [OsciSocketRequestAO](#).

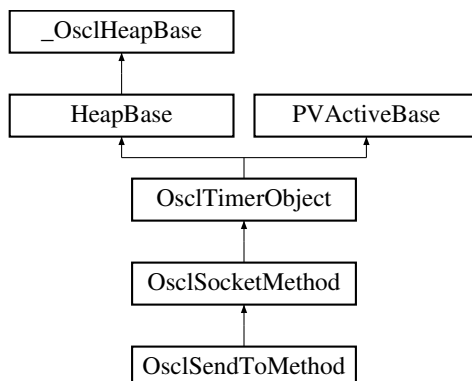
The documentation for this class was generated from the following file:

- [osci\\_socket\\_send.h](#)

## 7.211 OsciSendToMethod Class Reference

```
#include <osci_socket_send_to.h>
```

Inheritance diagram for OsciSendToMethod::



### Public Methods

- [~OsciSendToMethod \(\)](#)
- [TPVSocketEvent SendTo](#) (const uint8 \*&aPtr, uint32 aLen, [OsciNetworkAddress](#) &aAddress, int32 aTimeout)
- uint8 \* [GetSendData](#) (int32 \*aLength)
- [OsciSendToRequest](#) \* [SendToRequest](#) ()

### Static Public Methods

- OsciSendToMethod \* [NewL](#) ([OsciIPSocketI](#) &c)

### 7.211.1 Constructor & Destructor Documentation

#### 7.211.1.1 OsciSendToMethod::~~OsciSendToMethod ()

### 7.211.2 Member Function Documentation

#### 7.211.2.1 uint8\* OsciSendToMethod::GetSendData (int32 \*aLength)

#### 7.211.2.2 OsciSendToMethod\* OsciSendToMethod::NewL ([OsciIPSocketI](#) &c) [static]

#### 7.211.2.3 [TPVSocketEvent](#) OsciSendToMethod::SendTo (const uint8 \*&aPtr, uint32 aLen, [OsciNetworkAddress](#) &aAddress, int32 aTimeout)

#### 7.211.2.4 [OsciSendToRequest](#)\* OsciSendToMethod::SendToRequest () [inline]

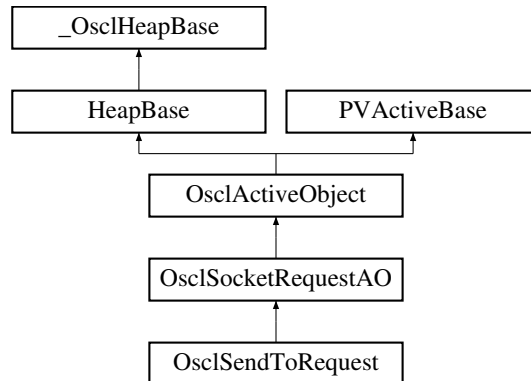
The documentation for this class was generated from the following file:

- [osci\\_socket\\_send\\_to.h](#)

## 7.212 OsciSendToRequest Class Reference

```
#include <osci_socket_send_to.h>
```

Inheritance diagram for OsciSendToRequest::



### Public Methods

- [OsciSendToRequest](#) ([OsciSocketMethod](#) &c)
- void [SendTo](#) (const uint8 \*&aPtr, uint32 aLen, [OsciNetworkAddress](#) &aAddress)
- void [Success](#) ()
- uint8 \* [GetSendData](#) (int32 \*aLength)

### 7.212.1 Detailed Description

This is the AO that interacts with the socket server

### 7.212.2 Constructor & Destructor Documentation

**7.212.2.1** [OsciSendToRequest::OsciSendToRequest](#) ([OsciSocketMethod](#) &c) [inline]

### 7.212.3 Member Function Documentation

**7.212.3.1** uint8\* [OsciSendToRequest::GetSendData](#) (int32 \*aLength)

**7.212.3.2** void [OsciSendToRequest::SendTo](#) (const uint8 \*&aPtr, uint32 aLen, [OsciNetworkAddress](#) &aAddress)

**7.212.3.3** void [OsciSendToRequest::Success](#) () [virtual]

Reimplemented from [OsciSocketRequestAO](#).

The documentation for this class was generated from the following file:

- [osci\\_socket\\_send\\_to.h](#)

## 7.213 OsciSharedPtr< TheClass > Class Template Reference

A parameterized smart pointer class.

```
#include <osci_shared_ptr.h>
```

### Public Methods

- [OsciSharedPtr](#) ()  
*Constructor.*
- [OsciSharedPtr](#) (TheClass \*inClassPtr, [OsciRefCount](#) \*in\_refcnt)  
*Constructor.*
- [OsciSharedPtr](#) (const OsciSharedPtr &inSharedPtr)  
*Copy constructor.*
- virtual [~OsciSharedPtr](#) ()  
*Destructor.*
- TheClass \* [operator →](#) ()
- TheClass & [operator \\*](#) ()  
*The indirection operator returns a reference to an object of the parameterized type.*
- [operator TheClass \\*](#) ()  
*Casting operator.*
- TheClass \* [GetRep](#) ()  
*Use this function to get a pointer to the wrapped object.*
- [OsciRefCount](#) \* [GetRefCount](#) ()  
*Get the refcount pointer. This should primarily be used for conversion operations.*
- int [get\\_count](#) ()  
*Get a count of how many references to the object exist.*
- void [Bind](#) (const OsciSharedPtr &inHandle)  
*Use this function to bind an existing OsciSharedPtr to a already-wrapped object.*
- void [Bind](#) (TheClass \*ptr, [OsciRefCount](#) \*in\_refcnt)  
*Use this function to bind an existing OsciSharedPtr to a new (unwrapped) object.*
- void [Unbind](#) ()  
*Use this function of unbind an existing OsciSharedPtr.*
- OsciSharedPtr & [operator=](#) (const OsciSharedPtr &inSharedPtr)  
*Assignment operator.*
- bool [operator==](#) (const OsciSharedPtr &b) const  
*Test for equality to see if two PVHandles wrap the same object.*

## 7.213.1 Detailed Description

**template<class TheClass> class OsciSharedPtr< TheClass >**

A parameterized smart pointer class.

## 7.213.2 Constructor & Destructor Documentation

**7.213.2.1 template<class TheClass> OsciSharedPtr< TheClass >::OsciSharedPtr () [inline]**

Constructor.

**7.213.2.2 template<class TheClass> OsciSharedPtr< TheClass >::OsciSharedPtr (TheClass \* *inClassPtr*, *OsciRefCounter* \* *in\_refcnt*) [inline]**

Constructor.

### Parameters:

*inClassPtr* A pointer to an instance of the parameterized type that the new OsciSharedPtr will wrap.

**7.213.2.3 template<class TheClass> OsciSharedPtr< TheClass >::OsciSharedPtr (const OsciSharedPtr< TheClass > & *inSharedPtr*) [inline]**

Copy constructor.

**7.213.2.4 template<class TheClass> virtual OsciSharedPtr< TheClass >::~~OsciSharedPtr () [inline, virtual]**

Destructor.

## 7.213.3 Member Function Documentation

**7.213.3.1 template<class TheClass> int OsciSharedPtr< TheClass >::get\_count () [inline]**

Get a count of how many references to the object exist.

**7.213.3.2 template<class TheClass> *OsciRefCounter*\* OsciSharedPtr< TheClass >::GetRefCounter () [inline]**

Get the refcount pointer. This should primarily be used for conversion operations.

**7.213.3.3 template<class TheClass> TheClass\* OsciSharedPtr< TheClass >::GetRep () [inline]**

Use this function to get a pointer to the wrapped object.

**7.213.3.4** `template<class TheClass> TheClass& OsciSharedPtr< TheClass >::operator * ()`  
[inline]

The indirection operator returns a reference to an object of the parameterized type.

**7.213.3.5** `template<class TheClass> OsciSharedPtr< TheClass >::operator TheClass * ()`  
[inline]

Casting operator.

**7.213.3.6** `template<class TheClass> TheClass* OsciSharedPtr< 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> OsciSharedPtr& OsciSharedPtr< TheClass >::operator=`  
`(const OsciSharedPtr< TheClass > & inSharedPtr) [inline]`

Assignment operator.

**7.213.3.8** `template<class TheClass> void OsciSharedPtr< TheClass >::Unbind () [inline]`

Use this function of unbind an existing OsciSharedPtr.

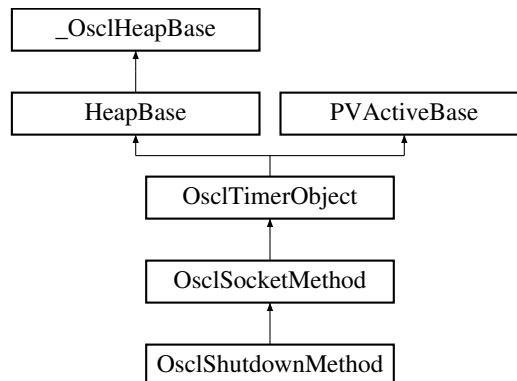
The documentation for this class was generated from the following file:

- [osci\\_shared\\_ptr.h](#)

## 7.214 OsciShutdownMethod Class Reference

```
#include <osci_socket_shutdown.h>
```

Inheritance diagram for OsciShutdownMethod::



### Public Methods

- [~OsciShutdownMethod \(\)](#)
- [TPVSocketEvent Shutdown \(TPVSocketShutdown aHow, int32 aTimeout\)](#)
- [OsciShutdownRequest \\* ShutdownRequest \(\)](#)

### Static Public Methods

- [OsciShutdownMethod \\* NewL \(OsciIPSocketI &c\)](#)

### 7.214.1 Constructor & Destructor Documentation

#### 7.214.1.1 OsciShutdownMethod::~~OsciShutdownMethod ()

### 7.214.2 Member Function Documentation

#### 7.214.2.1 OsciShutdownMethod\* OsciShutdownMethod::NewL ([OsciIPSocketI](#) &c) [static]

#### 7.214.2.2 [TPVSocketEvent](#) OsciShutdownMethod::Shutdown ([TPVSocketShutdown](#) aHow, int32 aTimeout)

#### 7.214.2.3 [OsciShutdownRequest](#)\* OsciShutdownMethod::ShutdownRequest () [inline]

The documentation for this class was generated from the following file:

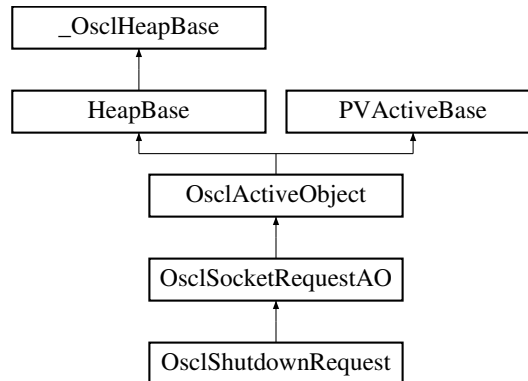
- [osci\\_socket\\_shutdown.h](#)



## 7.215 OsciShutdownRequest Class Reference

```
#include <osci_socket_shutdown.h>
```

Inheritance diagram for OsciShutdownRequest::



### Public Methods

- [OsciShutdownRequest](#) ([OsciSocketMethod](#) &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** [OsciShutdownRequest::OsciShutdownRequest](#) ([OsciSocketMethod](#) & c) [inline]

### 7.215.3 Member Function Documentation

**7.215.3.1** void [OsciShutdownRequest::Shutdown](#) ([TPVSocketShutdown](#) aHow)

The documentation for this class was generated from the following file:

- [osci\\_socket\\_shutdown.h](#)

## 7.216 OsciSingleton< T, ID, Registry > Class Template Reference

```
#include <osci_singleton.h>
```

### Public Methods

- [OsciSingleton](#) ()
- [~OsciSingleton](#) ()
- 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 = OsciSingletonRegistry> class OsciSingleton< T, ID, Registry >
```

### 7.216.1 Constructor & Destructor Documentation

7.216.1.1 `template<class T, uint32 ID, class Registry = OsciSingletonRegistry> OsciSingleton< T, ID, Registry >::OsciSingleton () [inline]`

7.216.1.2 `template<class T, uint32 ID, class Registry = OsciSingletonRegistry> OsciSingleton< T, ID, Registry >::~~OsciSingleton () [inline]`

### 7.216.2 Member Function Documentation

7.216.2.1 `template<class T, uint32 ID, class Registry = OsciSingletonRegistry> T& OsciSingleton< T, ID, Registry >::operator * () const [inline]`

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsciSingleton can be used like the regular pointer that it was initialized with.

7.216.2.2 `template<class T, uint32 ID, class Registry = OsciSingletonRegistry> T* OsciSingleton< T, ID, Registry >::operator → () const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsciSingleton can be used like the regular pointer that it was initialized with.

**7.216.2.3** `template<class T, uint32 ID, class Registry = OsciSingletonRegistry> bool  
OsciSingleton< T, ID, Registry >::set () [inline]`

`set()` method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

### 7.216.3 Field Documentation

**7.216.3.1** `template<class T, uint32 ID, class Registry = OsciSingletonRegistry> T*  
OsciSingleton< T, ID, Registry >::_Ptr [protected]`

The documentation for this class was generated from the following file:

- [osci\\_singleton.h](#)

## 7.217 OsciSingletonRegistry Class Reference

```
#include <osci_singleton.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF [OsciAny](#) \* [getInstance](#) (uint32 ID, int32 &error)
- OSCL\_IMPORT\_REF void [registerInstance](#) ([OsciAny](#) \*ptr, uint32 ID, int32 &error)
- OSCL\_IMPORT\_REF [OsciAny](#) \* [lockAndGetInstance](#) (uint32 ID, int32 &error)
- OSCL\_IMPORT\_REF void [registerInstanceAndUnlock](#) ([OsciAny](#) \*ptr, uint32 ID, int32 &error)

### Friends

- class [OsciBase](#)

### 7.217.1 Member Function Documentation

7.217.1.1 OSCL\_IMPORT\_REF [OsciAny](#)\* [OsciSingletonRegistry::getInstance](#) (uint32 *ID*, int32 & *error*) [static]

7.217.1.2 OSCL\_IMPORT\_REF [OsciAny](#)\* [OsciSingletonRegistry::lockAndGetInstance](#) (uint32 *ID*, int32 & *error*) [static]

7.217.1.3 OSCL\_IMPORT\_REF void [OsciSingletonRegistry::registerInstance](#) ([OsciAny](#) \* *ptr*, uint32 *ID*, int32 & *error*) [static]

7.217.1.4 OSCL\_IMPORT\_REF void [OsciSingletonRegistry::registerInstanceAndUnlock](#) ([OsciAny](#) \* *ptr*, uint32 *ID*, int32 & *error*) [static]

### 7.217.2 Friends And Related Function Documentation

7.217.2.1 friend class [OsciBase](#) [friend]

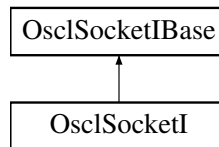
The documentation for this class was generated from the following file:

- [osci\\_singleton.h](#)

## 7.218 OsciSocketI Class Reference

```
#include <osci_socket_imp_pv.h>
```

Inheritance diagram for OsciSocketI::



### Public Methods

- [~OsciSocketI](#) ()
- [int32 Open](#) ([OsciSocketServI](#) &aServer, uint32 addrFamily, uint32 sockType, uint32 protocol)
- [int32 Open](#) ([OsciSocketServI](#) &aServer)
- [int32 Bind](#) ([OsciNetworkAddress](#) &anAddr)
- [int32 SetSockOpt](#) ([TPVSocketOptionLevel](#) aOptionLevel, [TPVSocketOptionName](#) aOptionName, [OsciAny](#) \*aOptionValue, int32 aOptionLen)
- [int32 GetPeerName](#) ([OsciNetworkAddress](#) &peerName)
- [int32 Join](#) ([OsciNetworkAddress](#) &anAddr)
- [int32 Close](#) ()
- [int32 Listen](#) (uint32 qSize)
- [int32 SetRecvBufferSize](#) (uint32 size)
- [TPVSocketEvent ThreadLogoff](#) ()
- [TPVSocketEvent ThreadLogon](#) ([OsciSocketServI](#) \*aServ)
- [void Connect](#) ([ConnectParam](#) &, [OsciSocketRequestAO](#) &)
- [void Accept](#) ([AcceptParam](#) &, [OsciSocketRequestAO](#) &)
- [void Shutdown](#) ([ShutdownParam](#) &, [OsciSocketRequestAO](#) &)
- [void Send](#) ([SendParam](#) &, [OsciSocketRequestAO](#) &)
- [void SendSuccess](#) ([SendParam](#) &)
- [void SendTo](#) ([SendToParam](#) &, [OsciSocketRequestAO](#) &)
- [void SendToSuccess](#) ([SendToParam](#) &)
- [void Recv](#) ([RecvParam](#) &, [OsciSocketRequestAO](#) &)
- [void RecvSuccess](#) ([RecvParam](#) &)
- [void RecvFrom](#) ([RecvFromParam](#) &, [OsciSocketRequestAO](#) &)
- [void RecvFromSuccess](#) ([RecvFromParam](#) &)
- [TOsciSocket Socket](#) ()
- [void ProcessConnect](#) ([OsciSocketServRequestQElem](#) \*)
- [void ProcessShutdown](#) ([OsciSocketServRequestQElem](#) \*)
- [void ProcessAccept](#) ([OsciSocketServRequestQElem](#) \*)
- [void ProcessSendTo](#) ([OsciSocketServRequestQElem](#) \*)
- [void ProcessRecvFrom](#) ([OsciSocketServRequestQElem](#) \*)
- [void ProcessSend](#) ([OsciSocketServRequestQElem](#) \*)
- [void ProcessRecv](#) ([OsciSocketServRequestQElem](#) \*)
- [PVLogger \\* Logger](#) ()

## Static Public Methods

- OslSocketI \* [NewL](#) ([Osl\\_DefAlloc](#) &a)
- bool [MakeAddr](#) ([OslNetworkAddress](#) &in, [TOslSockAddr](#) &addr)
- void [MakeAddr](#) ([TOslSockAddr](#) &in, [OslNetworkAddress](#) &addr)
- bool [MakeMulticastGroupInformation](#) ([OslIpMReq](#) &in, [TIpMReq](#) &addr)
- void [MakeMulticastGroupInformation](#) ([TIpMReq](#) &in, [OslIpMReq](#) &addr)

## Friends

- class [OslAcceptRequest](#)
- class [OslConnectRequest](#)
- class [OslRecvRequest](#)
- class [OslRecvFromRequest](#)
- class [OslSendRequest](#)
- class [OslSendToRequest](#)
- class [OslShutdownRequest](#)
- class [OslUDPSocket](#)
- class [OslTCPSocket](#)

## 7.218.1 Detailed Description

Socket implementation class

## 7.218.2 Constructor & Destructor Documentation

### 7.218.2.1 OslSocketI::~~OslSocketI ()

## 7.218.3 Member Function Documentation

### 7.218.3.1 void OslSocketI::Accept ([AcceptParam](#) &, [OslSocketRequestAO](#) &) [virtual]

Implements [OslSocketIBase](#).

### 7.218.3.2 int32 OslSocketI::Bind ([OslNetworkAddress](#) & *anAddr*) [virtual]

Implements [OslSocketIBase](#).

### 7.218.3.3 int32 OslSocketI::Close () [virtual]

Implements [OslSocketIBase](#).

### 7.218.3.4 void OslSocketI::Connect ([ConnectParam](#) &, [OslSocketRequestAO](#) &) [virtual]

Implements [OslSocketIBase](#).

**7.218.3.5** `int32 OsciSocketI::GetPeerName (OsciNetworkAddress & peerName)`

**7.218.3.6** `int32 OsciSocketI::Join (OsciNetworkAddress & anAddr) [virtual]`

Implements [OsciSocketIBase](#).

**7.218.3.7** `int32 OsciSocketI::Listen (uint32 qSize) [virtual]`

Implements [OsciSocketIBase](#).

**7.218.3.8** `PVLogger* OsciSocketI::Logger () [inline]`

**7.218.3.9** `void OsciSocketI::MakeAddr (TOsciSockAddr & in, OsciNetworkAddress & addr) [static]`

**7.218.3.10** `bool OsciSocketI::MakeAddr (OsciNetworkAddress & in, TOsciSockAddr & addr) [static]`

**7.218.3.11** `void OsciSocketI::MakeMulticastGroupInformation (TIpMReq & in, OsciIpMReq & addr) [static]`

**7.218.3.12** `bool OsciSocketI::MakeMulticastGroupInformation (OsciIpMReq & in, TIpMReq & addr) [static]`

**7.218.3.13** `OsciSocketI* OsciSocketI::NewL (Osci_DefAlloc & a) [static]`

**7.218.3.14** `int32 OsciSocketI::Open (OsciSocketServI & aServer) [virtual]`

Implements [OsciSocketIBase](#).

**7.218.3.15** `int32 OsciSocketI::Open (OsciSocketServI & aServer, uint32 addrFamily, uint32 sockType, uint32 protocol) [virtual]`

Implements [OsciSocketIBase](#).

**7.218.3.16** void OsciSocketI::ProcessAccept ([OsciSocketServRequestQElem \\*](#))

**7.218.3.17** void OsciSocketI::ProcessConnect ([OsciSocketServRequestQElem \\*](#))

**7.218.3.18** void OsciSocketI::ProcessRecv ([OsciSocketServRequestQElem \\*](#))

**7.218.3.19** void OsciSocketI::ProcessRecvFrom ([OsciSocketServRequestQElem \\*](#))

**7.218.3.20** void OsciSocketI::ProcessSend ([OsciSocketServRequestQElem \\*](#))

**7.218.3.21** void OsciSocketI::ProcessSendTo ([OsciSocketServRequestQElem \\*](#))

**7.218.3.22** void OsciSocketI::ProcessShutdown ([OsciSocketServRequestQElem \\*](#))

**7.218.3.23** void OsciSocketI::Recv ([RecvParam &](#), [OsciSocketRequestAO &](#)) [virtual]

Implements [OsciSocketIBase](#).

**7.218.3.24** void OsciSocketI::RecvFrom ([RecvFromParam &](#), [OsciSocketRequestAO &](#)) [virtual]

Implements [OsciSocketIBase](#).

**7.218.3.25** void OsciSocketI::RecvFromSuccess ([RecvFromParam &](#)) [virtual]

Implements [OsciSocketIBase](#).

**7.218.3.26** void OsciSocketI::RecvSuccess ([RecvParam &](#)) [virtual]

Implements [OsciSocketIBase](#).

**7.218.3.27** void OsciSocketI::Send ([SendParam &](#), [OsciSocketRequestAO &](#)) [virtual]

Implements [OsciSocketIBase](#).

**7.218.3.28** void OsciSocketI::SendSuccess ([SendParam &](#)) [virtual]

Implements [OsciSocketIBase](#).

**7.218.3.29** void OsciSocketI::SendTo ([SendToParam &](#), [OsciSocketRequestAO &](#)) [virtual]

Implements [OsciSocketIBase](#).

**7.218.3.30** void OsciSocketI::SendToSuccess ([SendToParam &](#)) [virtual]

Implements [OsciSocketIBase](#).



- 7.218.3.31 `int32 OslSocketI::SetRecvBufferSize (uint32 size)`
- 7.218.3.32 `int32 OslSocketI::SetSockOpt (TPVSocketOptionLevel aOptionLevel,  
TPVSocketOptionName aOptionName, OslAny * aOptionValue, int32 aOptionLen)`
- 7.218.3.33 `void OslSocketI::Shutdown (ShutdownParam &, OslSocketRequestAO &)`  
[virtual]

Implements [OslSocketIBase](#).

- 7.218.3.34 `TOslSocket OslSocketI::Socket ()` [inline]
- 7.218.3.35 `TPVSocketEvent OslSocketI::ThreadLogoff ()`
- 7.218.3.36 `TPVSocketEvent OslSocketI::ThreadLogon (OslSocketServI * aServ)`

## 7.218.4 Friends And Related Function Documentation

- 7.218.4.1 `friend class OslAcceptRequest` [friend]
- 7.218.4.2 `friend class OslConnectRequest` [friend]
- 7.218.4.3 `friend class OslRecvFromRequest` [friend]
- 7.218.4.4 `friend class OslRecvRequest` [friend]
- 7.218.4.5 `friend class OslSendRequest` [friend]
- 7.218.4.6 `friend class OslSendToRequest` [friend]
- 7.218.4.7 `friend class OslShutdownRequest` [friend]
- 7.218.4.8 `friend class OslTCPSocket` [friend]

Reimplemented from [OslSocketIBase](#).

- 7.218.4.9 `friend class OslUDPSocket` [friend]

Reimplemented from [OslSocketIBase](#).

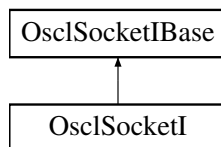
The documentation for this class was generated from the following file:

- [osl\\_socket\\_imp\\_pv.h](#)

## 7.219 OslSocketIBase Class Reference

```
#include <osl_socket_imp_base.h>
```

Inheritance diagram for OslSocketIBase::



### Public Methods

- virtual [~OslSocketIBase](#) ()
- virtual int32 [Open](#) ([OslSocketServI](#) &aServer, uint32 addrFamily, uint32 sockType, uint32 protocol)=0
- virtual int32 [Open](#) ([OslSocketServI](#) &aServer)=0
- virtual int32 [Bind](#) ([OslNetworkAddress](#) &anAddr)=0
- virtual int32 [Join](#) ([OslNetworkAddress](#) &anAddr)=0
- virtual int32 [Close](#) ()=0
- virtual int32 [Listen](#) (uint32 qSize)=0
- virtual void [Connect](#) ([ConnectParam](#) &, [OslSocketRequestAO](#) &)=0
- virtual void [Accept](#) ([AcceptParam](#) &, [OslSocketRequestAO](#) &)=0
- virtual void [Shutdown](#) ([ShutdownParam](#) &, [OslSocketRequestAO](#) &)=0
- virtual void [Send](#) ([SendParam](#) &, [OslSocketRequestAO](#) &)=0
- virtual void [SendSuccess](#) ([SendParam](#) &)=0
- virtual void [SendTo](#) ([SendToParam](#) &, [OslSocketRequestAO](#) &)=0
- virtual void [SendToSuccess](#) ([SendToParam](#) &)=0
- virtual void [Recv](#) ([RecvParam](#) &, [OslSocketRequestAO](#) &)=0
- virtual void [RecvSuccess](#) ([RecvParam](#) &)=0
- virtual void [RecvFrom](#) ([RecvFromParam](#) &, [OslSocketRequestAO](#) &)=0
- virtual void [RecvFromSuccess](#) ([RecvFromParam](#) &)=0
- virtual void [BindAsync](#) ([BindParam](#) &, [OslSocketRequestAO](#) &)
- virtual void [ListenAsync](#) ([ListenParam](#) &, [OslSocketRequestAO](#) &)
- void [CancelFxn](#) ([TPVSocketFxn](#))

### Static Public Methods

- bool [HasAsyncBind](#) ()
- bool [HasAsyncListen](#) ()

### Protected Methods

- [OslSocketIBase](#) ([Osl\\_DefAlloc](#) &a)
- virtual void [CancelConnect](#) ()=0
- virtual void [CancelAccept](#) ()=0
- virtual void [CancelShutdown](#) ()=0
- virtual void [CancelSend](#) ()=0

- virtual void [CancelSendTo](#) ()=0
- virtual void [CancelRecv](#) ()=0
- virtual void [CancelRecvFrom](#) ()=0
- virtual void [CancelBind](#) ()
- virtual void [CancelListen](#) ()
- virtual bool [IsOpen](#) ()=0

## Static Protected Methods

- int [GetShutdown](#) (TPVSocketShutdown aOsciVal)

## Protected Attributes

- [Osci\\_DefAlloc](#) & [iAlloc](#)
- [OsciSocketServI](#) \* [iSocketServ](#)

## Friends

- class [OsciSocketRequest](#)
- class [OsciSocketMethod](#)
- class [OsciSocketRequestAO](#)
- class [OsciUDPSocket](#)
- class [OsciTCPSocket](#)

### 7.219.1 Detailed Description

Socket implementation base class

### 7.219.2 Constructor & Destructor Documentation

**7.219.2.1** virtual [OsciSocketIBase::~~OsciSocketIBase](#) () [virtual]

**7.219.2.2** [OsciSocketIBase::OsciSocketIBase](#) ([Osci\\_DefAlloc](#) & *a*) [protected]

### 7.219.3 Member Function Documentation

**7.219.3.1** virtual void [OsciSocketIBase::Accept](#) ([AcceptParam](#) &, [OsciSocketRequestAO](#) &) [pure virtual]

Implemented in [OsciSocketI](#).

**7.219.3.2** virtual int32 [OsciSocketIBase::Bind](#) ([OsciNetworkAddress](#) & *anAddr*) [pure virtual]

Implemented in [OsciSocketI](#).

- 7.219.3.3 **virtual void OsciSocketIBase::BindAsync ([BindParam](#) &, [OsciSocketRequestAO](#) &)**  
[inline, virtual]
- 7.219.3.4 **virtual void OsciSocketIBase::CancelAccept ()** [protected, pure virtual]
- 7.219.3.5 **virtual void OsciSocketIBase::CancelBind ()** [inline, protected, virtual]
- 7.219.3.6 **virtual void OsciSocketIBase::CancelConnect ()** [protected, pure virtual]
- 7.219.3.7 **void OsciSocketIBase::CancelFxn ([TPVSocketFxn](#))**
- 7.219.3.8 **virtual void OsciSocketIBase::CancelListen ()** [inline, protected, virtual]
- 7.219.3.9 **virtual void OsciSocketIBase::CancelRecv ()** [protected, pure virtual]
- 7.219.3.10 **virtual void OsciSocketIBase::CancelRecvFrom ()** [protected, pure virtual]
- 7.219.3.11 **virtual void OsciSocketIBase::CancelSend ()** [protected, pure virtual]
- 7.219.3.12 **virtual void OsciSocketIBase::CancelSendTo ()** [protected, pure virtual]
- 7.219.3.13 **virtual void OsciSocketIBase::CancelShutdown ()** [protected, pure virtual]
- 7.219.3.14 **virtual int32 OsciSocketIBase::Close ()** [pure virtual]

Implemented in [OsciSocketI](#).

- 7.219.3.15 **virtual void OsciSocketIBase::Connect ([ConnectParam](#) &, [OsciSocketRequestAO](#) &)**  
[pure virtual]

Implemented in [OsciSocketI](#).

- 7.219.3.16 **int OsciSocketIBase::GetShutdown ([TPVSocketShutdown](#) *aOsciVal*)** [static, protected]
- 7.219.3.17 **bool OsciSocketIBase::HasAsyncBind ()** [static]
- 7.219.3.18 **bool OsciSocketIBase::HasAsyncListen ()** [static]
- 7.219.3.19 **virtual bool OsciSocketIBase::IsOpen ()** [protected, pure virtual]
- 7.219.3.20 **virtual int32 OsciSocketIBase::Join ([OsciNetworkAddress](#) & *anAddr*)** [pure virtual]

Implemented in [OsciSocketI](#).

- 7.219.3.21 **virtual int32 OsciSocketIBase::Listen (uint32 *qSize*)** [pure virtual]

Implemented in [OsciSocketI](#).

**7.219.3.22** `virtual void OsciSocketIBase::ListenAsync (ListenParam &, OsciSocketRequestAO &) [inline, virtual]`

**7.219.3.23** `virtual int32 OsciSocketIBase::Open (OsciSocketServI & aServer) [pure virtual]`

Implemented in [OsciSocketI](#).

**7.219.3.24** `virtual int32 OsciSocketIBase::Open (OsciSocketServI & aServer, uint32 addrFamily, uint32 sockType, uint32 protocol) [pure virtual]`

Implemented in [OsciSocketI](#).

**7.219.3.25** `virtual void OsciSocketIBase::Recv (RecvParam &, OsciSocketRequestAO &) [pure virtual]`

Implemented in [OsciSocketI](#).

**7.219.3.26** `virtual void OsciSocketIBase::RecvFrom (RecvFromParam &, OsciSocketRequestAO &) [pure virtual]`

Implemented in [OsciSocketI](#).

**7.219.3.27** `virtual void OsciSocketIBase::RecvFromSuccess (RecvFromParam &) [pure virtual]`

Implemented in [OsciSocketI](#).

**7.219.3.28** `virtual void OsciSocketIBase::RecvSuccess (RecvParam &) [pure virtual]`

Implemented in [OsciSocketI](#).

**7.219.3.29** `virtual void OsciSocketIBase::Send (SendParam &, OsciSocketRequestAO &) [pure virtual]`

Implemented in [OsciSocketI](#).

**7.219.3.30** `virtual void OsciSocketIBase::SendSuccess (SendParam &) [pure virtual]`

Implemented in [OsciSocketI](#).

**7.219.3.31** `virtual void OsciSocketIBase::SendTo (SendToParam &, OsciSocketRequestAO &) [pure virtual]`

Implemented in [OsciSocketI](#).

**7.219.3.32** virtual void OslSocketIBase::SendToSuccess ([SendToParam](#) &) [pure virtual]

Implemented in [OslSocketI](#).

**7.219.3.33** virtual void OslSocketIBase::Shutdown ([ShutdownParam](#) &, [OslSocketRequestAO](#) &) [pure virtual]

Implemented in [OslSocketI](#).

## 7.219.4 Friends And Related Function Documentation

**7.219.4.1** friend class OslSocketMethod [friend]

**7.219.4.2** friend class OslSocketRequest [friend]

**7.219.4.3** friend class OslSocketRequestAO [friend]

**7.219.4.4** friend class OslTCPSocket [friend]

Reimplemented in [OslSocketI](#).

**7.219.4.5** friend class OslUDPSocket [friend]

Reimplemented in [OslSocketI](#).

## 7.219.5 Field Documentation

**7.219.5.1** [Osl\\_DefAlloc](#)& OslSocketIBase::iAlloc [protected]

**7.219.5.2** [OslSocketServI](#)\* OslSocketIBase::iSocketServ [protected]

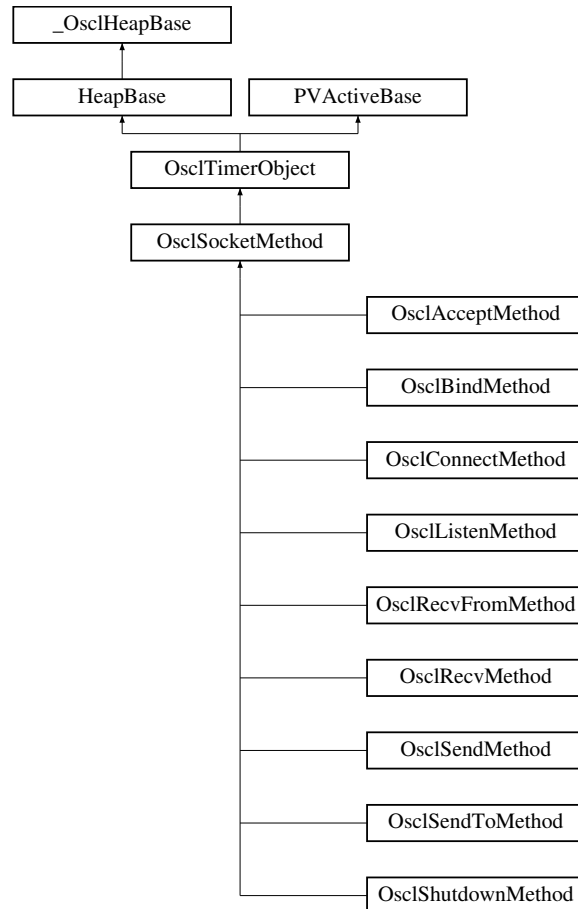
The documentation for this class was generated from the following file:

- [osl\\_socket\\_imp\\_base.h](#)

## 7.220 OsciSocketMethod Class Reference

```
#include <osci_socket_method.h>
```

Inheritance diagram for OsciSocketMethod::



### Public Methods

- [OsciSocketMethod](#) ([OsciIPSocketI](#) &aContainer, const char \*name, [TPVSocketFxn](#) fxn)
- virtual [~OsciSocketMethod](#) ()
- void [Abort](#) ()
- void [AbortAll](#) ()
- void [CancelMethod](#) ()
- [Osci\\_DefAlloc](#) & [Alloc](#) ()
- [TPVSocketEvent](#) [ThreadLogon](#) ()
- [TPVSocketEvent](#) [ThreadLogoff](#) ()

### Data Fields

- [OsciIPSocketI](#) & [iContainer](#)
- [TPVSocketFxn](#) [iSocketFxn](#)

## Protected Methods

- void [ConstructL](#) ([OsciSocketRequestAO](#) \*aAO)
- bool [StartMethod](#) (int32 aTimeoutMsec)
- void [MethodDone](#) ()
- void [Run](#) ()

## Protected Attributes

- [OsciSocketRequestAO](#) \* [iSocketRequestAO](#)

### 7.220.1 Detailed Description

OsciSocketMethod 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 OsciSocketMethod class implements the timeout and contains the request completion AO.

### 7.220.2 Constructor & Destructor Documentation

**7.220.2.1** [OsciSocketMethod::OsciSocketMethod](#) ([OsciIPSocketI](#) & *aContainer*, const char \* *name*, [TPVSocketFxn](#) *fxn*) [inline]

**7.220.2.2** [virtual OsciSocketMethod::~~OsciSocketMethod](#) () [inline, virtual]

### 7.220.3 Member Function Documentation

**7.220.3.1** [void OsciSocketMethod::Abort](#) () [inline]

**7.220.3.2** [void OsciSocketMethod::AbortAll](#) () [inline]

**7.220.3.3** [Osci\\_DefAlloc](#) & [OsciSocketMethod::Alloc](#) () [inline]

**7.220.3.4** [void OsciSocketMethod::CancelMethod](#) () [inline]

**7.220.3.5** [void OsciSocketMethod::ConstructL](#) ([OsciSocketRequestAO](#) \* *aAO*) [inline, protected]

**7.220.3.6** [void OsciSocketMethod::MethodDone](#) () [inline, protected]

**7.220.3.7** [void OsciSocketMethod::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 OslSocketMethod::StartMethod (int32 aTimeoutMsec)` [protected]

**7.220.3.9** `TPVSocketEvent OslSocketMethod::ThreadLogoff ()`

**7.220.3.10** `TPVSocketEvent OslSocketMethod::ThreadLogon ()`

## 7.220.4 Field Documentation

**7.220.4.1** `OslIPSocketI& OslSocketMethod::iContainer`

**7.220.4.2** `TPVSocketFxn OslSocketMethod::iSocketFxn`

**7.220.4.3** `OslSocketRequestAO* OslSocketMethod::iSocketRequestAO` [protected]

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_method.h](#)

## 7.221 OslSocketObserver Class Reference

```
#include <osl_socket_types.h>
```

### Public Methods

- virtual OSCL\_IMPORT\_REF void [HandleSocketEvent](#) (int32 aId, [TPVSocketFxn](#) aFxn, [TPVSocketEvent](#) aEvent, int32 aError)=0
- virtual [~OslSocketObserver](#) ()

### 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 OslSocketObserver::~~OslSocketObserver () [inline, virtual]

### 7.221.3 Member Function Documentation

7.221.3.1 virtual OSCL\_IMPORT\_REF void OslSocketObserver::HandleSocketEvent (int32 aId, [TPVSocketFxn](#) aFxn, [TPVSocketEvent](#) aEvent, int32 aError) [pure virtual]

Socket Event callback.

#### Parameters:

**aId:** The ID that was supplied when the socket was created.

**aFxn:** Type of socket function call.

**aEvent:** Function completion event. Will be EPVSocketSuccess, EPVSocketTimeout, or EPVSocket-Failure.

**aError:** When the event is EPVSocketFailure, this may contain a platform-specific error code, or zero if none is available.

The documentation for this class was generated from the following file:

- [osl\\_socket\\_types.h](#)

## 7.222 OsciSocketRequest Class Reference

```
#include <osci_socket_request.h>
```

### Public Methods

- [OsciSocketRequest](#) ()
- [TPVSocketFxn](#) Fxn ()
- void [CancelRequest](#) ()
- void [Activate](#) ([SocketRequestParam](#) \*iParam, [OsciSocketRequestAO](#) &a)
- void [Complete](#) ([OsciSocketServRequestQElem](#) \*, int32 aStatus, int32 aSockErr=0)

### Data Fields

- [OsciSocketRequestAO](#) \* iSocketRequestAO
- [SocketRequestParam](#) \* iParam
- [OsciSocketI](#) \* 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 [OsciSocketRequest::OsciSocketRequest](#) () [inline]

### 7.222.3 Member Function Documentation

7.222.3.1 void [OsciSocketRequest::Activate](#) ([SocketRequestParam](#) \* iParam, [OsciSocketRequestAO](#) &a)

7.222.3.2 void [OsciSocketRequest::CancelRequest](#) ()

7.222.3.3 void [OsciSocketRequest::Complete](#) ([OsciSocketServRequestQElem](#) \*, int32 aStatus, int32 aSockErr = 0)

7.222.3.4 [TPVSocketFxn](#) [OsciSocketRequest::Fxn](#) () [inline]

### 7.222.4 Field Documentation

7.222.4.1 [SocketRequestParam](#)\* [OsciSocketRequest::iParam](#)

7.222.4.2 [OsciSocketI](#)\* [OsciSocketRequest::iSocketI](#)

7.222.4.3 [OsciSocketRequestAO](#)\* [OsciSocketRequest::iSocketRequestAO](#)

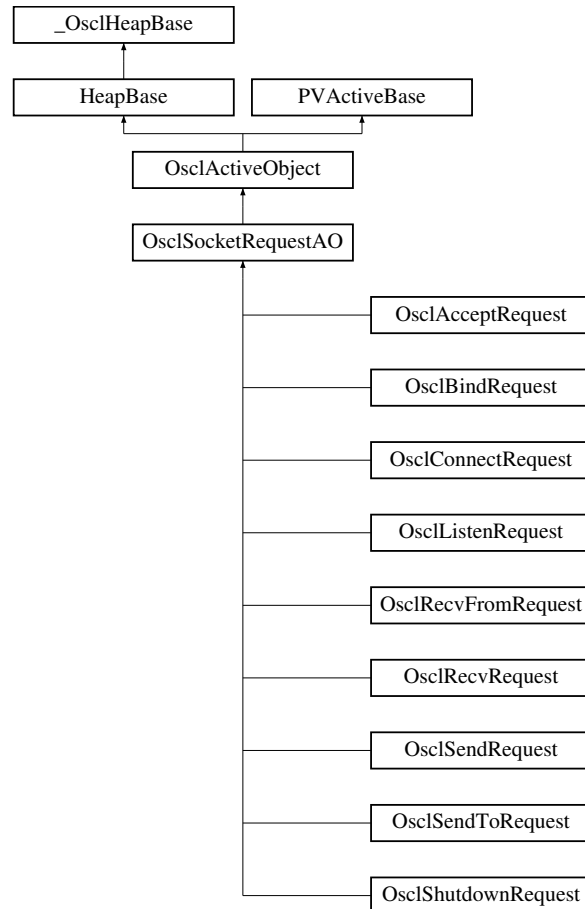
The documentation for this class was generated from the following file:

- [osci\\_socket\\_request.h](#)

## 7.223 OsciSocketRequestAO Class Reference

```
#include <osci_socket_method.h>
```

Inheritance diagram for OsciSocketRequestAO::



### Public Methods

- void [ConstructL](#) ()

### Protected Methods

- [OsciSocketRequestAO](#) ([OsciSocketMethod](#) &aContainer, const char \*name)
- virtual [~OsciSocketRequestAO](#) ()
- [OsciAny](#) \* [NewRequest](#) (const uint32 size)
- void [CleanupParam](#) (bool deallocate=false)
- void [Abort](#) ()
- void [RequestDone](#) ()
- int [GetSocketError](#) ()
- void [DoCancel](#) ()
- void [Run](#) ()

- virtual void [Success](#) ()
- [OsciSocketI](#) \* [SocketI](#) ()
- [OsciSocketObserver](#) \* [SocketObserver](#) ()
- uint32 [Id](#) ()
- [Osci\\_DefAlloc](#) & [Alloc](#) ()

## Protected Attributes

- [OsciSocketMethod](#) & [iContainer](#)
- int32 [iSocketError](#)
- [SocketRequestParam](#) \* [iParam](#)
- uint32 [iParamSize](#)

## Friends

- class [OsciSocketI](#)
- class [OsciSocketMethod](#)
- class [OsciSocketRequest](#)

### 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 [OsciSocketMethod](#) object

### 7.223.2 Constructor & Destructor Documentation

**7.223.2.1** [OsciSocketRequestAO::OsciSocketRequestAO](#) ([OsciSocketMethod](#) & *aContainer*, const char \* *name*) [inline, protected]

**7.223.2.2** virtual [OsciSocketRequestAO::~~OsciSocketRequestAO](#) () [inline, protected, virtual]

### 7.223.3 Member Function Documentation

**7.223.3.1** void [OsciSocketRequestAO::Abort](#) () [inline, protected]

**7.223.3.2** [Osci\\_DefAlloc](#) & [OsciSocketRequestAO::Alloc](#) () [inline, protected]

**7.223.3.3** void [OsciSocketRequestAO::CleanupParam](#) (bool *deallocate* = false) [protected]

**7.223.3.4** void [OsciSocketRequestAO::ConstructL](#) () [inline]

**7.223.3.5** void [OsciSocketRequestAO::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 [OsciActiveObject](#).

**7.223.3.6** `int OsciSocketRequestAO::GetSocketError ()` [inline, protected]

**7.223.3.7** `uint32 OsciSocketRequestAO::Id ()` [inline, protected]

**7.223.3.8** `OsciAny* OsciSocketRequestAO::NewRequest (const uint32 size)` [protected]

**7.223.3.9** `void OsciSocketRequestAO::RequestDone ()` [inline, protected]

**7.223.3.10** `void OsciSocketRequestAO::Run ()` [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implements `PVActiveBase`.

**7.223.3.11** `OsciSocketI* OsciSocketRequestAO::SocketI ()` [inline, protected]

**7.223.3.12** `OsciSocketObserver* OsciSocketRequestAO::SocketObserver ()` [inline, protected]

**7.223.3.13** `virtual void OsciSocketRequestAO::Success ()` [inline, protected, virtual]

Reimplemented in `OsciRecvRequest`, `OsciRecvFromRequest`, `OsciSendRequest`, and `OsciSendToRequest`.

## 7.223.4 Friends And Related Function Documentation

7.223.4.1 friend class OslSocketI [friend]

7.223.4.2 friend class OslSocketMethod [friend]

7.223.4.3 friend class OslSocketRequest [friend]

## 7.223.5 Field Documentation

7.223.5.1 [OslSocketMethod](#)& OslSocketRequestAO::iContainer [protected]

7.223.5.2 [SocketRequestParam](#)\* OslSocketRequestAO::iParam [protected]

7.223.5.3 uint32 OslSocketRequestAO::iParamSize [protected]

7.223.5.4 int32 OslSocketRequestAO::iSocketError [protected]

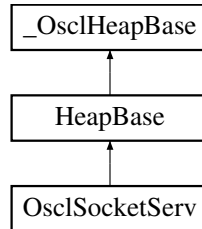
The documentation for this class was generated from the following file:

- [osl\\_socket\\_method.h](#)

## 7.224 OslSocketServ Class Reference

```
#include <osl_socket.h>
```

Inheritance diagram for OslSocketServ::



### Public Methods

- OSCL\_IMPORT\_REF [~OslSocketServ](#) ()
- OSCL\_IMPORT\_REF int32 [Connect](#) (uint32 aMessageSlots=8, bool aShareSession=false)
- OSCL\_IMPORT\_REF void [Close](#) (bool aCleanup=true)

### Static Public Methods

- OSCL\_IMPORT\_REF OslSocketServ \* [NewL](#) ([Osl\\_DefAlloc](#) &alloc)

### Friends

- class [OslTCPSocket](#)
- class [OslUDPSocket](#)
- class [OslDNS](#)

### 7.224.1 Constructor & Destructor Documentation

#### 7.224.1.1 OSCL\_IMPORT\_REF OslSocketServ::~~OslSocketServ ()

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 OslSocketServ::Close (bool aCleanup = true)

Close socket server. This is a synchronous method.

#### Parameters:

*aCleanup*: cleanup the socket system? the socket cleanup should only be done when all parts of the application are done using sockets.



### 7.224.2.2 OSCL\_IMPORT\_REF int32 OsciSocketServ::Connect (uint32 *aMessageSlots* = 8, bool *aShareSession* = false)

Connect to socket server. This is a synchronous method.

**Parameters:**

*Number* of message slots.

**Returns:**

Returns OsciErrNone for success, or a platform-specific code.

### 7.224.2.3 OSCL\_IMPORT\_REF OsciSocketServ\* OsciSocketServ::NewL (Osci\_DefAlloc & *alloc*) [static]

Create a socket server. May leave if failure.

**Parameters:**

*alloc*: Memory allocator.

**Returns:**

Returns pointer to socket server

## 7.224.3 Friends And Related Function Documentation

### 7.224.3.1 friend class OsciDNS [friend]

### 7.224.3.2 friend class OsciTCPSocket [friend]

### 7.224.3.3 friend class OsciUDPSocket [friend]

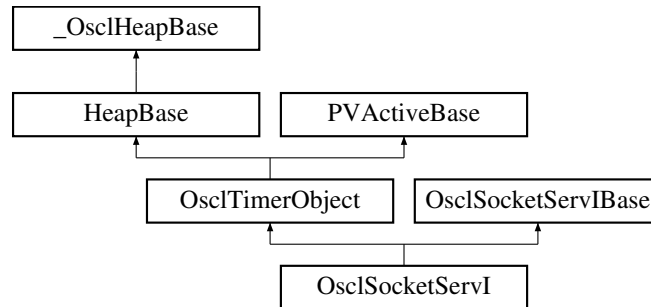
The documentation for this class was generated from the following file:

- [osci\\_socket.h](#)

## 7.225 OsciSocketServI Class Reference

```
#include <osci_socket_serv_imp_pv.h>
```

Inheritance diagram for OsciSocketServI::



### Public Methods

- int32 [Connect](#) (uint32 aMessageSlots, bool aShareSession)
- void [Close](#) (bool)
- bool [IsServerThread](#) ()

### Static Public Methods

- OsciSocketServI \* [NewL](#) (Osci\_DefAlloc &a)

### Friends

- class [OsciSocketServRequestList](#)
- class [LoopbackSocket](#)
- class [OsciTCPSocketI](#)
- class [OsciUDPSocketI](#)
- class [OsciSocketI](#)
- class [OsciDNSI](#)
- class [OsciSocketRequest](#)
- class [OsciSocketServ](#)

### 7.225.1 Detailed Description

PV socket server implementation

### 7.225.2 Member Function Documentation

#### 7.225.2.1 void OsciSocketServI::Close (bool) [virtual]

Implements [OsciSocketServIBase](#).

**7.225.2.2** `int32 OsciSocketServI::Connect (uint32 aMessageSlots, bool aShareSession)`  
[virtual]

Implements [OsciSocketServIBase](#).

**7.225.2.3** `bool OsciSocketServI::IsServerThread ()`

**7.225.2.4** `OsciSocketServI* OsciSocketServI::NewL (Osci\_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 OsciDNSI` [friend]

**7.225.3.3** `friend class OsciSocketI` [friend]

**7.225.3.4** `friend class OsciSocketRequest` [friend]

**7.225.3.5** `friend class OsciSocketServ` [friend]

**7.225.3.6** `friend class OsciSocketServRequestList` [friend]

**7.225.3.7** `friend class OsciTCPSocketI` [friend]

**7.225.3.8** `friend class OsciUDPSocketI` [friend]

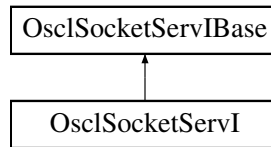
The documentation for this class was generated from the following file:

- [osci\\_socket\\_serv\\_imp\\_pv.h](#)

## 7.226 OsciSocketServIBase Class Reference

```
#include <osci_socket_serv_imp_base.h>
```

Inheritance diagram for OsciSocketServIBase::



### Public Methods

- virtual [~OsciSocketServIBase](#) ()
- 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

- [OsciSocketServIBase](#) ([Osci\\_DefAlloc](#) &a)
- [TSocketServState](#) [State](#) () const
- bool [IsServConnected](#) () const

### Protected Attributes

- [Osci\\_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 OsciSocketServIBase::TSocketServState [protected]

Enumeration values:

[ESocketServ\\_Idle](#)

ESocketServ\_Connected

ESocketServ\_Error

### 7.226.3 Constructor & Destructor Documentation

**7.226.3.1** virtual OslSocketServIBase::~OslSocketServIBase () [inline, virtual]

**7.226.3.2** OslSocketServIBase::OslSocketServIBase ([Osl\\_DefAlloc](#) & *a*) [inline, protected]

### 7.226.4 Member Function Documentation

**7.226.4.1** virtual void OslSocketServIBase::Close (bool) [pure virtual]

Implemented in [OslSocketServI](#).

**7.226.4.2** virtual int32 OslSocketServIBase::Connect (uint32 *aMessageSlots*, bool *aShareSession*) [pure virtual]

Implemented in [OslSocketServI](#).

**7.226.4.3** bool OslSocketServIBase::IsServConnected () const [inline, protected]

**7.226.4.4** [TSocketServState](#) OslSocketServIBase::State () const [inline, protected]

### 7.226.5 Field Documentation

**7.226.5.1** [Osl\\_DefAlloc](#)& OslSocketServIBase::iAlloc [protected]

**7.226.5.2** [PVLogger\\*](#) OslSocketServIBase::iLogger

**7.226.5.3** int OslSocketServIBase::iServerError [protected]

**7.226.5.4** [TSocketServState](#) OslSocketServIBase::iServState [protected]

The documentation for this class was generated from the following file:

- [osl\\_socket\\_serv\\_imp\\_base.h](#)

## 7.227 OsciSocketServRequestList Class Reference

```
#include <osci_socket_serv_imp_reqlist.h>
```

### Public Methods

- [OsciSocketServRequestList](#) ()
- void [Add](#) ([OsciSocketRequest](#) \*)
- void [StartCancel](#) ([OsciSocketRequest](#) \*)
- void [Open](#) ([OsciSocketServI](#) \*s)
- void [Close](#) ()
- void [Wakeup](#) ()
- void [WaitOnRequests](#) ()
- void [Remove](#) ([OsciSocketServRequestQElem](#) \*aElem)

### Friends

- class [OsciSocketServI](#)

### 7.227.1 Detailed Description

PV socket server request queue

### 7.227.2 Constructor & Destructor Documentation

**7.227.2.1** [OsciSocketServRequestList::OsciSocketServRequestList](#) ()

### 7.227.3 Member Function Documentation

**7.227.3.1** void [OsciSocketServRequestList::Add](#) ([OsciSocketRequest](#) \*)

**7.227.3.2** void [OsciSocketServRequestList::Close](#) ()

**7.227.3.3** void [OsciSocketServRequestList::Open](#) ([OsciSocketServI](#) \*s)

**7.227.3.4** void [OsciSocketServRequestList::Remove](#) ([OsciSocketServRequestQElem](#) \* *aElem*)  
[inline]

**7.227.3.5** void [OsciSocketServRequestList::StartCancel](#) ([OsciSocketRequest](#) \*)

**7.227.3.6** void [OsciSocketServRequestList::WaitOnRequests](#) ()

**7.227.3.7** void [OsciSocketServRequestList::Wakeup](#) ()

### 7.227.4 Friends And Related Function Documentation

**7.227.4.1** friend class [OsciSocketServI](#) [friend]

The documentation for this class was generated from the following file:

- [osci\\_socket\\_serv\\_imp\\_reqlist.h](#)

## 7.228 OsciSocketServRequestQElem Class Reference

```
#include <osci_socket_serv_imp_reqlist.h>
```

### Public Methods

- [OsciSocketServRequestQElem](#) ([OsciSocketRequest](#) \*r)

### Data Fields

- [OsciSocketRequest](#) \* iSocketRequest
- uint8 iSelect
- bool iCancel

### 7.228.1 Constructor & Destructor Documentation

**7.228.1.1** [OsciSocketServRequestQElem::OsciSocketServRequestQElem](#) ([OsciSocketRequest](#) \* r)  
[inline]

### 7.228.2 Field Documentation

**7.228.2.1** bool [OsciSocketServRequestQElem::iCancel](#)

**7.228.2.2** uint8 [OsciSocketServRequestQElem::iSelect](#)

**7.228.2.3** [OsciSocketRequest](#)\* [OsciSocketServRequestQElem::iSocketRequest](#)

The documentation for this class was generated from the following file:

- [osci\\_socket\\_serv\\_imp\\_reqlist.h](#)



## 7.229 OsciSocketTOS Class Reference

```
#include <osci_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

- [OsciSocketTOS](#) ()
- 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 OsciSocketTOS::TPVServicePrecedence

Enumeration values:

[EPVRoutine](#)  
[EPVPriority](#)  
[EPVImmediate](#)  
[EPVFlash](#)  
[EPVOverrideFlash](#)  
[EPVCritic\\_Ecp](#)  
[EPVInetControl](#)  
[EPVNetControl](#)

#### 7.229.1.2 enum OsciSocketTOS::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]

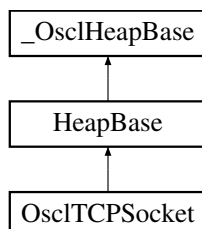
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_types.h](#)

## 7.230 OsciTCPSocket Class Reference

```
#include <osci_socket.h>
```

Inheritance diagram for OsciTCPSocket::



### Public Methods

- OSCL\_IMPORT\_REF [~OsciTCPSocket](#) ()
- OSCL\_IMPORT\_REF [TPVSocketEvent ThreadLogoff](#) ()
- OSCL\_IMPORT\_REF [TPVSocketEvent ThreadLogon](#) ([OsciSocketServ](#) &aServ, [OsciSocketObserver](#) \*aObserver)
- OSCL\_IMPORT\_REF int32 [Close](#) ()
- OSCL\_IMPORT\_REF int32 [Bind](#) ([OsciNetworkAddress](#) &aAddress)
- OSCL\_IMPORT\_REF [TPVSocketEvent BindAsync](#) ([OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void [CancelBind](#) ()
- OSCL\_IMPORT\_REF int32 [SetOptionToReuseAddress](#) ()
- OSCL\_IMPORT\_REF int32 [SetTOS](#) (const [OsciSocketTOS](#) &aTOS)
- OSCL\_IMPORT\_REF int32 [GetPeerName](#) ([OsciNetworkAddress](#) &aPeerName)
- OSCL\_IMPORT\_REF int32 [Listen](#) (int32 aQueueSize)
- OSCL\_IMPORT\_REF [TPVSocketEvent ListenAsync](#) (int32 aQueueSize, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void [CancelListen](#) ()
- OSCL\_IMPORT\_REF [OsciTCPSocket](#) \* [GetAcceptedSocketL](#) (uint32 aId)
- OSCL\_IMPORT\_REF uint8 \* [GetRecvData](#) (int32 \*aLength)
- OSCL\_IMPORT\_REF uint8 \* [GetSendData](#) (int32 \*aLength)
- OSCL\_IMPORT\_REF [TPVSocketEvent Connect](#) ([OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void [CancelConnect](#) ()
- OSCL\_IMPORT\_REF [TPVSocketEvent Shutdown](#) ([TPVSocketShutdown](#) aHow, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void [CancelShutdown](#) ()
- OSCL\_IMPORT\_REF [TPVSocketEvent Accept](#) (int32 aTimeout=-1)
- OSCL\_IMPORT\_REF void [CancelAccept](#) ()
- OSCL\_IMPORT\_REF [TPVSocketEvent Send](#) (const uint8 \*aPtr, uint32 aLen, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void [CancelSend](#) ()
- OSCL\_IMPORT\_REF [TPVSocketEvent Recv](#) (uint8 \*aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void [CancelRecv](#) ()

## Static Public Methods

- OSCL\_IMPORT\_REF OsciTCPSocket \* [NewL](#) ([OsciDefAlloc](#) &alloc, [OsciSocketServ](#) &aServ, [OsciSocketObserver](#) \*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 OsciTCPSocket::~~OsciTCPSocket ()

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](#) OsciTCPSocket::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 OsciTCPSocket::Bind ([OsciNetworkAddress](#) &*aAddress*)

Bind a TCP socket to an address. This is a synchronous method.

##### Parameters:

*aAddress*: Bind address.

##### Returns:

Returns OsciErrNone for success, or a platform-specific error code.

#### 7.230.3.3 OSCL\_IMPORT\_REF [TPVSocketEvent](#) OsciTCPSocket::BindAsync ([OsciNetworkAddress](#) &*aAddress*, int32 *aTimeoutMsec* = (-1))

Bind a TCP socket to an address. This is an asynchronous method.

##### Parameters:

*aAddress*: Bind address.

*aTimeoutMsec*: Optional timeout. Use a negative value for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**7.230.3.4 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 OsciTCPSocket::CancelShutdown ()

Cancel Shutdown

This method will cancel any pending Shutdown operation on the current socket, causing the Shutdown to complete with error EPVSocketCancel. If there is no pending Shutdown operation, this method will have no effect.

### 7.230.3.11 OSCL\_IMPORT\_REF int32 OsciTCPSocket::Close ()

Close a TCP socket. This is a synchronous method.

Once it is closed a socket cannot be re-opened. Sockets are automatically closed when they are deleted. This method may be used to see any error code returned from the platform's socket close call.

#### Returns:

Returns OsciErrNone for success, or a platform-specific error code.

### 7.230.3.12 OSCL\_IMPORT\_REF TPVSocketEvent OsciTCPSocket::Connect (OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1)

Connect to an address. This is an asynchronous method.

#### Parameters:

*aAddress*: a network address.

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

#### Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

### 7.230.3.13 OSCL\_IMPORT\_REF OsciTCPSocket\* OsciTCPSocket::GetAcceptedSocketL (uint32 aId)

Retrieve the accept socket after a successful Accept operation. This is a synchronous method.

#### Parameters:

*aId*: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

#### Returns:

Returns pointer to socket, or NULL if error. Note: The caller is responsible for deleting any accepted socket that it retrieves.

### 7.230.3.14 OSCL\_IMPORT\_REF int32 OsciTCPSocket::GetPeerName (OsciNetworkAddress & aPeerName)

Retrieves the peer address of the socket

**Parameters:**

***aPeerName:*** This will store the peer address when API returns successfully.

**Returns:**

Returns OsciErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

**7.230.3.15 OSCL\_IMPORT\_REF uint8\* OsciTCPSocket::GetRecvData (int32 \* *aLength*)**

Retrieve the received data after a successful Recv operation. This is a synchronous method.

**Parameters:**

***aLength:*** (output) number of bytes of data received.

**Returns:**

Returns pointer to received data, or NULL if none.

**7.230.3.16 OSCL\_IMPORT\_REF uint8\* OsciTCPSocket::GetSendData (int32 \* *aLength*)**

Retrieve the sent data after a successful Send operation. This is a synchronous method.

**Parameters:**

***aLength:*** (output) number of bytes of data sent.

**Returns:**

Returns pointer to sent data, or NULL if none.

**7.230.3.17 OSCL\_IMPORT\_REF int32 OsciTCPSocket::Listen (int32 *aQueueSize*)**

Listen. This is a synchronous method.

**Parameters:**

***aQueueSize:*** Queue size.

**Returns:**

Returns OsciErrNone for success, or a platform-specific error code.

**7.230.3.18 OSCL\_IMPORT\_REF TPVSocketEvent OsciTCPSocket::ListenAsync (int32 *aQueueSize*, int32 *aTimeoutMsec* = (-1))**

ListenAsync This is an asynchronous method.

**Parameters:**

***aQueueSize:*** Queue size.

***aTimeoutMsec:*** Optional timeout. Use a negative value for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**7.230.3.19** `OSCL_IMPORT_REF OslTCPSocket* OslTCPSocket::NewL (Osl_DefAlloc & alloc, OslSocketServ & aServ, OslSocketObserver * aObserver, uint32 aId)`  
[static]

Create a TCP Socket. May leave if failure.

**Parameters:**

*alloc*: Memory allocator.

*aServ*: Socket server. Must be connected.

*aObserver*: Socket observer.

*aId*: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

**Returns:**

Returns pointer to socket.

**7.230.3.20** `OSCL_IMPORT_REF TPVSocketEvent OslTCPSocket::Recv (uint8 * aPtr, uint32 aMaxLen, int32 aTimeoutMsec = -1)`

Receive Data. This is an asynchronous method.

**Parameters:**

*aPtr*: Buffer for received data.

*aMaxLen*: Length of buffer.

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

**7.230.3.21** `OSCL_IMPORT_REF TPVSocketEvent OslTCPSocket::Send (const uint8 * aPtr, uint32 aLen, int32 aTimeoutMsec = -1)`

Send Data. This is an asynchronous method.

**Parameters:**

*aPtr*: Data to send.

*aLen*: Length of data to send.

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.



### 7.230.3.22 OSCL\_IMPORT\_REF int32 OsciTCPSocket::SetOptionToReuseAddress ()

Allows the server to bind to an address which is in a TIME\_WAIT state.

#### Returns:

Returns: OsciErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK\_ERR\_BAD\_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

### 7.230.3.23 OSCL\_IMPORT\_REF int32 OsciTCPSocket::SetTOS (const OsciSocketTOS & aTOS)

Sets the Type of Service field of each outgoing IP datagram.

#### Parameters:

*aTOS*: Specifies the type of service requested.

#### Returns:

Returns: OsciErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK\_ERR\_BAD\_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

### 7.230.3.24 OSCL\_IMPORT\_REF TPVSocketEvent OsciTCPSocket::Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec = -1)

Shutdown a socket. This is an asynchronous method.

#### Parameters:

*aHow*: type of shutdown

*aTimeoutMsec*: Timeout in milliseconds, or (-1) for infinite wait.

#### Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

### 7.230.3.25 OSCL\_IMPORT\_REF TPVSocketEvent OsciTCPSocket::ThreadLogoff ()

Thread logoff routine. This will prepare for transfer and use of the socket in another thread. All socket requests must be complete prior to calling this routine. If any requests are still active, it will return EPVSocketFailure;

### 7.230.3.26 OSCL\_IMPORT\_REF TPVSocketEvent OsciTCPSocket::ThreadLogon (OsciSocketServ & aServ, OsciSocketObserver \* aObserver)

Thread logon routine. This will complete the transfer of a socket from another thread for use in the current thread. The ThreadLogoff API must be called in the original thread prior to calling ThreadLogon.

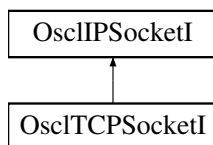
The documentation for this class was generated from the following file:

- [oscl\\_socket.h](#)

## 7.231 OsciTCPsSocketI Class Reference

```
#include <osci_tcp_socket.h>
```

Inheritance diagram for OsciTCPsSocketI::



### Public Methods

- virtual [~OsciTCPsSocketI](#) ()
- [TPVSocketEvent ThreadLogoff](#) ()
- [TPVSocketEvent ThreadLogon](#) ([OsciSocketServI](#) \*aServ, [OsciSocketObserver](#) \*aObserver)
- int32 [Close](#) ()
- int32 [Listen](#) (int aQueueSize)
- [OsciTCPsSocketI](#) \* [GetAcceptedSocketL](#) (uint32 aId)
- uint8 \* [GetRecvData](#) (int32 \*aLength)
- uint8 \* [GetSendData](#) (int32 \*aLength)
- [TPVSocketEvent BindAsync](#) ([OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=-1)
- void [CancelBind](#) ()
- [TPVSocketEvent ListenAsync](#) (uint32 qsize, int32 aTimeoutMsec=-1)
- void [CancelListen](#) ()
- [TPVSocketEvent Connect](#) ([OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=-1)
- void [CancelConnect](#) ()
- [TPVSocketEvent Shutdown](#) ([TPVSocketShutdown](#) aHow, int32 aTimeoutMsec=-1)
- void [CancelShutdown](#) ()
- [TPVSocketEvent Accept](#) (int32 aTimeout=-1)
- void [CancelAccept](#) ()
- [TPVSocketEvent Send](#) (const uint8 \*&aPtr, uint32 aLen, int32 aTimeoutMsec=-1)
- void [CancelSend](#) ()
- [TPVSocketEvent Recv](#) (uint8 \*&aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1)
- void [CancelRecv](#) ()

### Static Public Methods

- [OsciTCPsSocketI](#) \* [NewL](#) ([Osci\\_DefAlloc](#) &a, [OsciSocketServI](#) \*aServ, [OsciSocketObserver](#) \*a-Observer, uint32 aId)

### 7.231.1 Detailed Description

Internal implementation class for [OsciTCPsSocket](#)

## 7.231.2 Constructor & Destructor Documentation

7.231.2.1 `virtual OsciTCPsSocketI::~OsciTCPsSocketI ()` [virtual]

## 7.231.3 Member Function Documentation

7.231.3.1 [TPVSocketEvent](#) `OsciTCPsSocketI::Accept (int32 aTimeout = -1)` [inline]

7.231.3.2 [TPVSocketEvent](#) `OsciTCPsSocketI::BindAsync (OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1)` [inline]

7.231.3.3 `void OsciTCPsSocketI::CancelAccept ()` [inline]

7.231.3.4 `void OsciTCPsSocketI::CancelBind ()` [inline]

7.231.3.5 `void OsciTCPsSocketI::CancelConnect ()` [inline]

7.231.3.6 `void OsciTCPsSocketI::CancelListen ()` [inline]

7.231.3.7 `void OsciTCPsSocketI::CancelRecv ()` [inline]

7.231.3.8 `void OsciTCPsSocketI::CancelSend ()` [inline]

7.231.3.9 `void OsciTCPsSocketI::CancelShutdown ()` [inline]

7.231.3.10 `int32 OsciTCPsSocketI::Close ()` [virtual]

Implements [OsciIPSocketI](#).

7.231.3.11 [TPVSocketEvent](#) `OsciTCPsSocketI::Connect (OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1)` [inline]

7.231.3.12 `OsciTCPsSocketI* OsciTCPsSocketI::GetAcceptedSocketL (uint32 aId)`

7.231.3.13 `uint8 * OsciTCPsSocketI::GetRecvData (int32 * aLength)` [inline, virtual]

Implements [OsciIPSocketI](#).

7.231.3.14 `uint8 * OsciTCPsSocketI::GetSendData (int32 * aLength)` [inline, virtual]

Implements [OsciIPSocketI](#).

- 7.231.3.15 **int32** OslTCPSocketI::Listen (int *aQueueSize*) [inline]
- 7.231.3.16 **TPVSocketEvent** OslTCPSocketI::ListenAsync (uint32 *qsize*, int32 *aTimeoutMsec* = -1) [inline]
- 7.231.3.17 **OslTCPSocketI\*** OslTCPSocketI::NewL (**Osl\_DefAlloc** & *a*, **OslSocketServI** \* *aServ*, **OslSocketObserver** \* *aObserver*, uint32 *aId*) [static]
- 7.231.3.18 **TPVSocketEvent** OslTCPSocketI::Recv (uint8 \*& *aPtr*, uint32 *aMaxLen*, int32 *aTimeoutMsec* = -1) [inline]
- 7.231.3.19 **TPVSocketEvent** OslTCPSocketI::Send (const uint8 \*& *aPtr*, uint32 *aLen*, int32 *aTimeoutMsec* = -1) [inline]
- 7.231.3.20 **TPVSocketEvent** OslTCPSocketI::Shutdown (**TPVSocketShutdown** *aHow*, int32 *aTimeoutMsec* = -1) [inline]
- 7.231.3.21 **TPVSocketEvent** OslTCPSocketI::ThreadLogoff ()

Reimplemented from [OslIPSocketI](#).

- 7.231.3.22 **TPVSocketEvent** OslTCPSocketI::ThreadLogon (**OslSocketServI** \* *aServ*, **OslSocketObserver** \* *aObserver*)

The documentation for this class was generated from the following file:

- [oscl\\_tcp\\_socket.h](#)

## 7.232 OsciThread Class Reference

```
#include <osci_thread.h>
```

### Public Methods

- OSCI\_IMPORT\_REF [OsciThread](#) ()
- OSCI\_IMPORT\_REF [~OsciThread](#) ()
- OSCI\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) [Create](#) ([TOsciThreadFuncPtr](#) func, int32 stack\_size, [TOsciThreadFuncArg](#) argument, [OsciThread\\_State](#) state=Start\_on\_creation, bool ols-Joinable=false)
- OSCI\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) [GetPriority](#) ([OsciThreadPriority](#) &ref-ThreadPriority)
- OSCI\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) [SetPriority](#) ([OsciThreadPriority](#) ePriority)
- OSCI\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) [Suspend](#) ()
- OSCI\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) [Resume](#) ()
- OSCI\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) [Terminate](#) ([OsciAny](#) \*exitcode)
- OSCI\_IMPORT\_REF [TOsciThreadTerminate](#) [CanTerminate](#) ()

### Static Public Methods

- OSCI\_IMPORT\_REF void [Exit](#) ([OsciAny](#) \*exitcode)
- OSCI\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) [GetId](#) ([TOsciThreadId](#) &refThreadId)
- OSCI\_IMPORT\_REF bool [CompareId](#) ([TOsciThreadId](#) &t1, [TOsciThreadId](#) &t2)
- OSCI\_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 OSCI\_IMPORT\_REF OsciThread::OsciThread ()

Class constructor

#### 7.232.2.2 OSCI\_IMPORT\_REF OsciThread::~~OsciThread ()

Class destructor

### 7.232.3 Member Function Documentation

#### 7.232.3.1 OSCI\_IMPORT\_REF [TOsciThreadTerminate](#) OsciThread::CanTerminate ()

Tell if thread terminate will do join, immediate hard kill, or NOP.

**Returns:**

Terminate behavior.

**7.232.3.2 OSCL\_IMPORT\_REF bool OsciThread::CompareId (TOsciThreadId & t1, TOsciThreadId & t2) [static]**

Static routine to compare whether two thread ID's are equal.

**Parameters:**

*t1, t2*: thread ID passed by the application

**Returns:**

true if equal.

**7.232.3.3 OSCL\_IMPORT\_REF OsciProcStatus::eOsciProcError OsciThread::Create (TOsciThreadFuncPtr func, int32 stack\_size, TOsciThreadFuncArg argument, OsciThread\_State state = Start\_on\_creation, bool oIsJoinable = false)**

This routine will create a thread. The thread may be launched immediately or may be created in a suspended state and launched with a Resume call.

**Parameters:**

*func* = Name of the thread Function *stack\_size* = Size of the thread stack. If zero, then the platform-specific default stack size will be used. *argument* = Argument to be passed to thread function *state* = Enumeration which specifies the state of the thread on creation with values Running and Suspend. Note: the Suspend option may not be available on all platforms. If it is not supported, the Create call will return INVALID\_PARAM\_ERROR. *oIsJoinable* = A boolean, which when set to true, creates a Joinable thread. The default value for this is false, which creates a Detached thread. Note 1: When a joinable thread is created, it is imperative to call thread Terminate. Otherwise, it would cause a memory leak. Note 2: This is currently available only for platforms that have support for pthreads.

**Returns:**

eOsciProcError

**7.232.3.4 OSCL\_IMPORT\_REF void OsciThread::Exit (OsciAny \* exitcode) [static]**

Exit is a static function which is used to end the current thread. When called it just ends the execution of the current thread. 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 OsciProcStatus::eOsciProcError OsciThread::GetId (TOsciThreadId & refThreadId) [static]

Static routine to retrieve ID of calling thread.

#### Parameters:

*Thread* ID passed by the application

#### Returns:

Error code

### 7.232.3.6 OSCL\_IMPORT\_REF OsciProcStatus::eOsciProcError OsciThread::GetPriority (OsciThreadPriority & refThreadPriority)

GetThreadPriority gets the priority of the thread. It takes reference of the input argument and assigns priority to it from one of the already defined priorities.

#### Parameters:

*int16&* refThreadPriority : Output Priority value

#### Returns:

Error code

### 7.232.3.7 OSCL\_IMPORT\_REF OsciProcStatus::eOsciProcError OsciThread::Resume ()

ResumeThread resumes the suspended thread and brings it into execution.

#### Parameters:

*None*

#### Returns:

Error code Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

### 7.232.3.8 OSCL\_IMPORT\_REF OsciProcStatus::eOsciProcError OsciThread::SetPriority (OsciThreadPriority ePriority)

SetThreadPriority sets the priority of the thread. It takes priority as the input argument and assigns it to the thread referred.

#### Parameters:

*ePriorityLevel* : Input Priority value

#### Returns:

Error code Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.232.3.9 OSCL\_IMPORT\_REF void OsciThread::SleepMillisec (const int32 *msec*) [static]**

Suspend current thread execution for specified time.

**Parameters:**

*msec, t2*: sleep time in milliseconds.

**7.232.3.10 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciThread::Suspend ()**

This API suspends the thread being referred. The thread can later be brought into execution by calling OSCL\_ResumeThread() on it.

**Parameters:**

*None*

**Returns:**

Error code Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.232.3.11 OSCL\_IMPORT\_REF [OsciProcStatus::eOsciProcError](#) OsciThread::Terminate ([OsciAny](#) \* *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:

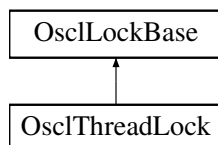
- [osci\\_thread.h](#)



## 7.233 OsciThreadLock Class Reference

```
#include <osci_mutex.h>
```

Inheritance diagram for OsciThreadLock::



### Public Methods

- OSCL\_IMPORT\_REF [OsciThreadLock](#) ()
- virtual OSCL\_IMPORT\_REF [~OsciThreadLock](#) ()
- OSCL\_IMPORT\_REF void [Lock](#) ()
- OSCL\_IMPORT\_REF void [Unlock](#) ()

### 7.233.1 Detailed Description

An implementation of [OsciLockBase](#) using a mutex

### 7.233.2 Constructor & Destructor Documentation

**7.233.2.1** OSCL\_IMPORT\_REF [OsciThreadLock::OsciThreadLock](#) ()

**7.233.2.2** virtual OSCL\_IMPORT\_REF [OsciThreadLock::~~OsciThreadLock](#) () [virtual]

### 7.233.3 Member Function Documentation

**7.233.3.1** OSCL\_IMPORT\_REF void [OsciThreadLock::Lock](#) () [virtual]

Implements [OsciLockBase](#).

**7.233.3.2** OSCL\_IMPORT\_REF void [OsciThreadLock::Unlock](#) () [virtual]

Implements [OsciLockBase](#).

The documentation for this class was generated from the following file:

- [osci\\_mutex.h](#)

## 7.234 OsciTickCount Class Reference

```
#include <osci_tickcount.h>
```

### Static Public Methods

- uint32 [TickCount](#) ()
- uint32 [TickCountFrequency](#) ()
- uint32 [TickCountPeriod](#) ()
- uint32 [TicksToMsec](#) (uint32 ticks)
- uint32 [MsecToTicks](#) (uint32 msec)

### 7.234.1 Detailed Description

OsciTickCount 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 OsciTickCount::MsecToTicks (uint32 msec) [static]

This function converts milliseconds to ticks

**Returns:**

ticks

#### 7.234.2.2 uint32 OsciTickCount::TickCount () [static]

This function returns the current system tick count

**Returns:**

returns the tick count

#### 7.234.2.3 uint32 OsciTickCount::TickCountFrequency () [static]

This function returns the tick frequency in ticks per second

**Returns:**

ticks per second

#### 7.234.2.4 uint32 OsciTickCount::TickCountPeriod () [static]

This function returns the tick period in microseconds per tick

**Returns:**

microseconds per tick

**7.234.2.5** `uint32 OsciTickCount::TicksToMsec (uint32 ticks)` `[static]`

This function converts ticks to milliseconds

**Returns:**

milliseconds

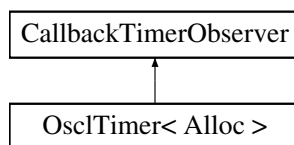
The documentation for this class was generated from the following file:

- [osci\\_tickcount.h](#)

## 7.235 OsciTimer< Alloc > Class Template Reference

```
#include <osci_timer.h>
```

Inheritance diagram for OsciTimer< Alloc >::



### Public Types

- typedef [CallbackTimer](#)< Alloc > [callback\\_timer\\_type](#)

### Public Methods

- [OsciTimer](#) (const char \*name, uint32 frequency=1, int32 priority=OsciActiveObject::EPriority-Nominal)
- virtual [~OsciTimer](#) ()
- void [SetObserver](#) ([OsciTimerObserver](#) \*obs)
- void [SetFrequency](#) (uint32 frequency)
- void [SetExactFrequency](#) (uint32 frequency)
- void [Request](#) (int32 timerID, int32 timeoutInfo, int32 cycles, [OsciTimerObserver](#) \*obs=0, bool recurring=0)
- void [Cancel](#) (int32 timerID, int32 timeoutInfo=-1)
- void [Clear](#) ()

### Protected Methods

- void [TimerBaseElapsed](#) ()

### Friends

- class [CallbackTimer](#)< Alloc >

```
template<class Alloc> class OsciTimer< Alloc >
```

## 7.235.1 Member Typedef Documentation

**7.235.1.1** `template<class Alloc> typedef CallbackTimer<Alloc> OsciTimer< Alloc >::callback_timer_type`

## 7.235.2 Constructor & Destructor Documentation

**7.235.2.1** `template<class Alloc> OsciTimer< Alloc >::OsciTimer (const char * name, uint32 frequency = 1, int32 priority = OsciActiveObject::EPriorityNominal)`

Constructor

### Parameters:

*frequency* The frequency of the timer in cycles/second. A value of 1 means the timer will cycle in 1 second intervals.

**7.235.2.2** `template<class Alloc> OsciTimer< Alloc >::~~OsciTimer () [virtual]`

## 7.235.3 Member Function Documentation

**7.235.3.1** `template<class Alloc> void OsciTimer< Alloc >::Cancel (int32 timerID, int32 timeoutInfo = -1)`

Cancel a timer

### Parameters:

*timerID* used to identify the timer to cancel.

*timeoutInfo* if not set to -1, this value will be used as additional matching criteria to cancel a timer.

**7.235.3.2** `template<class Alloc> void OsciTimer< Alloc >::Clear ()`

Cancel all pending timers.

**7.235.3.3** `template<class Alloc> void OsciTimer< Alloc >::Request (int32 timerID, int32 timeoutInfo, int32 cycles, OsciTimerObserver * obs = 0, bool recurring = 0)`

Request a timer

### Parameters:

*timerID* used to identify the timer for cancellation. This value will be returned as part of the timeout event.

*timeoutInfo* for user info. Returned to the observer on a timeout event

*cycles* the number of cycles to wait before a timeout event. If the timer frequency is 1 and the cycles are set to 2, then the timeout event will occur in 2 seconds.

*obs* a local observer object to be called on a timeout event. This observer overrides the global observer if set.

### 7.235.3.4 `template<class Alloc> void OsciTimer< 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 OsciTimer< 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 OsciTimer< Alloc >::SetObserver (OsciTimerObserver * 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 OsciTimer< 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:

- [osci\\_timer.h](#)

## 7.236 OsciTimerCompare Class Reference

```
#include <osci_scheduler_readyq.h>
```

### Static Public Methods

- int [compare](#) ([TOsciReady](#) &a, [TOsciReady](#) &b)

### 7.236.1 Member Function Documentation

**7.236.1.1** int OsciTimerCompare::compare ([TOsciReady](#) & *a*, [TOsciReady](#) & *b*) [static]

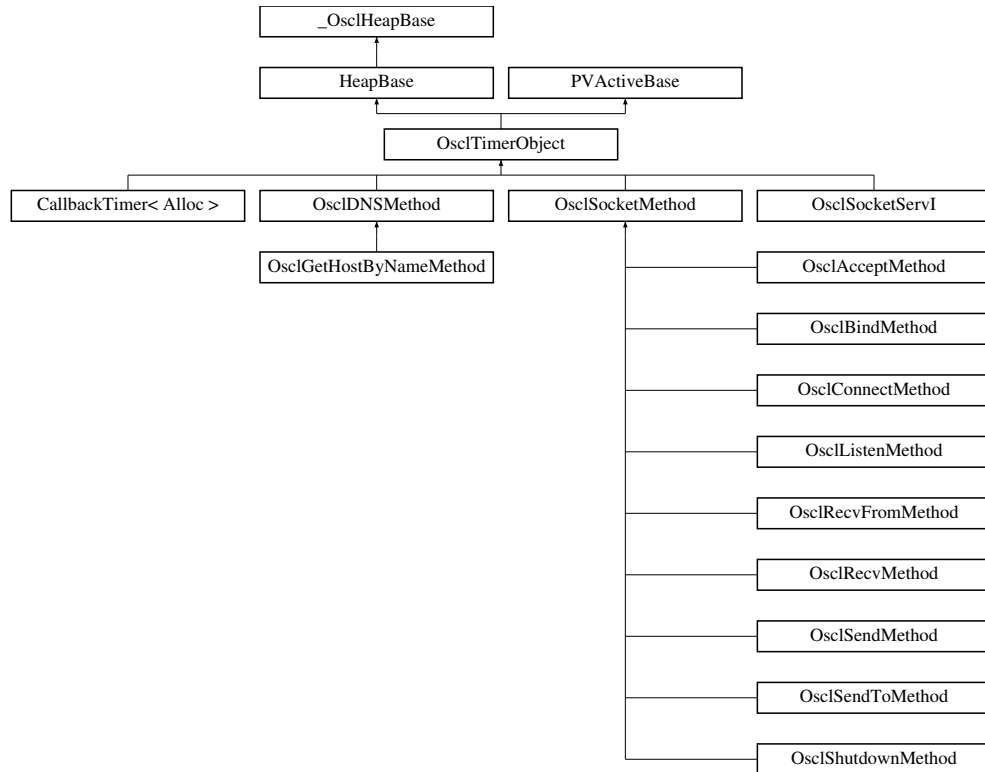
The documentation for this class was generated from the following file:

- [osci\\_scheduler\\_readyq.h](#)

## 7.237 OscTimerObject Class Reference

```
#include <oscl_scheduler_ao.h>
```

Inheritance diagram for OscTimerObject::



### Public Methods

- OSCL\_IMPORT\_REF [OscTimerObject](#) (int32 aPriority, const char name[ ])
- virtual OSCL\_IMPORT\_REF [~OscTimerObject](#) ()
- 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 [OscIAOStatus](#) & [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. OsciTimerObject defines an exec object with a timer.

## 7.237.2 Constructor & Destructor Documentation

### 7.237.2.1 OSCL\_IMPORT\_REF OsciTimerObject::OsciTimerObject (int32 *aPriority*, const char *name*[])

Constructor.

#### Parameters:

- aPriority* (input param): scheduling priority
- name* (input param): optional name for this AO.

### 7.237.2.2 virtual OSCL\_IMPORT\_REF OsciTimerObject::~~OsciTimerObject () [virtual]

Destructor.

## 7.237.3 Member Function Documentation

### 7.237.3.1 OSCL\_IMPORT\_REF void OsciTimerObject::AddToScheduler ()

Add this AO to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

### 7.237.3.2 OSCL\_IMPORT\_REF void OsciTimerObject::After (int32 *aDelayMicrosec*)

'After' sets the request ready, with request status OSCL\_REQUEST\_STATUS\_PENDING, and starts a timer. When the timer expires, the request will complete with status OSCL\_REQUEST\_ERR\_NONE. Must be called from the same thread in which the active object is scheduled. Will leave if the request is already readied, the object is not added to any scheduler, or the calling thread does not match the scheduling thread.

#### Parameters:

- anInterval*: timeout interval in microseconds.

### 7.237.3.3 OSCL\_IMPORT\_REF void OsciTimerObject::Cancel ()

Cancel any active request. If the request is pending, this will call the DoCancel routine, wait for the request to cancel, then set the request idle. The AO will not run. If the request is not active, it does nothing. Request must be canceled from the same thread in which it is scheduled.

Reimplemented from [PVActiveBase](#).

**7.237.3.4** `virtual OSCL_IMPORT_REF void OsciTimerObject::DoCancel ()` [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will cancel the timer. If any additional action is needed, the derived class may override this. If the derived class does override this, it should explicitly call [OsciTimerObject::DoCancel](#) in its own DoCancel routine.

Implements [PVActiveBase](#).

**7.237.3.5** `OSCL_IMPORT_REF bool OsciTimerObject::IsBusy ()`

Return true if this AO is active, false otherwise.

**7.237.3.6** `OSCL_IMPORT_REF int32 OsciTimerObject::Priority ()`

Return scheduling priority of this exec object.

**7.237.3.7** `OSCL_IMPORT_REF void OsciTimerObject::RemoveFromScheduler ()`

Remove this AO from its scheduler. Will leave if the calling thread context does not match the scheduling thread. Cancels any pending request before removing.

Reimplemented from [PVActiveBase](#).

**7.237.3.8** `virtual OSCL_IMPORT_REF int32 OsciTimerObject::RunError (int32 aError)` [protected, virtual]

Run Leave handler. This gets called by scheduler when the Run routine leaves. The default implementation simply returns the leave code. If the derived class wants to handle errors from Run, it may override this. The ExecError should return OsciErrNone if it handles the error, otherwise it should return the input error code.

**Parameters:**

*aError*: the leave code generated by the Run.

Implements [PVActiveBase](#).

**7.237.3.9** `OSCL_IMPORT_REF void OsciTimerObject::RunIfNotReady (uint32 aDelayMicrosec = 0)`

Complete the request after a time interval. RunIfNotReady is identical to [After\(\)](#) except that it first checks the request status, and if it is already readied, it does nothing.

**Parameters:**

*aDelayMicrosec* (input param): delay in microseconds.

**7.237.3.10** `OSCL_IMPORT_REF void OsciTimerObject::SetBusy ()`

Set request ready for this AO. Will leave if the request is already readied, or the exec object is not added to any scheduler, or the calling thread context does not match the scheduler thread.

**7.237.3.11** OSCL\_IMPORT\_REF void OsciTimerObject::SetStatus (int32)

**7.237.3.12** OSCL\_IMPORT\_REF int32 OsciTimerObject::Status ()

Request status access

**7.237.3.13** OSCL\_IMPORT\_REF [OsciAOSStatus&](#) OsciTimerObject::StatusRef ()

The documentation for this class was generated from the following file:

- [osci\\_scheduler\\_ao.h](#)

## 7.238 OsciTimerObserver Class Reference

```
#include <osci_timer.h>
```

### Public Methods

- virtual void [TimeoutOccurred](#) (int32 timerID, int32 timeoutInfo)=0
- virtual [~OsciTimerObserver](#) ()

### 7.238.1 Detailed Description

The observer class to receive timeout callbacks

### 7.238.2 Constructor & Destructor Documentation

**7.238.2.1** virtual OsciTimerObserver::~OsciTimerObserver () [inline, virtual]

### 7.238.3 Member Function Documentation

**7.238.3.1** virtual void OsciTimerObserver::TimeoutOccurred (int32 *timerID*, int32 *timeoutInfo*)  
[pure virtual]

This function will be called when the timer associated with this observer is executed

#### Parameters:

- timerID* The ID given at timer request.
- timeoutInfo* Any extra info given at timer request.

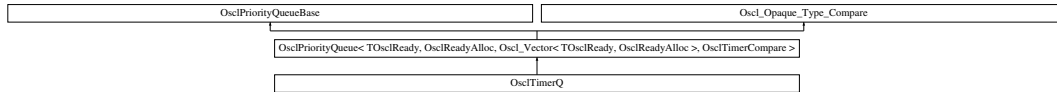
The documentation for this class was generated from the following file:

- [osci\\_timer.h](#)

## 7.239 OsciTimerQ Class Reference

```
#include <osci_scheduler_readyq.h>
```

Inheritance diagram for OsciTimerQ::



### Public Methods

- void [Construct](#) (int)
- void [Add](#) ([TOsciReady](#))
- void [Remove](#) ([TOsciReady](#))
- [TOsciReady](#) [PopTop](#) ()
- [TOsciReady](#) [Top](#) ()
- void [Pop](#) ([TOsciReady](#))
- bool [IsIn](#) ([TOsciReady](#))

### 7.239.1 Member Function Documentation

7.239.1.1 void OsciTimerQ::Add ([TOsciReady](#))

7.239.1.2 void OsciTimerQ::Construct (int)

7.239.1.3 bool OsciTimerQ::IsIn ([TOsciReady](#))

7.239.1.4 void OsciTimerQ::Pop ([TOsciReady](#))

7.239.1.5 [TOsciReady](#) OsciTimerQ::PopTop ()

7.239.1.6 void OsciTimerQ::Remove ([TOsciReady](#))

7.239.1.7 [TOsciReady](#) OsciTimerQ::Top ()

The documentation for this class was generated from the following file:

- [osci\\_scheduler\\_readyq.h](#)

## 7.240 OsciTLS< T, ID, Registry > Class Template Reference

```
#include <osci_tls.h>
```

### Public Methods

- [OsciTLS \(\)](#)
- [~OsciTLS \(\)](#)
- [T & operator \\* \(\) const](#)  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- [T \\* operator → \(\) const](#)  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- [bool set \(\)](#)  
*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- [T \\* \\_Ptr](#)

```
template<class T, uint32 ID, class Registry = OsciTLSRegistry> class OsciTLS< T, ID, Registry >
```

### 7.240.1 Constructor & Destructor Documentation

**7.240.1.1** `template<class T, uint32 ID, class Registry = OsciTLSRegistry> OsciTLS< T, ID, Registry >::OsciTLS () [inline]`

**7.240.1.2** `template<class T, uint32 ID, class Registry = OsciTLSRegistry> OsciTLS< T, ID, Registry >::~~OsciTLS () [inline]`

### 7.240.2 Member Function Documentation

**7.240.2.1** `template<class T, uint32 ID, class Registry = OsciTLSRegistry> T& OsciTLS< T, ID, Registry >::operator * () const [inline]`

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the OsciTLS can be used like the regular pointer that it was initialized with.

**7.240.2.2** `template<class T, uint32 ID, class Registry = OsciTLSRegistry> T* OsciTLS< T, ID, Registry >::operator → () const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsciTLS can be used like the regular pointer that it was initialized with.

**7.240.2.3** `template<class T, uint32 ID, class Registry = OsciTLSRegistry> bool OsciTLS< T, ID, Registry >::set () [inline]`

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

### 7.240.3 Field Documentation

**7.240.3.1** `template<class T, uint32 ID, class Registry = OsciTLSRegistry> T* OsciTLS< T, ID, Registry >::_Ptr [protected]`

The documentation for this class was generated from the following file:

- [osci\\_tls.h](#)

## 7.241 OsciTLSEx< T, ID, Registry > Class Template Reference

```
#include <osci_error.h>
```

### Public Methods

- [OsciTLSEx \(\)](#)
- [~OsciTLSEx \(\)](#)
- [T & operator \\* \(\) const](#)

*The indirection operator (\*) accesses a value indirectly, through a pointer.*

- [T \\* operator → \(\) const](#)

*The indirection operator (->) accesses a value indirectly, through a pointer.*

- [bool set \(\)](#)

*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- [T \\* \\_Ptr](#)

```
template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> class OsciTLSEx< T, ID, Registry >
```

#### 7.241.1 Constructor & Destructor Documentation

**7.241.1.1** `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> OsciTLSEx< T, ID, Registry >::OsciTLSEx () [inline]`

**7.241.1.2** `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> OsciTLSEx< T, ID, Registry >::~~OsciTLSEx () [inline]`

#### 7.241.2 Member Function Documentation

**7.241.2.1** `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> T& OsciTLSEx< T, ID, Registry >::operator * () const [inline]`

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the [OsciTLS](#) can be used like the regular pointer that it was initialized with.

**7.241.2.2** `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> T* OsciTLSEx< T, ID, Registry >::operator → () const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the [OsciTLS](#) can be used like the regular pointer that it was initialized with.



**7.241.2.3** `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> bool OsciTLSEx< T, ID, Registry >::set () [inline]`

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

### 7.241.3 Field Documentation

**7.241.3.1** `template<class T, uint32 ID, class Registry = OsciTLSRegistryEx> T* OsciTLSEx< T, ID, Registry >::_Ptr [protected]`

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 7.242 OsciTLSRegistry Class Reference

```
#include <osci_tls.h>
```

### Static Public Methods

- OSCL\_IMPORT\_REF [OsciAny](#) \* [getInstance](#) (uint32 ID, int32 &error)
- OSCL\_IMPORT\_REF void [registerInstance](#) ([OsciAny](#) \*ptr, uint32 ID, int32 &error)

### Friends

- class [OsciBase](#)

### 7.242.1 Member Function Documentation

**7.242.1.1** OSCL\_IMPORT\_REF [OsciAny](#)\* [OsciTLSRegistry::getInstance](#) (uint32 *ID*, int32 &*error*) [static]

**7.242.1.2** OSCL\_IMPORT\_REF void [OsciTLSRegistry::registerInstance](#) ([OsciAny](#) \**ptr*, uint32 *ID*, int32 &*error*) [static]

### 7.242.2 Friends And Related Function Documentation

**7.242.2.1** friend class [OsciBase](#) [friend]

The documentation for this class was generated from the following file:

- [osci\\_tls.h](#)

## 7.243 OsciTLSRegistryEx Class Reference

```
#include <osci_error.h>
```

### Static Public Methods

- [OsciAny](#) \* [getInstance](#) (uint32 ID)
- void [registerInstance](#) ([OsciAny](#) \*ptr, uint32 ID)

### 7.243.1 Member Function Documentation

**7.243.1.1**   [OsciAny](#)\* [OsciTLSRegistryEx::getInstance](#) (uint32 *ID*)   [inline, static]

**7.243.1.2**   void [OsciTLSRegistryEx::registerInstance](#) ([OsciAny](#) \**ptr*, uint32 *ID*)   [inline, static]

The documentation for this class was generated from the following file:

- [osci\\_error.h](#)

## 7.244 OsciTrapItem Class Reference

```
#include <osci_heapbase.h>
```

### Public Methods

- OSCL\_INLINE [OsciTrapItem](#) ([OsciTrapOperation](#) anOperation)
- OSCL\_INLINE [OsciTrapItem](#) ([OsciTrapOperation](#) anOperation, [OsciAny](#) \*aPtr)

### Friends

- class [OsciTrapStackItem](#)
- class [OsciTrapStack](#)

### 7.244.1 Constructor & Destructor Documentation

7.244.1.1 OSCL\_INLINE [OsciTrapItem::OsciTrapItem](#) ([OsciTrapOperation](#) *anOperation*)

7.244.1.2 OSCL\_INLINE [OsciTrapItem::OsciTrapItem](#) ([OsciTrapOperation](#) *anOperation*, [OsciAny](#) \* *aPtr*)

### 7.244.2 Friends And Related Function Documentation

7.244.2.1 friend class [OsciTrapStack](#) [friend]

7.244.2.2 friend class [OsciTrapStackItem](#) [friend]

The documentation for this class was generated from the following file:

- [osci\\_heapbase.h](#)

## 7.245 OsciTrapStack Class Reference

```
#include <osci_error_trapcleanup.h>
```

### Friends

- class [OsciError](#)
- class [OsciErrorTrap](#)
- class [OsciErrorTrapImp](#)

### 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 `OsciError` [friend]

**7.245.2.2** friend class `OsciErrorTrap` [friend]

**7.245.2.3** friend class `OsciErrorTrapImp` [friend]

The documentation for this class was generated from the following file:

- [osci\\_error\\_trapcleanup.h](#)

## 7.246 OsciTrapStackItem Class Reference

```
#include <osci_error_trapcleanup.h>
```

### Public Methods

- [OsciTrapStackItem \(\)](#)
- [OsciTrapStackItem \(\\_OsciHeapBase \\*aCBase\)](#)
- [OsciTrapStackItem \(OsciAny \\*aTAny\)](#)
- [OsciTrapStackItem \(OsciTrapItem aItem\)](#)

### Data Fields

- [\\_OsciHeapBase \\* iCBase](#)
- [OsciAny \\* iTAny](#)
- [OsciTrapOperation iTrapOperation](#)
- [OsciTrapStackItem \\* iNext](#)

### 7.246.1 Detailed Description

Internal cleanup stack item type.

### 7.246.2 Constructor & Destructor Documentation

**7.246.2.1** [OsciTrapStackItem::OsciTrapStackItem \(\)](#) [inline]

**7.246.2.2** [OsciTrapStackItem::OsciTrapStackItem \(\\_OsciHeapBase \\* aCBase\)](#) [inline]

**7.246.2.3** [OsciTrapStackItem::OsciTrapStackItem \(OsciAny \\* aTAny\)](#) [inline]

**7.246.2.4** [OsciTrapStackItem::OsciTrapStackItem \(OsciTrapItem aItem\)](#) [inline]

### 7.246.3 Field Documentation

**7.246.3.1** [\\_OsciHeapBase\\*](#) [OsciTrapStackItem::iCBase](#)

**7.246.3.2** [OsciTrapStackItem\\*](#) [OsciTrapStackItem::iNext](#)

**7.246.3.3** [OsciAny\\*](#) [OsciTrapStackItem::iTAny](#)

**7.246.3.4** [OsciTrapOperation](#) [OsciTrapStackItem::iTrapOperation](#)

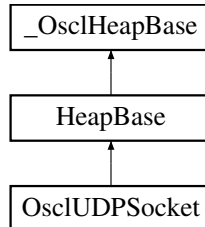
The documentation for this class was generated from the following file:

- [osci\\_error\\_trapcleanup.h](#)

## 7.247 OsciUDPSocket Class Reference

```
#include <osci_socket.h>
```

Inheritance diagram for OsciUDPSocket::



### Public Methods

- OSCL\_IMPORT\_REF [~OsciUDPSocket](#) ()
- OSCL\_IMPORT\_REF [TPVSocketEvent ThreadLogoff](#) ()
- OSCL\_IMPORT\_REF [TPVSocketEvent ThreadLogon](#) ([OsciSocketServ](#) &aServ, [OsciSocketObserver](#) \*aObserver)
- OSCL\_IMPORT\_REF int32 [Close](#) ()
- OSCL\_IMPORT\_REF int32 [Bind](#) ([OsciNetworkAddress](#) &aAddress)
- OSCL\_IMPORT\_REF int32 [Join](#) ([OsciNetworkAddress](#) &aAddress)
- OSCL\_IMPORT\_REF int32 [JoinMulticastGroup](#) ([OsciIpMReq](#) &aMReq)
- OSCL\_IMPORT\_REF int32 [SetMulticastTTL](#) (int32 aTTL)
- OSCL\_IMPORT\_REF int32 [SetOptionToReuseAddress](#) ()
- OSCL\_IMPORT\_REF int32 [SetTOS](#) (const [OsciSocketTOS](#) &aTOS)
- OSCL\_IMPORT\_REF int32 [GetPeerName](#) ([OsciNetworkAddress](#) &aPeerName)
- OSCL\_IMPORT\_REF [TPVSocketEvent BindAsync](#) ([OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void [CancelBind](#) ()
- OSCL\_IMPORT\_REF uint8 \* [GetRecvData](#) (int32 \*aLength)
- OSCL\_IMPORT\_REF uint8 \* [GetSendData](#) (int32 \*aLength)
- OSCL\_IMPORT\_REF [TPVSocketEvent SendTo](#) (const uint8 \*aPtr, uint32 aLen, [OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void [CancelSendTo](#) ()
- OSCL\_IMPORT\_REF [TPVSocketEvent RecvFrom](#) (uint8 \*aPtr, uint32 aMaxLen, [OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiRecvLimit=0, [Osci\\_Vector](#)< uint32, [OsciMemAllocator](#) > \*aPacketLen=NULL, [Osci\\_Vector](#)< [OsciNetworkAddress](#), [OsciMemAllocator](#) > \*aPacketSource=NULL)
- OSCL\_IMPORT\_REF void [CancelRecvFrom](#) ()
- OSCL\_IMPORT\_REF int32 [SetRecvBufferSize](#) (uint32 size)

### Static Public Methods

- OSCL\_IMPORT\_REF [OsciUDPSocket \\*](#) [NewL](#) ([Osci\\_DefAlloc](#) &alloc, [OsciSocketServ](#) &aServ, [OsciSocketObserver](#) \*aObserver, uint32 aId)

## 7.247.1 Detailed Description

The UDP Socket class

## 7.247.2 Constructor & Destructor Documentation

### 7.247.2.1 OSCL\_IMPORT\_REF OsciUDPSocket::~~OsciUDPSocket ()

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 OsciUDPSocket::Bind (OsciNetworkAddress & aAddress)

Bind a UDP socket to an address. This is a synchronous method.

#### Parameters:

*aAddress*: Bind address.

#### Returns:

Returns OsciErrNone for success, or a platform-specific error code.

### 7.247.3.2 OSCL\_IMPORT\_REF TPVSocketEvent OsciUDPSocket::BindAsync (OsciNetworkAddress & aAddress, int32 aTimeoutMsec = (-1))

Bind a UDP socket to an address. This is an asynchronous method.

#### Parameters:

*aAddress*: Bind address.

*aTimeoutMsec*: Optional timeout. Use a negative value for infinite wait.

#### Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

### 7.247.3.3 OSCL\_IMPORT\_REF void OsciUDPSocket::CancelBind ()

Cancel Bind

This method will cancel any pending BindAsync operation on the current socket, causing the BindAsync to complete with error EPVSocketCancel. If there is no pending BindAsync operation, this method will have no effect.

### 7.247.3.4 OSCL\_IMPORT\_REF void OsciUDPSocket::CancelRecvFrom ()

Cancel RecvFrom

This method will cancel any pending RecvFrom operation on the current socket, causing the RecvFrom to complete with error EPVSocketCancel. If there is no pending RecvFrom operation, this method will have no effect.



**7.247.3.5 OSCL\_IMPORT\_REF void OslUDPSocket::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 OslUDPSocket::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 OslErrNone for success, or a platform-specific error code.

**7.247.3.7 OSCL\_IMPORT\_REF int32 OslUDPSocket::GetPeerName (OslNetworkAddress & aPeerName)**

Retrieves the peer address of the socket

**Parameters:**

*aPeerName:* This will store the peer address when API returns successfully.

**Returns:**

Returns OslErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSL for the underlying OS

**7.247.3.8 OSCL\_IMPORT\_REF uint8\* OslUDPSocket::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\* OslUDPSocket::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 OsciUDPSocket::Join (OsciNetworkAddress & aAddress)

Bind a UDP socket to an address and Join the multicast group. This is a synchronous method.

#### Parameters:

*aAddress*: Bind address.

#### Returns:

Returns OsciErrNone for success, or a platform-specific error code. May throw an OsciErrNotSupported Exception

### 7.247.3.11 OSCL\_IMPORT\_REF int32 OsciUDPSocket::JoinMulticastGroup (OsciIpMReq & aMReq)

Join the multicast group.

#### Parameters:

*aMReq*: Multicast group information.

#### Returns:

Returns: OsciErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK\_ERR\_BAD\_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

### 7.247.3.12 OSCL\_IMPORT\_REF OsciUDPSocket\* OsciUDPSocket::NewL (Osci\_DefAlloc & alloc, OsciSocketServ & aServ, OsciSocketObserver \* aObserver, uint32 aId) [static]

Create a UDP Socket. May leave if failure.

#### Parameters:

*alloc*: Memory allocator.

*aServ*: Socket server. Must be connected.

*aObserver*: Socket observer.

*aId*: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

#### Returns:

Returns pointer to socket.

### 7.247.3.13 OSCL\_IMPORT\_REF TPVSocketEvent OsciUDPSocket::RecvFrom (uint8 \* aPtr, uint32 aMaxLen, OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiRecvLimit = 0, Osci\_Vector< uint32, OsciMemAllocator > \* aPacketLen = NULL, Osci\_Vector< OsciNetworkAddress, OsciMemAllocator > \* aPacketSource = NULL)

Receive Data. This is an asynchronous method.

**Parameters:**

***aPtr:*** Buffer to receive incoming data

***aMaxLen:*** Length of buffer.

***aAddress:*** (output) Source address.

***aTimeoutMsec:*** Timeout in milliseconds, or (-1) for infinite wait.

***aMultiRecvLimit*** (optional input): Configures multiple packet receive mode. As long as there are packets queued at the socket and at least *aMultiRecvLimit* bytes are available in the buffer, recvfrom operations will continue. A value of zero disabled multiple packet mode. The individual packet lengths can be retrieved in the *aPacketLen* parameter; and the individual packet source addresses can be retrieved in the *aPacketSource* parameter.

***aPacketLen:*** (optional output) a vector of packet lengths, in case multiple packets were received.

***aPacketSource:*** (optional output) a vector of source addresses, in case multiple packets were received.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

### 7.247.3.14 OSCL\_IMPORT\_REF TPVSocketEvent OsciUDPSocket::SendTo (const uint8 \* *aPtr*, uint32 *aLen*, OsciNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1)

Send Data. This is an asynchronous method.

**Parameters:**

***aPtr:*** Data to send.

***aLen:*** Length of data to send.

***aAddress:*** Destination address.

***aTimeoutMsec:*** Timeout in milliseconds, or (-1) for infinite wait.

**Returns:**

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

### 7.247.3.15 OSCL\_IMPORT\_REF int32 OsciUDPSocket::SetMulticastTTL (int32 *aTTL*)

Controls the number of intermediate systems through which a multicast datagram can be forwarded.

**Parameters:**

***aTTL:Specifies*** the time-to-live value for multicast datagrams sent through this socket.

**Returns:**

Returns: OsciErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK\_ERR\_BAD\_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

### 7.247.3.16 OSCL\_IMPORT\_REF int32 OslUDPSocket::SetOptionToReuseAddress ()

Allows the server to bind to an address which is in a TIME\_WAIT state.

#### Returns:

Returns: OslErrNone 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 OslUDPSocket::SetRecvBufferSize (uint32 size)

Set the buffer size of the socket This is a synchronous method.

#### Parameters:

*size*: buffer size

#### Returns:

Returns OslErrNone for success, or a platform-specific error code. May throw an OslErrNotSupported Exception.

### 7.247.3.18 OSCL\_IMPORT\_REF int32 OslUDPSocket::SetTOS (const OslSocketTOS & aTOS)

Sets the Type of Service field of each outgoing IP datagram.

#### Parameters:

*aTOS*: Specifies the type of service requested.

#### Returns:

Returns: OslErrNone 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 OslUDPSocket::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 OslUDPSocket::ThreadLogon (OslSocketServ & aServ, OslSocketObserver \* 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.

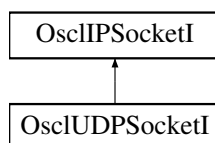
The documentation for this class was generated from the following file:

- [oscl\\_socket.h](#)

## 7.248 OsciUDPSocketI Class Reference

```
#include <osci_udp_socket.h>
```

Inheritance diagram for OsciUDPSocketI::



### Public Methods

- virtual [~OsciUDPSocketI](#) ()
- int32 [Close](#) ()
- int32 [JoinMulticastGroup](#) ([OsciIpMReq](#) &aMReq)
- int32 [SetMulticastTTL](#) (int32 aTTL)
- uint8 \* [GetRecvData](#) (int32 \*aLength)
- uint8 \* [GetSendData](#) (int32 \*aLength)
- [TPVSocketEvent](#) [ThreadLogoff](#) ()
- [TPVSocketEvent](#) [ThreadLogon](#) ([OsciSocketServI](#) \*aServ, [OsciSocketObserver](#) \*aObserver)
- [TPVSocketEvent](#) [BindAsync](#) ([OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=-1)
- void [CancelBind](#) ()
- [TPVSocketEvent](#) [SendTo](#) (const uint8 \*&aPtr, uint32 aLen, [OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=-1)
- void [CancelSendTo](#) ()
- [TPVSocketEvent](#) [RecvFrom](#) (uint8 \*&aPtr, uint32 aMaxLen, [OsciNetworkAddress](#) &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiMaxLen=0, [Osci\\_Vector](#)< uint32, [OsciMemAllocator](#) > \*aPacketLen=NULL, [Osci\\_Vector](#)< [OsciNetworkAddress](#), [OsciMemAllocator](#) > \*aPacketSource=NULL)
- void [CancelRecvFrom](#) ()

### Static Public Methods

- [OsciUDPSocketI](#) \* [NewL](#) ([Osci\\_DefAlloc](#) &a, [OsciSocketServI](#) \*aServ, [OsciSocketObserver](#) \*aObserver, uint32 aId)

### 7.248.1 Detailed Description

Internal implementation class for [OsciUDPSocket](#)

## 7.248.2 Constructor & Destructor Documentation

7.248.2.1 `virtual OsciUDPSocketI::~~OsciUDPSocketI ()` [virtual]

## 7.248.3 Member Function Documentation

7.248.3.1 **TPVSocketEvent** `OsciUDPSocketI::BindAsync (OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1)` [inline]

7.248.3.2 `void OsciUDPSocketI::CancelBind ()` [inline]

7.248.3.3 `void OsciUDPSocketI::CancelRecvFrom ()` [inline]

7.248.3.4 `void OsciUDPSocketI::CancelSendTo ()` [inline]

7.248.3.5 `int32 OsciUDPSocketI::Close ()` [virtual]

Implements [OsciIPSocketI](#).

7.248.3.6 `uint8 * OsciUDPSocketI::GetRecvData (int32 * aLength)` [inline, virtual]

Implements [OsciIPSocketI](#).

7.248.3.7 `uint8 * OsciUDPSocketI::GetSendData (int32 * aLength)` [inline, virtual]

Implements [OsciIPSocketI](#).

7.248.3.8 `int32 OsciUDPSocketI::JoinMulticastGroup (OsciIpMReq & aMReq)`

7.248.3.9 `OsciUDPSocketI* OsciUDPSocketI::NewL (Osci\_DefAlloc & a, OsciSocketServI * aServ, OsciSocketObserver * aObserver, uint32 aId)` [static]

7.248.3.10 **TPVSocketEvent** `OsciUDPSocketI::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiMaxLen = 0, Osci\_Vector< uint32, OsciMemAllocator > * aPacketLen = NULL, Osci\_Vector< OsciNetworkAddress, OsciMemAllocator > * aPacketSource = NULL)` [inline]

7.248.3.11 **TPVSocketEvent** `OsciUDPSocketI::SendTo (const uint8 *& aPtr, uint32 aLen, OsciNetworkAddress & aAddress, int32 aTimeoutMsec = -1)` [inline]

7.248.3.12 `int32 OsciUDPSocketI::SetMulticastTTL (int32 aTTL)`

7.248.3.13 **TPVSocketEvent** `OsciUDPSocketI::ThreadLogoff ()`

Reimplemented from [OsciIPSocketI](#).

7.248.3.14 **TPVSocketEvent** `OsciUDPSocketI::ThreadLogon (OsciSocketServI * aServ, OsciSocketObserver * aObserver)`

The documentation for this class was generated from the following file:

- [osci\\_udp\\_socket.h](#)

## 7.249 OslUuid Struct Reference

```
#include <osl_uuid.h>
```

### Public Methods

- [OslUuid](#) ()
- [OslUuid](#) (uint32 l, uint16 w1, uint16 w2, uint8 b1, uint8 b2, uint8 b3, uint8 b4, uint8 b5, uint8 b6, uint8 b7, uint8 b8)
- [OslUuid](#) (const OslUuid &uuid)
- OslUuid & [operator=](#) (const OslUuid &src)
- bool [operator==](#) (const OslUuid &src) const
- bool [operator!=](#) (const OslUuid &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 `OslUuid::OslUuid()` `[inline]`

7.249.2.2 `OslUuid::OslUuid (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 `OslUuid::OslUuid (const OslUuid & uuid)` `[inline]`

## 7.249.3 Member Function Documentation

7.249.3.1 `bool OslUuid::operator!= (const OslUuid & src) const` `[inline]`

7.249.3.2 `OslUuid& OslUuid::operator= (const OslUuid & src)` `[inline]`

7.249.3.3 `bool OslUuid::operator== (const OslUuid & src) const` `[inline]`

## 7.249.4 Field Documentation

7.249.4.1 `uint32 OslUuid::data1`

7.249.4.2 `uint16 OslUuid::data2`

7.249.4.3 `uint16 OslUuid::data3`

7.249.4.4 `uint8 OslUuid::data4[BYTES_IN_UUID_ARRAY]`

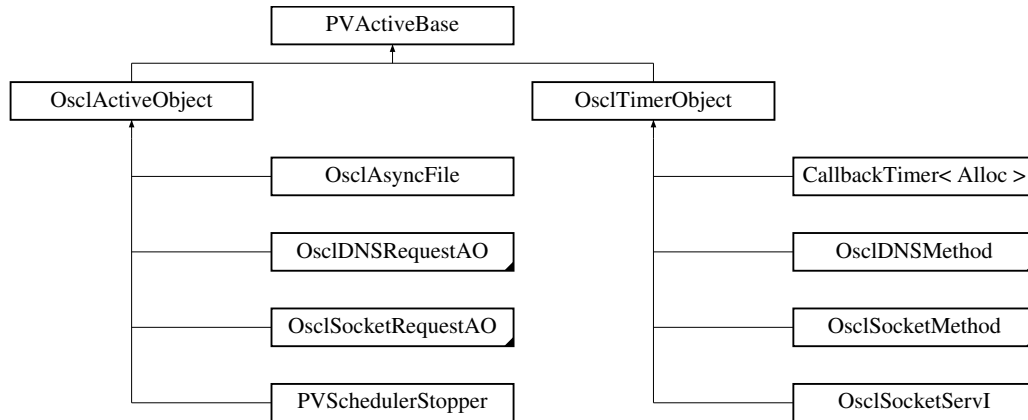
The documentation for this struct was generated from the following file:

- [osl\\_uuid.h](#)

## 7.250 PActiveBase Class Reference

```
#include <oscl_scheduler_aobase.h>
```

Inheritance diagram for PActiveBase::



### Public Methods

- [PActiveBase](#) (const char name[ ], int32 pri)
- virtual [~PActiveBase](#) ()
- bool [IsInAnyQ](#) ()
- virtual int32 [RunError](#) (int32 aError)=0
- virtual void [Run](#) ()=0
- virtual void [DoCancel](#) ()=0
- void [AddToScheduler](#) ()
- void [RemoveFromScheduler](#) ()
- void [Destroy](#) ()
- void [Activate](#) ()
- OSCL\_IMPORT\_REF bool [IsAdded](#) () const
- void [Cancel](#) ()

### Data Fields

- uint32 [iAddedNum](#)
- [OsciNameString](#)< PVEXECNAMELEN > [iName](#)
- [PVThreadContext](#) [iThreadContext](#)
- [PActiveStats](#) \* [iPActiveStats](#)
- [TReadyQueLink](#) [iPVReadyQLink](#)
- bool [iBusy](#)
- [OsciAOStatus](#) [iStatus](#)

### Friends

- class [PActiveStats](#)
- class [OsciSchedulerCommonBase](#)

- class [OscActiveObject](#)
- class [OscTimerObject](#)
- class [OscReadyQ](#)
- class [OscReadyCompare](#)
- class [OscReadySetPosition](#)
- class [OscExecScheduler](#)

## 7.250.1 Detailed Description

PV Scheduler internal AO base class. Both [OscActiveObject](#) and [OscTimerObject](#) derive from this class. For Symbian, this just container has the desired additions to the basic CTimer or OscActiveObj functionality. For non-Symbian, this class contains the entire AO implementation.

## 7.250.2 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 [OscActiveObject](#), and [OscTimerObject](#).

**7.250.3.3** `void PVActiveBase::Cancel ()`

Reimplemented in [OscActiveObject](#), and [OscTimerObject](#).

**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 [OscIDNSRequestAO](#), [OscSocketRequestAO](#), [OscActiveObject](#), and [OscTimerObject](#).

**7.250.3.6** `OSCL_IMPORT_REF bool PActiveBase::IsAdded ()`

**7.250.3.7** `bool PActiveBase::IsInAnyQ () [inline]`

**7.250.3.8** `void PActiveBase::RemoveFromScheduler ()`

Reimplemented in [OscActiveObject](#), and [OscTimerObject](#).

**7.250.3.9** `virtual void PActiveBase::Run () [pure virtual]`

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's [Run\(\)](#) function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implemented in [OscIDNSMethod](#), [OscIDNSRequestAO](#), [OscSocketMethod](#), [OscSocketRequestAO](#), and [CallbackTimer< Alloc >](#).

**7.250.3.10** `virtual int32 PActiveBase::RunError (int32 aError) [pure virtual]`

Virtual routine that gets called if the active object's `Run` routine leaves.

**Parameters:**

*aError*: the leave code generated by the `Run`.

**Returns:**

:returns `OscErrNone` if the error was handled, or returns the input `aError` value if not handled.

Implemented in [OscActiveObject](#), and [OscTimerObject](#).

## 7.250.4 Friends And Related Function Documentation

**7.250.4.1** friend class `OscActiveObject` [friend]

**7.250.4.2** friend class `OscExecScheduler` [friend]

**7.250.4.3** friend class `OscReadyCompare` [friend]

**7.250.4.4** friend class `OscReadyQ` [friend]

**7.250.4.5** friend class `OscReadySetPosition` [friend]

**7.250.4.6** friend class `OscSchedulerCommonBase` [friend]

**7.250.4.7** friend class `OscTimerObject` [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** `OscNameString<PVEXECNAMELEN>` `PVActiveBase::iName`

**7.250.5.4** `PVActiveStats*` `PVActiveBase::iPVActiveStats`

**7.250.5.5** `TReadyQueLink` `PVActiveBase::iPVReadyQLink`

**7.250.5.6** `OscIAOStatus` `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 [PActiveBase](#)
- class [OscExecScheduler](#)
- class [OscExecSchedulerCommonBase](#)
- class [OscActiveObject](#)
- class [OscTimerObject](#)
- class [OscReadyQ](#)

### 7.251.1 Detailed Description

PV AO statistics

### 7.251.2 Friends And Related Function Documentation

**7.251.2.1** friend class `OscActiveObject` [friend]

**7.251.2.2** friend class `OscExecScheduler` [friend]

**7.251.2.3** friend class `OscExecSchedulerCommonBase` [friend]

**7.251.2.4** friend class `OscReadyQ` [friend]

**7.251.2.5** friend class `OscTimerObject` [friend]

**7.251.2.6** friend class `PActiveBase` [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 [\\_OscBasicAllocator](#) [alloc\\_type](#)

### Public Methods

- void [SetLogLevel](#) ([log\\_level\\_type](#) level)
- OSCL\_IMPORT\_REF void [SetLogLevelAndPropagate](#) ([log\\_level\\_type](#) level)
- [log\\_level\\_type](#) [GetLogLevel](#) ()
- void [DisableAppenderInheritance](#) ()
- void [AddAppender](#) ([OscSharedPtr](#)< [PVLoggerAppender](#) > &appender)
- void [RemoveAppender](#) ([OscSharedPtr](#)< [PVLoggerAppender](#) > &appender)
- void [AddFilter](#) ([OscSharedPtr](#)< [PVLoggerFilter](#) > &filter)
- uint32 [GetNumAppendors](#) ()
- 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 [\\_OscBasicAllocator](#) 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 ([OscSharedPtr](#)< [PVLoggerAppender](#) > & *appender*) [inline]

This method adds an appender to the logging control point. Each logger maintains a list of appenders. Any msg to a logger if deemed active is logged to all the appenders.

### Parameters:

*appender* pointer to the appender to add

### Returns:

NONE

### Exceptions:

*leaves* if out of memory

7.252.3.2 void PVLogger::AddFilter ([OscSharedPtr](#)< [PVLoggerFilter](#) > & *filter*) [inline]

This method adds a message filter to the logging control point. Each logger maintains a list of filters. Any msg to a logger if deemed active is passed through the msg filters prior to logging.



**Parameters:**

*msgFilter* pointer to the filter to add

**Returns:**

NONE

**Exceptions:**

*leaves* if out of memory

### 7.252.3.3 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 (OsciSharedPtr< PVLoggerAppender > &appender) [inline]**

This method removes an appender from the logging control point. Each logger maintains a list of appenders. Any msg to a logger if deemed active is logged to all the appenders.

**Parameters:**

*appender* pointer to the appender to delete

**Returns:**

NONE

**7.252.3.16** 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]

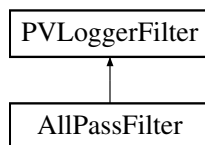
The documentation for this class was generated from the following file:

- [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](#) FilterOpaqueMessage (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](#) ([Osci\\_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 (Osl\_TagTree< PVLogger \*, alloc\_type >::node\_type \* *node*, log\_level\_type *level*)

This method recursively propagates the log level to all the descendents, of a node.

#### Parameters:

*node* Node ptr, associated with a logger, from the tag tree.

*level* log level associated with the logging control point

#### Returns:

NONE

**7.256.4.5 OSCL\_IMPORT\_REF bool PVLoggerRegistry::SetNodeLogLevelExplicit (char \* *tagIn*, [log\\_level\\_type](#) *level*)**

This method propagates the log level to all the descendents of the node, with a specified tag.

**Parameters:**

*tagIn* logger tag, viz. "x.y.z"

*level* log level associated with the logging control point

**Returns:**

true on success, else false.

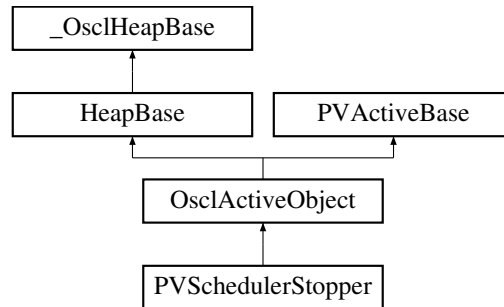
The documentation for this class was generated from the following file:

- [pvlogger\\_registry.h](#)

## 7.257 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 [OscActiveObject](#)
- class [OscTimerObject](#)
- class [OscExecScheduler](#)
- class [OscCoeActiveScheduler](#)
- class [OscExecSchedulerCommonBase](#)
- class [OscExecSchedulerBase](#)
- class [OscCoeActiveSchedulerBase](#)

### 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 Osci scheduler or native scheduler.

**7.260.3 Friends And Related Function Documentation**

**7.260.3.1 friend class OsciActiveObject** [friend]

**7.260.3.2 friend class OsciCoeActiveScheduler** [friend]

**7.260.3.3 friend class OsciCoeActiveSchedulerBase** [friend]

**7.260.3.4 friend class OsciExecScheduler** [friend]

**7.260.3.5 friend class OsciExecSchedulerBase** [friend]

**7.260.3.6 friend class OsciExecSchedulerCommonBase** [friend]

**7.260.3.7 friend class OsciTimerObject** [friend]

**7.260.3.8 friend class PVActiveBase** [friend]

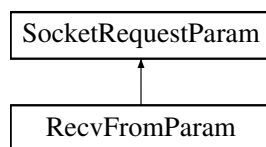
The documentation for this class was generated from the following file:

- [osci\\_scheduler\\_threadcontext.h](#)

## 7.261 RecvFromParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for RecvFromParam::



### Public Methods

- [RecvFromParam](#) (uint8 \**&aPtr*, uint32 *aMaxLen*, [OscNetworkAddress](#) &*aAddress*, uint32 *flags*, uint32 *aMultiMax*, [Osc\\_Vector](#)< uint32, [OscMemAllocator](#) > \**aPacketLen*, [Osc\\_Vector](#)< [OscNetworkAddress](#), [OscMemAllocator](#) > \**aPacketSource*)

### Data Fields

- [PVSockBufRecv](#) *iBufRecv*
- uint32 *iFlags*
- [OscNetworkAddress](#) & *iAddr*
- uint32 *iMultiMaxLen*
- [Osc\\_Vector](#)< uint32, [OscMemAllocator](#) > \* *iPacketLen*
- [Osc\\_Vector](#)< [OscNetworkAddress](#), [OscMemAllocator](#) > \* *iPacketSource*

### 7.261.1 Constructor & Destructor Documentation

- 7.261.1.1 [RecvFromParam::RecvFromParam](#) (uint8 \**&aPtr*, uint32 *aMaxLen*, [OscNetworkAddress](#) & *aAddress*, uint32 *flags*, uint32 *aMultiMax*, [Osc\\_Vector](#)< uint32, [OscMemAllocator](#) > \* *aPacketLen*, [Osc\\_Vector](#)< [OscNetworkAddress](#), [OscMemAllocator](#) > \* *aPacketSource*) [inline]

### 7.261.2 Field Documentation

- 7.261.2.1 [OscNetworkAddress](#)& [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 [Osc\\_Vector](#)<uint32, [OscMemAllocator](#)>\* [RecvFromParam::iPacketLen](#)

- 7.261.2.6 [Osc\\_Vector](#)<[OscNetworkAddress](#), [OscMemAllocator](#)>\* [RecvFromParam::iPacketSource](#)

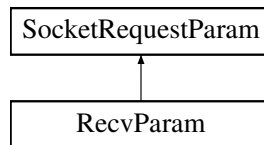
The documentation for this class was generated from the following file:

- [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

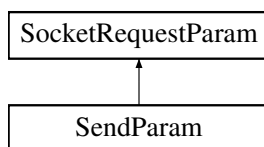
The documentation for this class was generated from the following file:

- [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

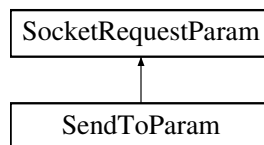
The documentation for this class was generated from the following file:

- [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, [OscNetworkAddress](#) &anAddr, uint32 flags)
- [~SendToParam](#) ()

### Data Fields

- [PVSockBufSend](#) iBufSend
- uint32 iFlags
- [OscNetworkAddress](#) iAddr
- uint32 iXferLen

### 7.264.1 Constructor & Destructor Documentation

**7.264.1.1** [SendToParam::SendToParam](#) (const uint8 \*& aPtr, uint32 aLen, [OscNetworkAddress](#) & anAddr, uint32 flags) [inline]

**7.264.1.2** [SendToParam::~~SendToParam](#) () [inline]

### 7.264.2 Field Documentation

**7.264.2.1** [OscNetworkAddress](#) [SendToParam::iAddr](#)

**7.264.2.2** [PVSockBufSend](#) [SendToParam::iBufSend](#)

**7.264.2.3** uint32 [SendToParam::iFlags](#)

**7.264.2.4** uint32 [SendToParam::iXferLen](#)

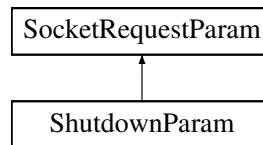
The documentation for this class was generated from the following file:

- [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](#)

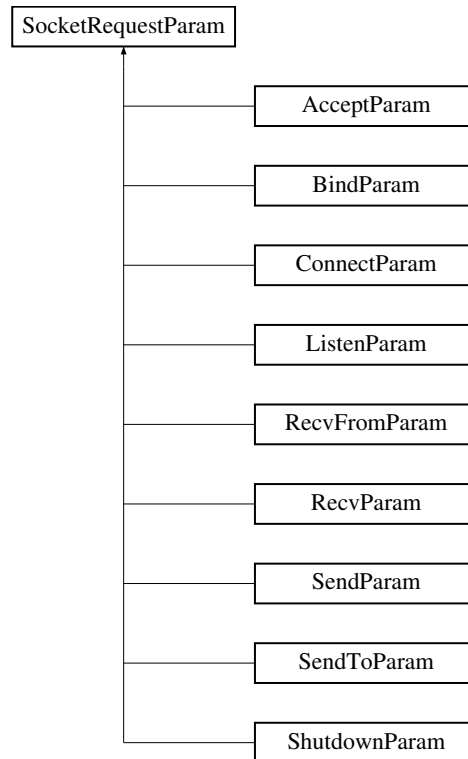
The documentation for this class was generated from the following file:

- [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:

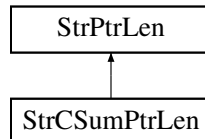
- [oscl\\_socket\\_request.h](#)

## 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:

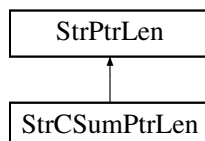
- [oscl\\_str\\_ptr\\_len.h](#)

## 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 [OscBasicTimeStruct](#) &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](#) ([OscBasicDateTimeStruct](#) in\_ts)  
*Create a TimeValue representing the absolute time specified by the BasicDateTimeStruct.*
- OSCL\_COND\_IMPORT\_REF int32 [get\\_local\\_time](#) ()  
*Get the local time after having adjusted for daylight saving.*
- OSCL\_COND\_IMPORT\_REF void [set\\_to\\_zero](#) ()  
*Set the time value to zero (represents a zero interval).*
- OSCL\_COND\_IMPORT\_REF void [set\\_to\\_current\\_time](#) ()  
*Set the time value to the current system time.*
- OSCL\_COND\_IMPORT\_REF void [set\\_from\\_ntp\\_time](#) (const uint32 ntp\_offset)  
*This method coverts a 32-bit NTP offset to system time.*
- OSCL\_COND\_IMPORT\_REF uint32 [get\\_sec](#) () const  
*Get a 32 bit value representing the seconds since the (system dependent) epoch.*
- OSCL\_COND\_IMPORT\_REF int32 [to\\_msec](#) () const
- OSCL\_COND\_IMPORT\_REF uint32 [get\\_usec](#) () const  
*Get a 32 bit value representing the number of microseconds in the time value.*
- OSCL\_COND\_IMPORT\_REF uint64 [get\\_timevalue\\_in\\_usec](#) () const  
*Get a 64 bit value representing the time value converted to microseconds.*
- OSCL\_IMPORT\_REF char \* [get\\_str\\_ctime](#) ([CtimeStrBuf](#) ctime\_strbuf)  
*Get a string containing a text representation of this TimeValue object.*

- OSCL\_IMPORT\_REF int [get\\_pv8601\\_str\\_time](#) (PV8601timeStrBuf time\_strbuf)  
*Get a PV extended text representation of the Time based on the PV 8601 format.*
- OSCL\_IMPORT\_REF int [get\\_ISO8601\\_str\\_time](#) (ISO8601timeStrBuf time\_strbuf)  
*Get a PV extended text representation of the Time based on the ISO 8601 format.*
- OSCL\_IMPORT\_REF char \* [get\\_rfc822\\_gmtime\\_str](#) (int max\_time\_strlen, char \*time\_str)  
*Get a text representation of the time in the GMT timezone based on the RFC 822 / RFC 1123 (also described in the HTTP spec RFC 2068 and RFC 2616).*
- OSCL\_COND\_IMPORT\_REF bool [is\\_zero](#) ()  
*Determine if the time value is zero.*
- OSCL\_COND\_IMPORT\_REF bool [is\\_zulu](#) () const  
*Manipulate internal flags to mark the time value as being in "zulu" (GMT) time.*
- OSCL\_COND\_IMPORT\_REF void [set\\_zulu](#) (bool is\_zulu)
- OSCL\_COND\_IMPORT\_REF TimeValue & [operator=](#) (const TimeValue &a)  
*Assignment operator.*
- OSCL\_COND\_IMPORT\_REF TimeValue & [operator+=](#) (const TimeValue &a)  
*+= operator*
- OSCL\_COND\_IMPORT\_REF TimeValue & [operator-=](#) (const TimeValue &a)  
*-= operator*
- OSCL\_COND\_IMPORT\_REF TimeValue & [operator\\*=](#) (const int scale)  
*This operator scales the time value by a constant.*
- OSCL\_COND\_IMPORT\_REF [OsciBasicTimeStruct](#) \* [get\\_timeval\\_ptr](#) ()
- OSCL\_COND\_IMPORT\_REF TimeValue & [operator+=](#) (const int32 aSeconds)
- OSCL\_COND\_IMPORT\_REF TimeValue & [operator-=](#) (const int32 aSeconds)

## Friends

- class [NTPTIME](#)
- OSCL\_COND\_IMPORT\_REF friend bool [operator==](#) (const TimeValue &a, const TimeValue &b)
- OSCL\_COND\_IMPORT\_REF friend bool [operator!=](#) (const TimeValue &a, const TimeValue &b)
- OSCL\_COND\_IMPORT\_REF friend bool [operator<=](#) (const TimeValue &a, const TimeValue &b)
- OSCL\_COND\_IMPORT\_REF friend bool [operator>=](#) (const TimeValue &a, const TimeValue &b)
- OSCL\_COND\_IMPORT\_REF friend bool [operator<](#) (const TimeValue &a, const TimeValue &b)
- OSCL\_COND\_IMPORT\_REF friend bool [operator>](#) (const TimeValue &a, const TimeValue &b)



## 7.269.1 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 OsciBasicTimeStruct & in\_tv)

Create a TimeValue representing the absolute time specified by the BasicTimeStruct.

#### Parameters:

- in\_tv* OsciBasicTimeStruct 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 (OsciBasicDateTimeStruct *in\_ts*)**

Create a TimeValue representing the absolute time specified by the BasicDateTimeStruct.

**Parameters:**

*in\_ts* OsciBasicDateTimeStruct 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 nubmer 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 OsciBasicTimeStruct\* 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 coverts 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](#) ([TOscITlsKey](#) \*key, [OscAny](#) \*ptr, int32 &)
- OSCL\_IMPORT\_REF [OscAny](#) \* [get\\_registry](#) ([TOscITlsKey](#) \*key)

### 7.270.1 Member Function Documentation

**7.270.1.1** OSCL\_IMPORT\_REF [OscAny](#)\* [TLSStorageOps::get\\_registry](#) ([TOscITlsKey](#) \* *key*)  
[static]

**7.270.1.2** OSCL\_IMPORT\_REF void [TLSStorageOps::save\\_registry](#) ([TOscITlsKey](#) \* *key*,  
[OscAny](#) \* *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 [OscAOStatus](#)

#### 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 [\\_OscBasicAllocator](#)

### 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 [OscBinIStream](#)
- class [OscBinIStreamBigEndian](#)
- class [OscBinIStreamLittleEndian](#)
- class [OscBinOStream](#)

*Class OscBinOStream implements the basic stream functions for an output stream.*

- class [OscBinOStreamBigEndian](#)

*Class OscBinOStreamBigEndian implements a binary output stream using big endian byte ordering.*

- class [OscBinOStreamLittleEndian](#)

*Class OscBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.*

- class [OscBinStream](#)

### 8.6.1 Detailed Description

Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.

## 8.7 oscl\_byte\_order.h File Reference

This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).

```
#include "oscl_base.h"
#include "oscl_byte_order.inl"
```

### Functions

- void [little\\_endian\\_to\\_host](#) (char \*data, uint32 size)  
*Convert little endian to host format.*
- void [host\\_to\\_little\\_endian](#) (char \*data, unsigned int size)  
*Convert host to little endian format.*
- void [big\\_endian\\_to\\_host](#) (char \*data, unsigned int size)  
*Convert big endian to host format.*
- void [host\\_to\\_big\\_endian](#) (char \*data, unsigned int size)  
*Convert host to big endian format.*

### 8.7.1 Detailed Description

This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).



## 8.8 oscl\_defalloc.h File Reference

The file defines simple default memory allocator classes. These allocators are used by the [OscVector](#) and [OscMap](#) class, etc.

```
#include "oscl_base.h"
#include "osclconfig_compiler_warnings.h"
#include "oscl_mem_inst.h"
```

### Data Structures

- class [OscAlloc](#)
- class [OscDealloc](#)
- class [OscDefAlloc](#)
- class [OscTAlloc](#)
- class [OscAllocDestructDealloc](#)
- class [OscDestructDealloc](#)
- 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 [OscVector](#) and [OscMap](#) class, etc.

## 8.9 oscl\_dll.h File Reference

Defines a DLL entry point.

```
#include "osclconfig.h"
```

### Defines

- #define [OSCL\\_DLL\\_ENTRY\\_POINT](#)() void oscl\_dll\_entry\_point() { }
- #define [OSCL\\_DLL\\_ENTRY\\_POINT\\_DEFAULT](#)()

### 8.9.1 Detailed Description

Defines a DLL entry point.

## 8.10 oscl\_dns.h File Reference

The file [oscl\\_socket.h](#) defines the OSCL DNS APIs.

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_defalloc.h"
#include "oscl_socket.h"
```

### Data Structures

- class [OscIDNS](#)
- class [OscIDNSObserver](#)

### Enumerations

- enum [TPVDNSFxn](#) { [EPVDNSGetHostByName](#) }
- enum [TPVDNSEvent](#) { [EPVDNSSuccess](#), [EPVDNSPending](#), [EPVDNSTimeout](#), [EPVDNSFailure](#), [EPVDNSCancel](#) }

### 8.10.1 Detailed Description

The file [oscl\\_socket.h](#) defines the OSCL DNS APIs.

## 8.11 oscl\_dns\_gethostbyname.h File Reference

```
#include "oscl_dns_method.h"
#include "oscl_dns.h"
#include "osclconfig_io.h"
```

### Data Structures

- class [OscGetHostByNameMethod](#)
- class [OscGetHostByNameRequest](#)

## 8.12 oscl\_dns\_imp.h File Reference

```
#include "oscl_dns_tuneables.h"  
#include "oscl_dns_imp_pv.h"
```

## 8.13 oscl\_dns\_imp\_base.h File Reference

```
#include "oscl_socket_imp.h"
#include "oscl_dns_request.h"
#include "oscl_dns.h"
```

### Data Structures

- class [OscIDNSIBase](#)

## 8.14 oscl\_dns\_imp\_pv.h File Reference

```
#include "oscl_socket_imp_base.h"  
#include "oscl_dns_request.h"  
#include "oscl_dns_imp_base.h"
```

### Data Structures

- class [OscIDNSI](#)

## 8.15 oscl\_dns\_method.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_dns.h"
#include "pvlogger.h"
```

### Data Structures

- class [OscDNSMethod](#)
- class [OscDNSRequestAO](#)



## 8.16 oscl\_dns\_param.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_dns_tuneables.h"
#include "oscl_namestring.h"
#include "oscl_dns.h"
#include "oscl_mutex.h"
#include "oscl_semaphore.h"
```

### Data Structures

- class [DNSRequestParam](#)
- class [GetHostByNameParam](#)

### Typedefs

- typedef [OscMemAllocator](#) [TDNSRequestParamAllocator](#)

#### 8.16.1 Typedef Documentation

##### 8.16.1.1 typedef [OscMemAllocator](#) [TDNSRequestParamAllocator](#)

## 8.17 oscl\_dns\_request.h File Reference

```
#include "oscl_dns_tuneables.h"
#include "oscl_namestring.h"
#include "oscl_dns.h"
#include "oscl_socket_types.h"
```

### Data Structures

- class [OscDNSRequest](#)

## 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 [OscDoubleLink](#)
- class [OscDoubleList](#)
- class [OscDoubleListBase](#)
- class [OscDoubleRunner](#)
- class [OscPriorityLink](#)
- class [OscPriorityList](#)

### Defines

- #define [QUE\\_ITER\\_BEGIN](#)(\_type, \_qname)
- #define [QUE\\_ITER\\_END](#)(\_qname)

### Functions

- template<class T, class S> T \* [OscPtrAdd](#) (T \*aPtr, S aVal)
- template<class T, class S> T \* [OscPtrSub](#) (T \*aPtr, S aVal)

### 8.19.1 Detailed Description

Internal use types for scheduler.

## 8.20 oscl\_errno.h File Reference

Defines functions to access additional information on errors where supported through an errno or similar service.

```
#include "oscl_base.h"
#include "osclconfig_error.h"
#include "oscl_errno.inl"
```

### Functions

- OSCL\_IMPORT\_REF bool [OSCL\\_IsErrnoSupported](#) ()  
*This function determines if a particular system saves the error number that occurs on a system call.*
- OSCL\_IMPORT\_REF int [OSCL\\_GetLastError](#) ()  
*This function returns the value of the system's global error number variable.*
- OSCL\_IMPORT\_REF bool [OSCL\\_SetLastError](#) (int newVal)  
*This function sets the last error code for the system.*
- OSCL\_IMPORT\_REF char \* [OSCL\\_StrError](#) (int errnum)  
*This function maps an error number to an error-message string.*

### 8.20.1 Detailed Description

Defines functions to access additional information on errors where supported through an errno or similar service.

## 8.21 oscl\_error.h File Reference

OSCL Error trap and cleanup include file.

```
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_error_codes.h"
#include "oscl_singleton.h"
#include "oscl_assert.h"
#include "oscl_tls.h"
```

### Data Structures

- class [OscError](#)
- class [OscErrorTrap](#)
- class [OscTLSEx](#)
- class [OscTLSRegistryEx](#)

### Defines

- #define [OSCL\\_TRAPSTACK\\_PUSH\(a\)](#) [OscError::PushL\(a\)](#)
- #define [OSCL\\_TRAPSTACK\\_POP\(\)](#) [OscError::Pop\(\)](#)
- #define [OSCL\\_TRAPSTACK\\_POPDEALLOC\(\)](#) [OscError::PopDealloc\(\)](#)

### 8.21.1 Detailed Description

OSCL Error trap and cleanup include file.

## 8.22 oscl\_error\_allocator.h File Reference

Defines a memory allocation class used by the oscl error layer.

```
#include "oscl_base.h"
#include "oscl_base_macros.h"
#include "osclconfig_error.h"
#include "oscl_assert.h"
#include "oscl_defalloc.h"
```

### Data Structures

- class [OscErrorAllocator](#)

*This class provides static methods to invoke the user defined memory allocation routines.*

### 8.22.1 Detailed Description

Defines a memory allocation class used by the oscl error layer.

## 8.23 oscl\_error\_codes.h File Reference

Defines basic error and leave codes.

### Defines

- #define [OscErrNone](#) 0
- #define [OscErrGeneral](#) 100
- #define [OscErrNoMemory](#) 101
- #define [OscErrCancelled](#) 102
- #define [OscErrNotSupported](#) 103
- #define [OscErrArgument](#) 104
- #define [OscErrBadHandle](#) 105
- #define [OscErrAlreadyExists](#) 106
- #define [OscErrBusy](#) 107
- #define [OscErrNotReady](#) 108
- #define [OscErrCorrupt](#) 109
- #define [OscErrTimeout](#) 110
- #define [OscErrOverflow](#) 111
- #define [OscErrUnderflow](#) 112
- #define [OscErrInvalidState](#) 113
- #define [OscErrNoResources](#) 114
- #define [OscErrNotInstalled](#) 115
- #define [OscErrAlreadyInstalled](#) 116
- #define [OscErrSystemCallFailed](#) 117
- #define [OscErrNoHandler](#) 118
- #define [OscErrThreadContextIncorrect](#) 119
- #define [OSCL\\_ERR\\_NONE](#) [OscErrNone](#)
- #define [OSCL\\_BAD\\_ALLOC\\_EXCEPTION\\_CODE](#) [OscErrNoMemory](#)
- #define [OscSuccess](#) 0
- #define [OscPending](#) 1
- #define [OscFailure](#) -1

### Typedefs

- typedef int32 [OscLeaveCode](#)
- typedef int32 [OscReturnCode](#)

#### 8.23.1 Detailed Description

Defines basic error and leave codes.



## 8.24 oscl\_error\_imp.h File Reference

Internal error implementation support.

```
#include "osclconfig_error.h"
```

```
#include "oscl_error_imp_jumps.h"
```

### Defines

- #define [PERROR\\_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 [PVErrors\\_DoLeave\(\)](#) [internalLeave](#) \_\_ilv;\_\_ilv.a=0;throw(\_\_ilv)
- #define [\\_PV\\_TRAP](#)(\_\_r, \_\_s)
- #define [\\_PV\\_TRAP\\_NO\\_TLS](#)(\_\_trapimp, \_\_r, \_\_s)

### 8.25.1 Detailed Description

Implementation File for Leave using C++ exceptions.

## 8.26 oscl\_error\_imp\_fatalerror.h File Reference

Implementation File for Leave using system fatal error.

```
#include "oscl_assert.h"
```

### Defines

- #define [PVErrDoLeave](#)() \_OSCL\_Abort()
- #define [\\_PV\\_TRAP](#)(\_\_r, \_\_s)
- #define [\\_PV\\_TRAP\\_NO\\_TLS](#)(\_\_tr, \_\_r, \_\_s)

### 8.26.1 Detailed Description

Implementation File for Leave using system fatal error.

### 8.26.2 Define Documentation

#### 8.26.2.1 #define [\\_PV\\_TRAP](#)(\_\_r, \_\_s)

**Value:**

```
__r=OscErrNone;\n{__s;}
```

#### 8.26.2.2 #define [\\_PV\\_TRAP\\_NO\\_TLS](#)(\_\_tr, \_\_r, \_\_s)

**Value:**

```
__r=OscErrNone;\n{__s;}
```

#### 8.26.2.3 #define [PVErrDoLeave](#)() [\\_OSCL\\_Abort](#)()

## 8.27 oscl\_error\_imp\_jumps.h File Reference

Implementation of using Setjmp / Longjmp.

```
#include "oscl_error_trapcleanup.h"
#include "oscl_assert.h"
#include "osclconfig_error.h"
#include "oscl_defalloc.h"
#include "oscl_error.h"
```

### Data Structures

- class [OscJump](#)

### Defines

- #define [OSCL\\_JUMP\\_MAX\\_JUMP\\_MARKS](#) OSCL\_MAX\_TRAP\_LEVELS
- #define [internalLeave](#) (-1)
- #define [PVErrDoLeave](#)() OscJump::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=OscErrNone;\
{\
    OscErrorTrapImp* __trap=OscErrorTrapImp::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=OscErrNone;\
{\
    OscErrorTrapImp* __trap=OscErrorTrapImp::TrapNoTls(__trapimp);\
    if(!__trap){__s;}else{\
        int __tr=setjmp(*(__trap->iJumpData->Top()));\
        if (__tr==0)\
            {__s;}\
        else if (__tr==internalLeave)\
            {__r=__trap->iLeave;}\
            __trap->UnTrap();}\
    }
```

### 8.27.2.3 #define PVErrDoLeave() OscJump::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 [OscTrapStack](#)
- class [OscTrapStackItem](#)

### Defines

- #define [OSCL\\_MAX\\_TRAP\\_LEVELS](#) 20
- #define [PERRORTRAP\\_REGISTRY\\_ID](#) [OSCL\\_TLS\\_ID\\_PERRORTRAP](#)
- #define [PERRORTRAP\\_REGISTRY](#) [OscTLSRegistry](#)

### 8.28.1 Detailed Description

OSCL Error trap and cleanup implementation include file.

## 8.29 oscl\_exception.h File Reference

contains all the exception handling macros and classes

```
#include "oscl_error.h"
#include "oscl_error_imp.h"
```

### Data Structures

- class [OsclException](#)

*oscl\_exception.h contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from*

### Defines

- #define [OSCL\\_LEAVE](#)(\_leave\_status) OsclError::Leave(\_leave\_status)  
*Use this macro to cause a Leave. It terminates the execution of the current active function.*
- #define [OSCL\\_TRY](#)(\_leave\_status, \_statements) \_PV\_TRAP(\_leave\_status, \_statements)  
*This macro will be used to set up a try block.*
- #define [OSCL\\_TRY\\_NO\\_TLS](#)(\_\_trapimp, \_leave\_status, \_statements) \_PV\_TRAP\_NO\_TLS(\_\_trapimp, \_leave\_status, \_statements)
- #define [OSCL\\_FIRST\\_CATCH\\_ANY](#)(\_leave\_status, \_statements) if (\_leave\_status!=OsclErrNone) { \_statements; }  
*This section defines the macros to be used in the catch block following the try block Use this macro to call a function that handles all exception types thrown in the preceding try block.*
- #define [OSCL\\_FIRST\\_CATCH](#)(\_leave\_status, \_catch\_value, \_statements) if (\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){ \_statements; }  
*Use this macro to define a block of code that catches the first exception type thrown in the preceding try block.*
- #define [OSCL\\_CATCH](#)(\_leave\_status, \_catch\_value, \_statements) else if (\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){ \_statements; }  
*Use this macro to define a block of code for catching additional exception types.*
- #define [OSCL\\_CATCH\\_ANY](#)(\_leave\_status, \_statements) else if (\_leave\_status!=OsclErrNone){ \_statements; }  
*Use this macro to call a function that will catch all remaining exception types.*
- #define [OSCL\\_LAST\\_CATCH](#)(\_leave\_status) else if (\_leave\_status!=OsclErrNone){ OSCL\_LEAVE(\_leave\_status); }  
*Use this macro if OSCL\_CATCH\_ANY has not been used. It will mark the end of the catch block.*

### 8.29.1 Detailed Description

contains all the exception handling macros and classes

## 8.30 oscl\_exclusive\_ptr.h File Reference

This file defines the [OscExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

```
#include "oscl_defalloc.h"
```

### Data Structures

- class [OscExclusiveArrayPtr](#)

*The OscExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OscExclusiveArrayPtr expires, its destructor uses delete to free the memory.*

- class [OscExclusivePtr](#)

*The OscExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OscExclusivePtr expires, its destructor uses delete to free the memory.*

- class [OscExclusivePtrA](#)

*The OscExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or indirectly) through Alloc. When the OscExclusivePtrA expires, Alloc is used to free the memory.*

### 8.30.1 Detailed Description

This file defines the [OscExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.



## 8.31 oscl\_file\_async\_read.h File Reference

```
#include "oscl_base.h"
#include "osclconfig_io.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_scheduler_ao.h"
#include "oscl_file_io.h"
#include "oscl_semaphore.h"
```

### Data Structures

- class [OscAsyncFile](#)
- class [OscAsyncFileBuffer](#)
- class [OscBuf](#)
- class [OscPtr](#)
- class [OscPtrC](#)

## 8.32 oscl\_file\_cache.h File Reference

The file [oscl\\_file\\_cache.h](#) defines the class [OscFileCache](#).

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_file_io.h"
```

### Data Structures

- class [OscFileCache](#)
- class [OscFileCacheBuffer](#)

### 8.32.1 Detailed Description

The file [oscl\\_file\\_cache.h](#) defines the class [OscFileCache](#).

## 8.33 oscl\_file\_dir\_utils.h File Reference

The file [oscl\\_file\\_dir\\_utils.h](#) defines some unix-style directory ops.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
```

### Data Structures

- struct [oscl\\_fsstat](#)
- struct [oscl\\_stat\\_buf](#)

### Typedefs

- typedef [oscl\\_fsstat](#) OSCL\_FSSTAT
- typedef [oscl\\_stat\\_buf](#) OSCL\_STAT\_BUF

### Enumerations

- enum [OSCL\\_FILEMGMT\\_PERMS](#) { [OSCL\\_FILEMGMT\\_PERMS\\_READ](#) = 0x1, [OSCL\\_FILEMGMT\\_PERMS\\_WRITE](#) = 0x2, [OSCL\\_FILEMGMT\\_PERMS\\_EXECUTE](#) = 0x4 }
- enum [OSCL\\_FILEMGMT\\_MODES](#) { [OSCL\\_FILEMGMT\\_MODE\\_DIR](#) = 0x1 }
- enum [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) { [OSCL\\_FILEMGMT\\_E\\_OK](#) = 0, [OSCL\\_FILEMGMT\\_E\\_PATH\\_TOO\\_LONG](#), [OSCL\\_FILEMGMT\\_E\\_PATH\\_NOT\\_FOUND](#), [OSCL\\_FILEMGMT\\_E\\_ALREADY\\_EXISTS](#), [OSCL\\_FILEMGMT\\_E\\_NOT\\_EMPTY](#), [OSCL\\_FILEMGMT\\_E\\_PERMISSION\\_DENIED](#), [OSCL\\_FILEMGMT\\_E\\_NO\\_MATCH](#), [OSCL\\_FILEMGMT\\_E\\_UNKNOWN](#), [OSCL\\_FILEMGMT\\_E\\_SYS\\_SPECIFIC](#), [OSCL\\_FILEMGMT\\_E\\_NOT\\_IMPLEMENTED](#) }

### Functions

- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_getcwd](#) ([oscl\\_wchar](#) \*path, uint32 size)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_getcwd](#) (char \*path, uint32 size)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_stat](#) (const [oscl\\_wchar](#) \*path, [OSCL\\_STAT\\_BUF](#) \*statbuf)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_stat](#) (const char \*path, [OSCL\\_STAT\\_BUF](#) \*statbuf)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_mkdir](#) (const [oscl\\_wchar](#) \*path)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_mkdir](#) (const char \*path)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_rmdir](#) (const [oscl\\_wchar](#) \*path)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_rmdir](#) (const char \*path)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_chdir](#) (const [oscl\\_wchar](#) \*path)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_chdir](#) (const char \*path)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_rename](#) (const [oscl\\_wchar](#) \*oldpath, const [oscl\\_wchar](#) \*newpath)
- OSCL\_IMPORT\_REF [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) [oscl\\_rename](#) (const char \*oldpath, const char \*newpath)

- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_statfs (OSCL\_FSSTAT \*stats, const char \*path)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_statfs (OSCL\_FSSTAT \*stats, const oscl\_wchar \*path)

### 8.33.1 Detailed Description

The file [oscl\\_file\\_dir\\_utils.h](#) defines some unix-style directory ops.

## 8.34 oscl\_file\_find.h File Reference

The file [oscl\\_file\\_find.h](#) defines the class [OscL\\_FileFind](#).

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_mem.h"
#include "oscl_vector.h"
#include "oscl_string_containers.h"
#include "oscl_file_types.h"
```

### Data Structures

- class [OscL\\_FileFind](#)

#### 8.34.1 Detailed Description

The file [oscl\\_file\\_find.h](#) defines the class [OscL\\_FileFind](#).

## 8.35 oscl\_file\_handle.h File Reference

The file [oscl\\_file\\_handle.h](#) defines the class [OscFileHandle](#).

```
#include "osclconfig_io.h"
#include "oscl_base.h"
```

### Data Structures

- class [OscFileHandle](#)

### Typedefs

- typedef FILE \* [TOscFileHandle](#)

#### 8.35.1 Detailed Description

The file [oscl\\_file\\_handle.h](#) defines the class [OscFileHandle](#).

## 8.36 oscl\_file\_io.h File Reference

The file [oscl\\_file\\_io.h](#) defines the class [OscFile](#). This is the public API to the basic file I/O operations.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_mem.h"
#include "oscl_vector.h"
#include "oscl_file_server.h"
#include "oscl_file_find.h"
#include "oscl_file_dir_utils.h"
#include "oscl_file_handle.h"
```

### Data Structures

- class [OscFile](#)
- class [OscFixedCacheParam](#)
- class [OscCacheObserver](#)

### Defines

- #define [TOscFileOffsetInt32](#) int32

### 8.36.1 Detailed Description

The file [oscl\\_file\\_io.h](#) defines the class [OscFile](#). This is the public API to the basic file I/O operations.

## 8.37 oscl\_file\_manager.h File Reference

File management class.

```
#include "oscl_base.h"
```

### Data Structures

- class [OscFileManager](#)

#### 8.37.1 Detailed Description

File management class.



## 8.38 oscl\_file\_native.h File Reference

The file [oscl\\_file\\_native.h](#) defines the class [OscNativeFile](#). This is the porting layer for basic file I/O operations.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_aostatus.h"
#include "oscl_file_io.h"
```

### Data Structures

- class [OscNativeFile](#)

#### 8.38.1 Detailed Description

The file [oscl\\_file\\_native.h](#) defines the class [OscNativeFile](#). This is the porting layer for basic file I/O operations.

## 8.39 oscl\_file\_server.h File Reference

The file [oscl\\_file\\_server.h](#) defines the class [OscL\\_FileServer](#). This is the porting layer for file server implementations.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
```

### Data Structures

- class [OscL\\_FileServer](#)

#### 8.39.1 Detailed Description

The file [oscl\\_file\\_server.h](#) defines the class [OscL\\_FileServer](#). This is the porting layer for file server implementations.

## 8.40 oscl\_file\_stats.h File Reference

File stats class.

```
#include "oscl_base.h"
#include "osclconfig_io.h"
```

### Data Structures

- class [OscFileStats](#)
- class [OscFileStatsItem](#)

### Defines

- #define [OSCL\\_FILE\\_STATS\\_LOGGER\\_NODE](#) "OscFileStats"

### Enumerations

- enum [TOscFileOp](#) { [EOscFileOp\\_Open](#), [EOscFileOp\\_Close](#), [EOscFileOp\\_Read](#), [EOscFileOp\\_Write](#), [EOscFileOp\\_Seek](#), [EOscFileOp\\_Tell](#), [EOscFileOp\\_Size](#), [EOscFileOp\\_Flush](#), [EOscFileOp\\_EndOfFile](#), [EOscFileOp\\_SetSize](#), [EOscFileOp\\_NativeOpen](#), [EOscFileOp\\_NativeClose](#), [EOscFileOp\\_NativeRead](#), [EOscFileOp\\_NativeWrite](#), [EOscFileOp\\_NativeSeek](#), [EOscFileOp\\_NativeTell](#), [EOscFileOp\\_NativeSize](#), [EOscFileOp\\_NativeFlush](#), [EOscFileOp\\_NativeEndOfFile](#), [EOscFileOp\\_NativeSetSize](#), [EOscFileOp\\_Last](#) }

#### 8.40.1 Detailed Description

File stats class.

## 8.41 oscl\_file\_types.h File Reference

The file [oscl\\_file\\_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.

### Data Structures

- class [OscNativeFileParams](#)

### Defines

- #define [OSCL\\_IO\\_FILENAME\\_MAXLEN](#) 512
- #define [OSCL\\_IO\\_EXTENSION\\_MAXLEN](#) 512
- #define [OSCL\\_FILE\\_WCHAR\\_PATH\\_DELIMITER](#) \_STRLIT("/")
- #define [OSCL\\_FILE\\_CHAR\\_PATH\\_DELIMITER](#) \_STRLIT\_CHAR("/")

#### 8.41.1 Detailed Description

The file [oscl\\_file\\_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.

## 8.42 oscl\_heapbase.h File Reference

OSCL Heap Base include file.

```
#include "osclconfig_error.h"
#include "oscl_base.h"
#include "oscl_heapbase.inl"
```

### Data Structures

- class [\\_OscHeapBase](#)
- class [OscTrapItem](#)

### Typedefs

- typedef void(\* [OscTrapOperation](#) )(OscAny \*)

#### 8.42.1 Detailed Description

OSCL Heap Base include file.

## 8.43 oscl\_init.h File Reference

Global oscl initialization.

```
#include "oscl_base.h"
#include <stdio.h>
```

### Data Structures

- class [OscInit](#)
- class [OscSelect](#)

### 8.43.1 Detailed Description

Global oscl initialization.

## 8.44 oscl\_int64\_utils.h File Reference

```
#include "oscl_base.h"
```

### Data Structures

- class [Osc\\_Int64\\_Utils](#)  
*The Osc\_Int64\_Utils class provides a wrapper for commonly used int64/uint64 operations.*
- struct [OscInteger64Transport](#)

### Typedefs

- typedef [OscInteger64Transport](#) [\\_OscInteger64Transport](#)

#### 8.44.1 Typedef Documentation

##### 8.44.1.1 typedef struct [OscInteger64Transport](#) [\\_OscInteger64Transport](#)

[OscInteger64Transport](#) Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

## 8.45 oscl\_ip\_socket.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OscIPSocketI](#)



## 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\\_MTLlinked\\_List](#)

### 8.46.1 Detailed Description

The file [oscl\\_linked\\_list.h](#) defines the template class [OscL\\_Linked\\_List](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

## 8.47 oscl\_lock\_base.h File Reference

This file defines an abstract lock class, [OscLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OscNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OscScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.

### Data Structures

- class [OscLockBase](#)
- class [OscNullLock](#)
- class [OscScopedLock](#)

*The [OscScopedLock](#) class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the [OscScopedLock](#) goes out of scope.*

### 8.47.1 Detailed Description

This file defines an abstract lock class, [OscLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OscNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OscScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.

## 8.48 oscl\_map.h File Reference

The file [oscl\\_map.h](#) defines the template class [OscL\\_Map](#) which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_tree.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- struct [OscL\\_Less](#)
- class [OscL\\_Map](#)
- struct [OscL\\_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 [BufFragGroup](#)
- class [MediaData](#)
- class [MemAllocator](#)

### Typedefs

- typedef void(\* [BufferFreeFuncPtr](#) )(void \*)
- typedef uint32 [MediaTimestamp](#)

### 8.50.1 Detailed Description

Defines a container class for media data made up of a collection of memory fragments.

## 8.51 oscl\_media\_status.h File Reference

Defines a status values for the [MediaData](#) containers.

### Data Structures

- class [BufFragStatusClass](#)
- class [MediaStatusClass](#)

### Variables

- const int32 [APPEND\\_MEDIA\\_AT\\_END](#) = -1

#### 8.51.1 Detailed Description

Defines a status values for the [MediaData](#) containers.

## 8.52 oscl\_mem.h File Reference

This file contains basic memory definitions for common use across platforms.

```
#include "osclconfig_memory.h"
#include "oscl_base.h"
#include "oscl_types.h"
#include "oscl_assert.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_lock_base.h"
#include "osclconfig_compiler_warnings.h"
#include "oscl_mem_inst.h"
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_refcounter.h"
#include "oscl_error.h"
#include "oscl_exception.h"
#include "oscl_mem.inl"
```

### Data Structures

- class [HeapBase](#)
- class [OscIAuditCB](#)
- class [OscIMem](#)
- class [OscIMemAllocator](#)
- class [OscIMemAllocDestructDealloc](#)
- class [OscIMemBasicAllocator](#)
- class [OscIMemBasicAllocDestructDealloc](#)
- class [OscIMemGlobalAuditObject](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [OSCL\\_HAS\\_GLOBAL\\_NEW\\_DELETE](#) 1
- #define [OSCL\\_CLEANUP\\_BASE\\_CLASS\(T\)](#) \_OSCL\_CLEANUP\_BASE\_CLASS(T)
- #define [OSCL\\_ALLOC\\_NEW\(T\\_allocator, T, params\)](#) new(T\_allocator.allocate(1)) T params
- #define [OSCL\\_TRAP\\_ALLOC\\_NEW\(T\\_ptr, T\\_allocator, T, params\)](#) \_OSCL\_TRAP\_NEW(T\_allocator.allocate(1), T\_allocator.deallocate, T\_ptr, T, params)
- #define [OSCL\\_ALLOC\\_DELETE\(ptr, T\\_allocator, T\)](#)
- #define [OSCL\\_MALLOC\(count\)](#) \_oscl\_default\_audit\_malloc(count)
- #define [oscl\\_malloc\(a\)](#) OSCL\_MALLOC(a)
- #define [OSCL\\_DEFAULT\\_MALLOC\(x\)](#) OSCL\_MALLOC(x)
- #define [OSCL\\_AUDIT\\_MALLOC\(auditCB, count\)](#) \_oscl\_audit\_malloc(count, auditCB)
- #define [OSCL\\_CALLOC\(num, size\)](#) \_oscl\_default\_audit\_calloc(num, size)
- #define [oscl\\_calloc\(a, b\)](#) OSCL\_CALLOC(a, b)

- #define [OSCL\\_AUDIT\\_CALLOC](#)(auditCB, num, size) \_oscl\_audit\_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 [OscMemAudit](#)
- class [OscMemStatsNode](#)

### 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, [Osc\\_TAlloc](#)< char, [OscMemBasicAllocator](#) > > [MMAudit-CharAutoPtr](#)
- typedef [OSCLMemAutoPtr](#)< uint8, [Osc\\_TAlloc](#)< uint8, [\\_OscBasicAllocator](#) > > [MMAudit-Uint8AutoPtr](#)
- typedef [OSCLMemAutoPtr](#)< [MM\\_AllocNode](#), [Osc\\_TAlloc](#)< [MM\\_AllocNode](#), [OscMemBasicAllocator](#) > > [MM\\_AllocNodeAutoPtr](#)
- typedef [OSCLMemAutoPtr](#)< [OscMemStatsNode](#), [Osc\\_TAlloc](#)< [OscMemStatsNode](#), [OscMemBasicAllocator](#) > > [MM\\_StatsNodeTagTreeType](#)

- typedef `OSCLMemAutoPtr< OsciMemStatsNode, Osci_TAlloc< OsciMemStatsNode, OsciMemBasicAllocator > > OsciMemStatsNodeAutoPtr`
- typedef `Osci_TAlloc< MM_StatsNodeTagTreeType, OsciMemBasicAllocator > TagTree_Allocator`
- typedef `Osci_TagTree< MM_StatsNodeTagTreeType, TagTree_Allocator > OsciTagTreeType`

### 8.53.1 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 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 [OscMemPoolFixedChunkAllocator](#)
- class [OscMemPoolFixedChunkAllocatorObserver](#)
- class [OscMemPoolResizableAllocator](#)
- class [OscMemPoolResizableAllocatorMemoryObserver](#)
- class [OscMemPoolResizableAllocatorObserver](#)

### 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 [OscMutex](#)
- class [OscThreadLock](#)

### Typedefs

- typedef [OscMutex](#) [OscNoYieldMutex](#)

#### 8.59.1 Detailed Description

This file provides implementation of mutex.

#### 8.59.2 Typedef Documentation

##### 8.59.2.1 typedef [OscMutex](#) [OscNoYieldMutex](#)

Class [OscNoYieldMutex](#) can be used in use cases where there will be no CPU-yielding operation done while the Mutex is locked.

CPU-yielding operations include [OscMutex::Lock](#), [OscSemaphore::Wait](#), [OscThread::Sleep](#), and [OscBrewThreadUtil::BThreadYield](#).

The behavior of [OscNoYieldMutex](#) depends on whether the threading model is pre-emptive or not. When threading is pre-emptive, it is identical to [OscMutex](#). When threading is non-pre-emptive, it is a NO-OP.

An example of this type of use case is for simple data protection.

## 8.60 oscl\_namestring.h File Reference

Name string class include file.

```
#include "oscl_base.h"
```

### Data Structures

- class [OscNameString](#)

#### 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 [OscCompareLess](#)
- class [OscPriorityQueue](#)
- class [OscPriorityQueueBase](#)

### 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 [OscProcStatus](#)

## 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 [OscRand](#)

### 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 manage its lifetime through the refcount.

```
#include "oscl_base.h"
#include "oscl_refcounter.h"
```

### Data Structures

- class [OscRefCountMemFrag](#)

#### 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 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 [OscRegistryAccessClient](#)

### 8.68.1 Detailed Description

Client-side implementation Registry Access implementation.

## 8.69 oscl\_registry\_client.h File Reference

Client-side implementation of OsciRegistry.

```
#include "oscl_registry_types.h"
#include "oscl_mem.h"
#include "oscl_string.h"
```

### Data Structures

- class [OsciRegistryClient](#)

### 8.69.1 Detailed Description

Client-side implementation of OsciRegistry.

## 8.70 oscl\_registry\_client\_impl.h File Reference

Client-side implementation of OsciRegistryInterface.

```
#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 [OsciRegistryAccessClientImpl](#)
- class [OsciRegistryAccessClientTlsImpl](#)
- class [OsciRegistryClientImpl](#)
- class [OsciRegistryClientTlsImpl](#)

### 8.70.1 Detailed Description

Client-side implementation of OsciRegistryInterface.

## 8.71 oscl\_registry\_serv\_impl.h File Reference

Server-side implementation of OsciRegistry 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 [OsciComponentRegistry](#)
- class [OsciComponentRegistryData](#)
- class [OsciComponentRegistryElement](#)

### 8.71.1 Detailed Description

Server-side implementation of OsciRegistry 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 [OscRegistryServTlsImpl](#)

## 8.74 oscl\_registry\_types.h File Reference

Common types used in Osl registry interfaces.

```
#include "oscl_types.h"
#include "oscl_string_containers.h"
```

### Data Structures

- class [OslRegistryAccessElement](#)

### Typedefs

- typedef [OslAny](#) \* [OslComponentFactory](#)

#### 8.74.1 Detailed Description

Common types used in Osl 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 [OscExecScheduler](#)
- class [OscExecSchedulerCommonBase](#)
- class [OscScheduler](#)
- class [OscSchedulerObserver](#)
- class [PVSchedulerStopper](#)

### Defines

- #define [PVSCHEDNAMELEN](#) 30

## 8.76 oscl\_scheduler\_ao.h File Reference

Osc Scheduler user execution object classes.

```
#include "oscl_scheduler_aobase.h"
#include "oscl_mem.h"
#include "oscl_scheduler_types.h"
```

### Data Structures

- class [OscActiveObject](#)
- class [OscTimerObject](#)

### 8.76.1 Detailed Description

Osc Scheduler user execution object classes.

## 8.77 oscl\_scheduler\_aobase.h File Reference

Osc Scheduler internal active object classes.

```
#include "oscl_namestring.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_scheduler_readyq.h"
#include "oscl_string_containers.h"
#include "oscl_scheduler_types.h"
```

### Data Structures

- class [PVActiveBase](#)
- class [PVActiveStats](#)

### Defines

- #define [OSCL\\_ZEROIZE](#)(ptr, size) oscl\_memset(ptr, 0, size)
- #define [PVEXECNAMELEN](#) 30

#### 8.77.1 Detailed Description

Osc 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 [OscReadyAlloc](#)
- class [OscReadyCompare](#)
- class [OscReadyQ](#)
- class [OscTimerCompare](#)
- class [OscTimerQ](#)
- class [TReadyQueLink](#)

### Typedefs

- typedef [PVActiveBase](#) \* [TOscReady](#)

### 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\\_OscIThread](#), [EPVThreadContext\\_NonOscIThread](#), [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 Osci 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 Osci 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 [OscExecSchedulerBase](#)

### 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 [OscSemaphore](#)

### 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 [OscSharedPtr](#) which is a "smart pointer" to the parameterized type.

```
#include "oscl_base.h"
#include "oscl_refcounter.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- class [OscSharedPtr](#)  
*A parameterized smart pointer class.*

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_RETURN\\_TYPE\\_NOT\\_UDT](#)

#### 8.83.1 Detailed Description

This file defines a template class [OscSharedPtr](#) which is a "smart pointer" to the parameterized type.

## 8.84 oscl\_singleton.h File Reference

This file defines the [OscSingleton](#) class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

### Data Structures

- class [OscSingleton](#)
- class [OscSingletonRegistry](#)
- class [SingletonTable](#)

### Variables

- const uint32 [OSCL\\_SINGLETON\\_ID\\_TEST](#) = 0
- const uint32 [OSCL\\_SINGLETON\\_ID\\_OSCLMEM](#) = 1
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVLOGGER](#) = 2
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVSCHEDULER](#) = 3
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PVERRORTRAP](#) = 4
- const uint32 [OSCL\\_SINGLETON\\_ID\\_SDPMEDIAPARSER](#) = 5
- const uint32 [OSCL\\_SINGLETON\\_ID\\_PAYLOADPARSER](#) = 6
- const uint32 [OSCL\\_SINGLETON\\_ID\\_CPM\\_PLUGIN](#) = 7
- const uint32 [OSCL\\_SINGLETON\\_ID\\_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 [OscSingleton](#) class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

[OscSingleton](#) is initialized in `OscBase::Init`.

## 8.84.2 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_PERRORTRAP = 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 OSL Socket APIs.

```
#include "osclconfig_io.h"
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_types.h"
```

### Data Structures

- class [OscSocketServ](#)
- class [OscTCPSocket](#)
- class [OscUDPSocket](#)

### 8.86.1 Detailed Description

The file [oscl\\_socket.h](#) defines the OSL 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 [OscAcceptMethod](#)
- class [OscAcceptRequest](#)

## 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 [OscBindMethod](#)
- class [OscBindRequest](#)

## 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 [OscConnectMethod](#)
- class [OscConnectRequest](#)



## 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 [OscSocketIBase](#)

## 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\\_SOCKET\\_NOT\\_OPEN](#) (-2)
- #define [PVSOCK\\_ERR\\_SOCKET\\_NO\\_SERV](#) (-3)
- #define [PVSOCK\\_ERR\\_SERV\\_NOT\\_CONNECTED](#) (-4)
- #define [PVSOCK\\_ERR\\_SOCKET\\_NOT\\_CONNECTED](#) (-5)
- #define [PVSOCK\\_ERR\\_NOT\\_IMPLEMENTED](#) (-6)
- #define [PVSOCK\\_ERR\\_NOT\\_SUPPORTED](#) (-7)

### 8.92.1 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\_SOCKET\_NO\_SERV (-3)

#### 8.92.1.6 #define PVSOCK\_ERR\_SOCKET\_NOT\_CONNECTED (-5)

#### 8.92.1.7 #define PVSOCK\_ERR\_SOCKET\_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 [OscListenMethod](#)
- class [OscListenRequest](#)

### 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 [OscSocketMethod](#)
- class [OscSocketRequestAO](#)

### 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 [OscRecvMethod](#)
- class [OscRecvRequest](#)

## 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 [OscRecvFromMethod](#)
- class [OscRecvFromRequest](#)

## 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 [OscSocketRequest](#)
- 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 [OscSendMethod](#)
- class [OscSendRequest](#)

## 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 [OscSendToMethod](#)
- class [OscSendToRequest](#)

## 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 [OscSocketServIBase](#)

## 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 [OscSocketServI](#)

### 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 [OscSocketServRequestList](#)
- class [OscSocketServRequestQElem](#)

## 8.104 oscl\_socket\_shutdown.h File Reference

```
#include "oscl_socket_types.h"  
#include "oscl_socket_method.h"
```

### Data Structures

- class [OscShutdownMethod](#)
- class [OscShutdownRequest](#)

## 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](#) { [EOsclSocket\\_RequestAO\\_Success](#), [EOsclSocket\\_RequestAO\\_Canceled](#), [EOsclSocket\\_RequestAO\\_Error](#), [EOsclSocket\\_RequestAO\\_Timeout](#), [EOsclSocket\\_ServRequestIssued](#), [EOsclSocket\\_ServPoll](#), [EOsclSocket\\_OS](#), [EOsclSocket\\_Readable](#), [EOsclSocket\\_Writable](#), [EOsclSocket\\_Except](#), [EOsclSocket\\_DataRecv](#), [EOsclSocket\\_DataSent](#), [EOsclSocket\\_ServRequestComplete](#), [EOsclSocket\\_ServRequestCancelIssued](#), [EOsclSocketServ\\_LoopsockOk](#), [EOsclSocketServ\\_LoopsockError](#) }
- enum [TOsclSocketServStatEvent](#) { [EOsclSocketServ\\_SelectNoActivity](#) = 0, [EOsclSocketServ\\_SelectActivity](#), [EOsclSocketServ\\_SelectRescheduleAsap](#), [EOsclSocketServ\\_SelectReschedulePoll](#), [EOsclSocketServ\\_LastEvent](#) }

### 8.105.1 Enumeration Type Documentation

#### 8.105.1.1 enum [TOsclSocketServStatEvent](#)

Enumeration values:

[EOsclSocketServ\\_SelectNoActivity](#)  
[EOsclSocketServ\\_SelectActivity](#)  
[EOsclSocketServ\\_SelectRescheduleAsap](#)  
[EOsclSocketServ\\_SelectReschedulePoll](#)  
[EOsclSocketServ\\_LastEvent](#)

#### 8.105.1.2 enum [TOsclSocketStatEvent](#)

Socket diagnostics.

Enumeration values:

[EOsclSocket\\_RequestAO\\_Success](#)  
[EOsclSocket\\_RequestAO\\_Canceled](#)  
[EOsclSocket\\_RequestAO\\_Error](#)  
[EOsclSocket\\_RequestAO\\_Timeout](#)  
[EOsclSocket\\_ServRequestIssued](#)  
[EOsclSocket\\_ServPoll](#)  
[EOsclSocket\\_OS](#)  
[EOsclSocket\\_Readable](#)  
[EOsclSocket\\_Writable](#)



**EOsclSocket\_Except**  
**EOsclSocket\_DataRecv**  
**EOsclSocket\_DataSent**  
**EOsclSocket\_ServRequestComplete**  
**EOsclSocket\_ServRequestCancelIssued**  
**EOsclSocketServ\_LoopsockOk**  
**EOsclSocketServ\_LoopsockError**

## 8.106 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\\_SERVER\\_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::EPriorityNominal

[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 (OsciActiveObject::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 loop 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\_SERVER\_STATS 0**

For detailed performance breakdown of time spend in [OsciSocketServer](#) AO. Output is logged under "OsciSchedulerPerfStats" 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 [OscIpMReq](#)
- class [OscNetworkAddress](#)
- class [OscSocketObserver](#)
- class [OscSocketTOS](#)

### Defines

- #define [PVNETWORKADDRESS\\_LEN](#) 50

### Enumerations

- enum [TPVSocketFxn](#) { [EPVSocketSend](#) = 0, [EPVSocketSendTo](#), [EPVSocketRecv](#), [EPVSocketRecvFrom](#), [EPVSocketConnect](#), [EPVSocketAccept](#), [EPVSocketShutdown](#), [EPVSocketBind](#), [EPVSocketListen](#), [EPVSocket\\_Last](#) }
- enum [TPVSocketEvent](#) { [EPVSocketSuccess](#), [EPVSocketPending](#), [EPVSocketTimeout](#), [EPVSocketFailure](#), [EPVSocketCancel](#), [EPVSocketNotImplemented](#) }
- enum [TPVSocketShutdown](#) { [EPVSocketSendShutdown](#), [EPVSocketRecvShutdown](#), [EPVSocketBothShutdown](#) }
- enum [TPVSocketOptionName](#) { [EPVIPMulticastTTL](#), [EPVIPAddMembership](#), [EPVIPTOS](#), [EPVSocketReuseAddr](#) }
- 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:**

**EPVIPMulticastTTL**  
**EPVIPAddMembership**  
**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\\_strcmp](#) (const [oscl\\_wchar](#) \*str1, const [oscl\\_wchar](#) \*str2)
- OSCL\_IMPORT\_REF int32 [oscl\\_strncmp](#) (const char \*str1, const char \*str2, uint32 count)
- OSCL\_IMPORT\_REF int32 [oscl\\_strncmp](#) (const [oscl\\_wchar](#) \*str1, const [oscl\\_wchar](#) \*str2, uint32 count)
- OSCL\_IMPORT\_REF char \* [oscl\\_strncat](#) (char \*dest, const char \*src, uint32 count)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl\\_strncat](#) ([oscl\\_wchar](#) \*dest, const [oscl\\_wchar](#) \*src, uint32 count)
- OSCL\_IMPORT\_REF const char \* [oscl\\_strchr](#) (const char \*str, int32 c)
- OSCL\_IMPORT\_REF char \* [oscl\\_strchr](#) (char \*str, int32 c)
- OSCL\_IMPORT\_REF const [oscl\\_wchar](#) \* [oscl\\_strchr](#) (const [oscl\\_wchar](#) \*str, int32 c)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl\\_strchr](#) ([oscl\\_wchar](#) \*str, int32 c)
- OSCL\_IMPORT\_REF const char \* [oscl\\_strchr](#) (const char \*str, int32 c)
- OSCL\_IMPORT\_REF char \* [oscl\\_strchr](#) (char \*str, int32 c)
- OSCL\_IMPORT\_REF const [oscl\\_wchar](#) \* [oscl\\_strchr](#) (const [oscl\\_wchar](#) \*str, int32 c)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl\\_strchr](#) ([oscl\\_wchar](#) \*str, int32 c)
- OSCL\_IMPORT\_REF char \* [oscl\\_strset](#) (char \*dest, char val, uint32 count)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl\\_strset](#) ([oscl\\_wchar](#) \*dest, [oscl\\_wchar](#) val, uint32 count)
- OSCL\_IMPORT\_REF int32 [oscl\\_CIstrcmp](#) (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF int32 [oscl\\_CIstrcmp](#) (const [oscl\\_wchar](#) \*str1, const [oscl\\_wchar](#) \*str2)
- OSCL\_IMPORT\_REF int32 [oscl\\_CIstrncmp](#) (const char \*str1, const char \*str2, uint32 count)
- OSCL\_IMPORT\_REF int32 [oscl\\_CIstrncmp](#) (const [oscl\\_wchar](#) \*str1, const [oscl\\_wchar](#) \*str2, uint32 count)
- OSCL\_IMPORT\_REF char [oscl\\_tolower](#) (const char car)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) [oscl\\_tolower](#) (const [oscl\\_wchar](#) car)
- OSCL\_IMPORT\_REF bool [oscl\\_isLetter](#) (const char car)
- OSCL\_IMPORT\_REF const char \* [oscl\\_strstr](#) (const char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF char \* [oscl\\_strstr](#) (char \*str1, const char \*str2)
- OSCL\_IMPORT\_REF const [oscl\\_wchar](#) \* [oscl\\_strstr](#) (const [oscl\\_wchar](#) \*str1, const [oscl\\_wchar](#) \*str2)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl\\_strstr](#) ([oscl\\_wchar](#) \*str1, const [oscl\\_wchar](#) \*str2)
- OSCL\_IMPORT\_REF char \* [oscl\\_strcat](#) (char \*dest, const char \*src)
- OSCL\_IMPORT\_REF [oscl\\_wchar](#) \* [oscl\\_strcat](#) ([oscl\\_wchar](#) \*dest, const [oscl\\_wchar](#) \*src)

### 8.108.1 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 [CHepRep](#)
- 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, [OscFloat](#) &value)
- OSCL\_IMPORT\_REF bool [PV\\_atof](#) (const char \*buf, int length, [OscFloat](#) &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 [OscITCPSocketI](#)

## 8.119 oscl\_thread.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_procstatus.h"
#include "oscl_base.h"
```

### Data Structures

- class [OscThread](#)

### Typedefs

- typedef [TOscThreadFuncRet](#)(OSCL\_THREAD\_DECL \* [TOscThreadFuncPtr](#) )(TOscThreadFuncArg)

### Enumerations

- enum [OscThread\\_State](#) { [Start\\_on\\_creation](#), [Suspend\\_on\\_creation](#) }
- enum [OscThreadPriority](#) { [ThreadPriorityLowest](#), [ThreadPriorityLow](#), [ThreadPriorityBelowNormal](#), [ThreadPriorityNormal](#), [ThreadPriorityAboveNormal](#), [ThreadPriorityHighest](#), [ThreadPriorityTimeCritical](#) }
- enum [TOscThreadTerminate](#) { [EOscThreadTerminate\\_Join](#), [EOscThreadTerminate\\_Kill](#), [EOscThreadTerminate\\_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 [TOscThreadFuncRet](#)(OSCL\_THREAD\_DECL \* [TOscThreadFuncPtr](#))([TOscThreadFuncArg](#))

### 8.119.3 Enumeration Type Documentation

8.119.3.1 enum [OscThread\\_State](#)

Enumeration values:

[Start\\_on\\_creation](#)

[Suspend\\_on\\_creation](#)

8.119.3.2 enum [OscThreadPriority](#)

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 [OscTickCount](#)

### 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 to classes [NTPTIME](#) and [TimeValue](#) for getting, manipulating, and formatting time values. The [TimeValue](#) class is based on the native system time format while [NTPTIME](#) is used for the standard Network Time Protocol format.

```
#include "oscl_base.h"
#include "osclconfig_time.h"
#include "oscl_int64_utils.h"
#include "oscl_time.inl"
```

### Data Structures

- class [NTPTIME](#)

*The NTPTIME class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900.*

- class [TimeValue](#)

*The TimeValue class represents a time value in a format native to the system.*

### Typedefs

- typedef char [CtimeStrBuf](#) [CTIME\_BUFFER\_SIZE]
- typedef char [PV8601timeStrBuf](#) [PV8601TIME\_BUFFER\_SIZE]
- typedef char [ISO8601timeStrBuf](#) [ISO8601TIME\_BUFFER\_SIZE]

### Enumerations

- enum [TimeUnits](#) { SECONDS = 0, MILLISECONDS = 1, MICROSECONDS = 2 }

*The TimeUnits enum can be used when constructing a [TimeValue](#) class.*

### Functions

- OSCL\_IMPORT\_REF void [PV8601ToRFC822](#) ([PV8601timeStrBuf](#) pv8601\_buffer, [CtimeStrBuf](#) ctime\_buffer)
- OSCL\_IMPORT\_REF void [ISO8601ToRFC822](#) ([ISO8601timeStrBuf](#) iso8601\_buffer, [CtimeStrBuf](#) ctime\_buffer)
- OSCL\_IMPORT\_REF void [RFC822ToPV8601](#) ([CtimeStrBuf](#) ctime\_buffer, [PV8601timeStrBuf](#))
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) operator- (const [TimeValue](#) &a, const [TimeValue](#) &b)
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) operator+ (const [TimeValue](#) &a, const int32 bSeconds)
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) operator+ (const int32 aSeconds, const [TimeValue](#) &b)
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) operator- (const [TimeValue](#) &a, const int32 bSeconds)
- OSCL\_COND\_IMPORT\_REF [TimeValue](#) operator- (const int32 aSeconds, const [TimeValue](#) &b)

## Variables

- const int [CTIME\\_BUFFER\\_SIZE](#) = 26
- const int [PV8601TIME\\_BUFFER\\_SIZE](#) = 21
- const int [ISO8601TIME\\_BUFFER\\_SIZE](#) = 21
- const long [USEC\\_PER\\_SEC](#) = 1000000
- const long [MSEC\\_PER\\_SEC](#) = 1000
- const uint32 [unix\\_ntp\\_offset](#) = 2208988800U

### 8.121.1 Detailed Description

The file [oscl\\_time.h](#) defines to classes [NTPTIME](#) and [TimeValue](#) for getting, manipulating, and formatting time values. The [TimeValue](#) class is based on the native system time format while [NTPTIME](#) is used for the standard Network Time Protocol format.

## 8.122 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 [OscTimer](#)
- class [OscTimerObserver](#)

## 8.123 oscl\_tls.h File Reference

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

### Data Structures

- class [OscTLS](#)
- class [OscTLSRegistry](#)
- class [TLSStorageOps](#)

### Defines

- #define [OSCL\\_TLS\\_BASE\\_SLOTS](#) [OSCL\\_TLS\\_ID\\_BASE\\_LAST](#) +1
- #define [OSCL\\_TLS\\_EXTERNAL\\_SLOTS](#) 0
- #define [OSCL\\_TLS\\_MAX\\_SLOTS](#) ( [OSCL\\_TLS\\_BASE\\_SLOTS](#) + [OSCL\\_TLS\\_EXTERNAL\\_SLOTS](#) )

### Typedefs

- typedef [OscAny](#) [TOscTlsKey](#)

### Variables

- const uint32 [OSCL\\_TLS\\_ID\\_MAGICNUM](#) = 0
- const uint32 [OSCL\\_TLS\\_ID\\_ERRORHOOK](#) = 1
- const uint32 [OSCL\\_TLS\\_ID\\_PVLOGGER](#) = 2
- const uint32 [OSCL\\_TLS\\_ID\\_TEST](#) = 3
- const uint32 [OSCL\\_TLS\\_ID\\_PVSCHEDULER](#) = 4
- const uint32 [OSCL\\_TLS\\_ID\\_PERRORTRAP](#) = 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 [OscMemoryFragment](#)

### Typedefs

- typedef int [c\\_bool](#)  
*The c\_bool type is mapped to an integer to provide a bool type for C interfaces.*
- typedef void [OscAny](#)  
*The OscAny is meant to be used the context of a generic pointer (i.e., no specific type).*
- typedef char [mbchar](#)  
*mbchar is multi-byte char (e.g., UTF-8) with null termination.*
- typedef unsigned int [uint](#)  
*The uint type is a convenient abbreviation for unsigned int.*
- typedef uint8 [octet](#)  
*The octet type is meant to be used for referring to a byte or collection bytes without suggesting anything about the underlying meaning of the bytes.*
- typedef float [OscFloat](#)  
*The Float type defined as OscFloat.*
- typedef OSCL\_NATIVE\_INT64\_TYPE [int64](#)
- typedef OSCL\_NATIVE\_UINT64\_TYPE [uint64](#)
- typedef OSCL\_NATIVE\_WCHAR\_TYPE [oscl\\_wchar](#)
- typedef [oscl\\_wchar](#) [OSCL\\_TCHAR](#)  
*define OSCL\_TCHAR*

### 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 [OscUDPSocketI](#)

## 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 OslUId32.

```
#include "oscl_base_macros.h"
#include "oscl_mem_basic_functions.h"
```

### Data Structures

- struct [OslUuid](#)

### Defines

- #define [EMPTY\\_UUID](#) PVUuid(0,0,0,0,0,0,0,0,0,0)
- #define [BYTES\\_IN\\_UUID\\_ARRAY](#) 8

### Typedefs

- typedef uint32 [OslUId32](#)

#### 8.128.1 Detailed Description

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OslUId32.

#### 8.128.2 Define Documentation

**8.128.2.1** #define [BYTES\\_IN\\_UUID\\_ARRAY](#) 8

**8.128.2.2** #define [EMPTY\\_UUID](#) PVUuid(0,0,0,0,0,0,0,0,0,0)

#### 8.128.3 Typedef Documentation

**8.128.3.1** typedef uint32 [OslUId32](#)

## 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 [OscVector](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_mem_basic_functions.h"
#include "oscl_assert.h"
#include "oscl_opaque_type.h"
#include "oscl_defalloc.h"
#include "oscl_base.h"
```

### Data Structures

- class [OscVector](#)
- class [OscVectorBase](#)

#### 8.130.1 Detailed Description

The file [oscl\\_vector.h](#) defines the template class [OscVector](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

## 8.131 osclconfig.h File Reference

This file contains configuration information for the linux platform.

```
#include <dirent.h>
#include <dlfcn.h>
#include "osclconfig_limits_typedefs.h"
#include "osclconfig_unix_android.h"
#include "osclconfig_ix86.h"
#include "osclconfig_check.h"
```

### Defines

- #define [OSCL\\_HAS\\_ANDROID\\_SUPPORT](#) 1
- #define [OSCL\\_HAS\\_ANDROID\\_FILE\\_IO\\_SUPPORT](#) 1
- #define [OSCL\\_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 [OscIsValidInetAddr](#)(addr) (inet\_addr(addr)!=INADDR\_NONE)
- #define [OscMakeSockAddr](#)(sockaddr, port, addrstr, ok)
- #define [OscUnMakeSockAddr](#)(sockaddr, addrstr) addrstr=inet\_ntoa(sockaddr.sin\_addr);
- #define [OscMakeInAddr](#)(in\_addr, addrstr, ok)
- #define [OscUnMakeInAddr](#)(in\_addr, addrstr) addrstr=inet\_ntoa(in\_addr);
- #define [OscSetRecvBufferSize](#)(s, val, ok, err)
- #define [OscBind](#)(s, addr, ok, err)

- #define [OscSetSockOpt](#)(s, optLevel, optName, optVal, optLen, ok, err)
- #define [OscJoin](#)(s, addr, ok, err)
- #define [OscListen](#)(s, size, ok, err)
- #define [OscAccept](#)(s, accept\_s, ok, err, wouldblock)
- #define [OscSetNonBlocking](#)(s, ok, err)
- #define [OscShutdown](#)(s, how, ok, err)
- #define [OscSocket](#)(s, fam, type, prot, ok, err)
- #define [OscSendTo](#)(s, buf, len, addr, ok, err, nbytes, wouldblock)
- #define [OscSend](#)(s, buf, len, ok, err, nbytes, wouldblock)
- #define [OscCloseSocket](#)(s, ok, err)
- #define [OscConnect](#)(s, addr, ok, err, wouldblock)
- #define [OscGetPeerName](#)(s, name, namelen, ok, err)
- #define [OscGetAsyncSockErr](#)(s, ok, err)
- #define [OscPipe](#)(x) pipe(x)
- #define [OscReadFD](#)(fd, buf, cnt) read(fd,buf,cnt)
- #define [OscWriteFD](#)(fd, buf, cnt) write(fd,buf,cnt)
- #define [OscConnectComplete](#)(s, wset, eset, success, fail, ok, err)
- #define [OscRecv](#)(s, buf, len, ok, err, nbytes, wouldblock)
- #define [OscRecvFrom](#)(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)
- #define [OscSocketSelect](#)(nfd, rd, wr, ex, timeout, ok, err, nhandles)
- #define [OscSocketStartup](#)(ok)
- #define [OscSocketCleanup](#)(ok)
- #define [OscGethostbyname](#)(name, hostent, ok, err)
- #define [OscGetDottedAddr](#)(hostent, dottedaddr, ok)
- #define [OscGetDottedAddrVector](#)(hostent, dottedaddr, dottedaddrvect, ok)
- #define [OSCL\\_SD\\_RECEIVE](#) SHUT\_RD
- #define [OSCL\\_SD\\_SEND](#) SHUT\_WR
- #define [OSCL\\_SD\\_BOTH](#) SHUT\_RDWR
- #define [OSCL\\_AF\\_INET](#) AF\_INET
- #define [OSCL SOCK\\_STREAM](#) SOCK\_STREAM
- #define [OSCL SOCK\\_DGRAM](#) SOCK\_DGRAM
- #define [OSCL\\_IPPROTO\\_IP](#) IPPROTO\_IP
- #define [OSCL\\_IPPROTO\\_TCP](#) IPPROTO\_TCP
- #define [OSCL\\_IPPROTO\\_UDP](#) IPPROTO\_UDP
- #define [OSCL\\_SOL\\_SOCKET](#) SOL\_SOCKET
- #define [OSCL\\_SOL\\_IP](#) IPPROTO\_IP
- #define [OSCL\\_SOL\\_TCP](#) IPPROTO\_TCP
- #define [OSCL\\_SOL\\_UDP](#) IPPROTO\_UDP
- #define [OSCL SOCKOPT\\_IP\\_MULTICAST\\_TTL](#) IP\_MULTICAST\_TTL
- #define [OSCL SOCKOPT\\_IP\\_ADD\\_MEMBERSHIP](#) IP\_ADD\_MEMBERSHIP
- #define [OSCL SOCKOPT\\_IP\\_TOS](#) IP\_TOS
- #define [OSCL SOCKOPT\\_SOL\\_REUSEADDR](#) SO\_REUSEADDR

## Typedefs

- typedef int [TOscSocket](#)
- typedef sockaddr\_in [TOscSockAddr](#)
- typedef socklen\_t [TOscSockAddrLen](#)
- typedef ip\_mreq [TIpMReq](#)
- typedef hostent [TOscHostent](#)
- typedef off64\_t [TOscFileOffset](#)

### **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 OslBind(s, addr, ok, err)

**Value:**

```
TOslSockAddr* tmpadr = &addr;\
    sockaddr* sadr = OSL_STATIC_CAST(sockaddr*, tmpadr);\
    ok=(bind(s,sadr,sizeof(addr))!=(-1));\
    if (!ok)err=errno
```

### 8.139.2.33 #define OslCloseSocket(s, ok, err)

**Value:**

```
ok=(close(s)!=(-1));\
    if (!ok)err=errno
```

### 8.139.2.34 #define OslConnect(s, addr, ok, err, wouldblock)

**Value:**

```
TOslSockAddr* tmpadr = &addr;\
    sockaddr* sadr = OSL_STATIC_CAST(sockaddr*, tmpadr);\
    ok=(connect(s,sadr,sizeof(addr))!=(-1));\
    if (!ok){err=errno;wouldblock=(err==EINPROGRESS);}
```

### 8.139.2.35 #define OslConnectComplete(s, wset, eset, success, fail, ok, err)

**Value:**

```
success=fail=false;\
    if (FD_ISSET(s,&eset))\
        {fail=true;OslGetAsyncSockErr(s,ok,err);}\
    else if (FD_ISSET(s,&wset))\
        {OslGetAsyncSockErr(s,ok,err);if (ok && err==0)success=true;else fail=true;}
```

### 8.139.2.36 #define OslGetAsyncSockErr(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 OsciGetDottedAddr(hostent, dottedaddr, ok)

**Value:**

```
long *_hostaddr=(long*)hostent->h_addr_list[0];\
struct in_addr _inaddr;\
_inaddr.s_addr=*_hostaddr;\
dottedaddr=inet_ntoa(_inaddr);\
ok=(dottedaddr!=NULL);
```

### 8.139.2.38 #define OsciGetDottedAddrVector(hostent, dottedaddr, dottedaddrvect, ok)

**Value:**

```
if(dottedaddrvect)\
{\
long **_addrlist=(long**)hostent->h_addr_list;\
for(int i = 0; _addrlist[i] != NULL; i++){\\
struct in_addr _inaddr;\
_inaddr.s_addr=*_addrlist[i];\\
OsciNetworkAddress addr(inet_ntoa(_inaddr), 0);\\
dottedaddrvect->push_back(addr);\\
}\\
if (!dottedaddrvect->empty())\\
{dottedaddr->port = dottedaddrvect->front().port; dottedaddr->ipAddr.Set(dottedaddrvect->front().ipAddr);\
ok=(!dottedaddrvect->empty() && ((*dottedaddrvect)[0]).ipAddr.Str() != NULL);\\
}\\
else\\
{\
char *add;\
OsciGetDottedAddr(hostent,add,ok);\
if(ok) dottedaddr->ipAddr.Set(add);\
}
```

### 8.139.2.39 #define OsciGethostbyname(name, hostent, ok, err)

**Value:**

```
hostent=gethostbyname((const char*)name);\
ok=(hostent!=NULL);\
if (!ok)err=errno;
```

### 8.139.2.40 #define OsciGetPeerName(s, name, namelen, ok, err)

**Value:**

```
ok=(getpeername(s,(sockaddr*)&name,(socklen_t*)&namelen) != (-1) );\\
if (!ok)err=errno
```

### 8.139.2.41 #define OsciJoin(s, addr, ok, err)

**Value:**

```
{\
    struct ip_mreq mreq; \
    void* p = &addr; \
    ok=(bind(s,(sockaddr*)p,sizeof(addr))!=(-1));\
    mreq.imr_multiaddr.s_addr = addr.sin_addr.s_addr ; \
    mreq.imr_interface.s_addr = htonl(INADDR_ANY); \
    ok=(setsockopt(s, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq, sizeof(struct ip_mreq))!=(-1)); \
    if (!ok)err=errno;\
}
```

#### 8.139.2.42 #define OsciListen(s, size, ok, err)

##### Value:

```
ok=(listen(iSocket,qSize)!=(-1));\
if (!ok)err=errno
```

#### 8.139.2.43 #define OsciMakeInAddr(in\_addr, addrstr, ok)

##### Value:

```
int32 result = inet_aton((const char*)addrstr, &in_addr);\
ok=(result!=0);
```

#### 8.139.2.44 #define OsciMakeSockAddr(sockaddr, port, addrstr, ok)

##### Value:

```
sockaddr.sin_family=OSCL_AF_INET;\
sockaddr.sin_port=htons(port);\
int32 result=inet_aton((const char*)addrstr,&sockaddr.sin_addr);\
ok=(result!=0);
```

#### 8.139.2.45 #define OsciPipe(x) pipe(x)

#### 8.139.2.46 #define OsciReadFD(fd, buf, cnt) read(fd,buf,cnt)

#### 8.139.2.47 #define OsciRecv(s, buf, len, ok, err, nbytes, wouldblock)

##### Value:

```
nbytes=recv(s,(void *) (buf),(size_t)(len),0);\
ok=(nbytes!=(-1));\
if (!ok){err=errno;wouldblock=(err==EAGAIN);}
```



### 8.139.2.48 #define OsciRecvFrom(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)

Value:

```
{\
void* p=paddr;\
nbytes=recvfrom(s,(void*)(buf),(size_t)(len),0,(struct sockaddr*)p,paddrlen);\
    ok=(nbytes!=(-1));\
    if (!ok){err=errno;wouldblock=(err==EAGAIN);}\
}
```

### 8.139.2.49 #define OsciSend(s, buf, len, ok, err, nbytes, wouldblock)

Value:

```
nbytes=send(s,(const void*)(buf),(size_t)(len),0);\
    ok=(nbytes!=(-1));\
    if (!ok){err=errno;wouldblock=(err==EAGAIN || err==EWOULDBLOCK);}
```

### 8.139.2.50 #define OsciSendTo(s, buf, len, addr, ok, err, nbytes, wouldblock)

Value:

```
TOsciSockAddr* tmpadr = &addr;\
    sockaddr* sadr = OSC_STATIC_CAST(sockaddr*, tmpadr);\
    nbytes=sendto(s,(const void*)(buf),(size_t)(len),0,sadr,(socklen_t)sizeof(addr));\
    ok=(nbytes!=(-1));\
    if (!ok){err=errno;wouldblock=(err==EAGAIN || err==EWOULDBLOCK);}
```

### 8.139.2.51 #define OsciSetNonBlocking(s, ok, err)

Value:

```
ok=(fcntl(s,F_SETFL,O_NONBLOCK)!=(-1));\
    if (!ok)err=errno
```

### 8.139.2.52 #define OsciSetRecvBufferSize(s, val, ok, err)

Value:

```
ok=(setsockopt(s,SOL_SOCKET,SO_RCVBUF,(char*)&val, sizeof(int)) !=-1);\
    if (!ok)err=errno
```

### 8.139.2.53 #define OsciSetSockOpt(s, optLevel, optName, optVal, optLen, ok, err)

Value:

```
ok=(setsockopt(s,optLevel,optName,OSC_STATIC_CAST(const char*,optVal),optLen) != (-1));\
    if (!ok)err=errno
```

**8.139.2.54 #define OscShutdown(s, how, ok, err)****Value:**

```
ok=(shutdown(iSocket,how)!=(-1));\
if (!ok)err=errno
```

**8.139.2.55 #define OscSocket(s, fam, type, prot, ok, err)****Value:**

```
s=socket(fam,type,prot);\
ok=(s!=(-1));\
if (!ok)err=errno
```

**8.139.2.56 #define OscSocketCleanup(ok)****Value:**

```
signal(SIGPIPE,SIG_DFL);\
ok=true
```

**8.139.2.57 #define OscSocketSelect(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 OscSocketStartup(ok)****Value:**

```
signal(SIGPIPE,SIG_IGN);\
ok=true
```

**8.139.2.59** `#define OsciUnMakeInAddr(in_addr, addrstr) addrstr=inet_ntoa(in_addr);`

**8.139.2.60** `#define OsciUnMakeSockAddr(sockaddr, addrstr) addrstr=inet_ntoa(sockaddr.sin_addr);`

**8.139.2.61** `#define OsciValidInetAddr(addr) (inet_addr(addr)!=INADDR_NONE)`

**8.139.2.62** `#define OsciWriteFD(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 TOsciFileOffset`

**8.139.3.3** `typedef struct hostent TOsciHostent`

**8.139.3.4** `typedef struct sockaddr_in TOsciSockAddr`

**8.139.3.5** `typedef socklen_t TOsciSockAddrLen`

**8.139.3.6** `typedef int TOsciSocket`

## 8.140 osclconfig\_io\_check.h File Reference

### Typedefs

- typedef [TOscFileOffset](#) [\\_\\_verify\\_\\_TOscFileOffset\\_\\_defined\\_\\_](#)

#### 8.140.1 Typedef Documentation

##### 8.140.1.1 typedef [TOscFileOffset](#) [\\_\\_verify\\_\\_TOscFileOffset\\_\\_defined\\_\\_](#)

type `TOscFileOffset` should be defined as the type used for file size and offsets on the target platform.  
Example: `typedef size_t TOscFileOffset;`

## 8.141 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 oscconfig\_proc\_check.h File Reference

### Typedefs

- typedef [TOscThreadId](#) \_\_verify\_\_[TOscThreadId](#)\_\_defined\_\_
- typedef [TOscThreadFuncRet](#) \_\_verify\_\_[TOscThreadFuncRet](#)\_\_defined\_\_
- typedef [TOscThreadFuncArg](#) \_\_verify\_\_[TOscThreadFuncArg](#)\_\_defined\_\_
- typedef [TOscThreadObject](#) \_\_verify\_\_[TOscThreadObject](#)\_\_defined\_\_
- typedef [TOscMutexObject](#) \_\_verify\_\_[TOscMutexObject](#)\_\_defined\_\_
- typedef [TOscSemaphoreObject](#) \_\_verify\_\_[TOscSemaphoreObject](#)\_\_defined\_\_
- typedef [TOscConditionObject](#) \_\_verify\_\_[TOscConditionObject](#)\_\_defined\_\_

### 8.149.1 Typedef Documentation

#### 8.149.1.1 typedef [TOscConditionObject](#) \_\_verify\_\_[TOscConditionObject](#)\_\_defined\_\_

type [TOscConditionObject](#) should be defined as the type used as a condition variable on the target platform. Example: typedef pthread\_cond\_t [TOscConditionObject](#);

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 [TOscConditionObject](#); //not used

#### 8.149.1.2 typedef [TOscMutexObject](#) \_\_verify\_\_[TOscMutexObject](#)\_\_defined\_\_

type [TOscMutexObject](#) should be defined as the type used as a mutex object or handle on the target platform. Example: typedef pthread\_mutex\_t [TOscMutexObject](#);

#### 8.149.1.3 typedef [TOscSemaphoreObject](#) \_\_verify\_\_[TOscSemaphoreObject](#)\_\_defined\_\_

type [TOscSemaphoreObject](#) should be defined as the type used as a mutex object or handle on the target platform. Example: typedef sem\_t [TOscSemaphoreObject](#);

#### 8.149.1.4 typedef [TOscThreadFuncArg](#) \_\_verify\_\_[TOscThreadFuncArg](#)\_\_defined\_\_

type [TOscThreadFuncArg](#) should be defined as the type used as a thread function argument on the target platform. Example: typedef LPVOID [TOscThreadFuncArg](#);

#### 8.149.1.5 typedef [TOscThreadFuncRet](#) \_\_verify\_\_[TOscThreadFuncRet](#)\_\_defined\_\_

type [TOscThreadFuncRet](#) should be defined as the type used as a thread function return value on the target platform. Example: typedef DWORD [TOscThreadFuncRet](#);

#### 8.149.1.6 typedef [TOscThreadId](#) \_\_verify\_\_[TOscThreadId](#)\_\_defined\_\_

type [TOscThreadId](#) should be defined as the type used as a thread ID on the target platform. Example: typedef DWORD [TOscThreadId](#);

**8.149.1.7 typedef `TOscIThreadObject` `__verify__TOscIThreadObject__defined__`**

type `TOscIThreadObject` should be defined as the type used as a thread object or handle on the target platform. Example: `typedef pthread_t TOscIThreadObject;`

## 8.150 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 [TOscIThreadId](#)
- typedef void \* [TOscIThreadFuncArg](#)
- typedef void \* [TOscIThreadFuncRet](#)
- typedef pthread\_t [TOscIThreadObject](#)
- typedef pthread\_mutex\_t [TOscIMutexObject](#)
- typedef int [TOscISemaphoreObject](#)
- typedef pthread\_cond\_t [TOscIConditionObject](#)

### 8.150.1 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 oscconfig\_proc\_unix\_common.h File Reference

```
#include <time.h>
#include <semaphore.h>
#include <pthread.h>
#include <errno.h>
```

### Defines

- #define [OSCL\\_HAS\\_SYMBIAN\\_SCHEDULER](#) 0
- #define [OSCL\\_HAS\\_THREAD\\_SUPPORT](#) 1
- #define [OSCL\\_HAS\\_NON\\_PREEMPTIVE\\_THREAD\\_SUPPORT](#) 0
- #define [OSCL\\_HAS\\_SEM\\_TIMEDWAIT\\_SUPPORT](#) 1
- #define [OSCL\\_HAS\\_PTHREAD\\_SUPPORT](#) 1
- #define [OSCL\\_THREAD\\_DECL](#)

### Typedefs

- typedef pthread\_t [TOscIThreadId](#)
- typedef void \* [TOscIThreadFuncArg](#)
- typedef void \* [TOscIThreadFuncRet](#)
- typedef pthread\_t [TOscIThreadObject](#)
- typedef pthread\_mutex\_t [TOscIMutexObject](#)
- typedef sem\_t [TOscISemaphoreObject](#)
- typedef pthread\_cond\_t [TOscIConditionObject](#)

### 8.151.1 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 TOscConditionObject`

8.151.2.2 `typedef pthread_mutex_t TOscMutexObject`

8.151.2.3 `typedef sem_t TOscSemaphoreObject`

8.151.2.4 `typedef void* TOscThreadFuncArg`

8.151.2.5 `typedef void* TOscThreadFuncRet`

8.151.2.6 `typedef pthread_t TOscThreadId`

8.151.2.7 `typedef pthread_t TOscThreadObject`

## 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` [OscBasicTimeStruct](#)
- `typedef tm` [OscBasicDateTimeStruct](#)

#### 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` [OscBasicDateTimeStruct](#)

8.152.2.2 `typedef struct timeval` [OscBasicTimeStruct](#)

## 8.153 oscconfig\_time\_check.h File Reference

### Typedefs

- typedef [OscBasicTimeStruct](#) \_\_Validate\_\_BasicTimeStruct\_\_
- typedef [OscBasicDateTimeStruct](#) \_\_Validate\_\_BasicTimeDateStruct\_\_

### 8.153.1 Typedef Documentation

#### 8.153.1.1 typedef [OscBasicDateTimeStruct](#) \_\_Validate\_\_BasicTimeDateStruct\_\_

OscBasicDateTimeStruct type should be defined to the platform-specific date + time type.

#### 8.153.1.2 typedef [OscBasicTimeStruct](#) \_\_Validate\_\_BasicTimeStruct\_\_

OscBasicTimeStruct 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 [TOscITlsKey](#)
- typedef pthread\_mutex\_t [TOscBasicLockObject](#)



## 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 [TOscTlsKey](#)
- typedef pthread\_mutex\_t [TOscBasicLockObject](#)



## 8.155.1 Define Documentation

- 8.155.1.1 `#define _STRLIT(x) L ## x`
- 8.155.1.2 `#define _STRLIT_CHAR(x) x`
- 8.155.1.3 `#define _STRLIT_WCHAR(x) L ## x`
- 8.155.1.4 `#define INT64(x) x##LL`
- 8.155.1.5 `#define INT64_HILO(high, low) (((high##LL)<<32)|low)`
- 8.155.1.6 `#define OSCL_DISABLE_INLINES 0`
- 8.155.1.7 `#define OSCL_HAS_ANSI_MATH_SUPPORT 1`
- 8.155.1.8 `#define OSCL_HAS_ANSI_STDIO_SUPPORT 1`
- 8.155.1.9 `#define OSCL_HAS_ANSI_STDLIB_SUPPORT 1`
- 8.155.1.10 `#define OSCL_HAS_ANSI_STRING_SUPPORT 1`
- 8.155.1.11 `#define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 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 Message 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](#)
- ~OscIAcceptMethod
  - OscIAcceptMethod, [308](#)
- ~OscIActiveObject
  - OscIActiveObject, [311](#)
- ~OscIAllocDestructDealloc
  - OscIAllocDestructDealloc, [314](#)
- ~OscIAsyncFile
  - OscIAsyncFile, [317](#)
- ~OscIAsyncFileBuffer
  - OscIAsyncFileBuffer, [320](#)
- ~OscIBinIStream
  - OscIBinIStream, [324](#)
- ~OscIBinOStream
  - OscIBinOStream, [331](#)
- ~OscIBindMethod
  - OscIBindMethod, [322](#)
- ~OscICacheObserver
  - OscI\_File::OscICacheObserver, [187](#)
- ~OscIComponentRegistry
  - OscIComponentRegistry, [344](#)
- ~OscIComponentRegistryElement
  - OscIComponentRegistryElement, [346](#)
- ~OscIConnectMethod
  - OscIConnectMethod, [348](#)
- ~OscIDNS
  - OscIDNS, [351](#)
- ~OscIDNSI
  - OscIDNSI, [353](#)
- ~OscIDNSIBase
  - OscIDNSIBase, [356](#)
- ~OscIDNSObserver
  - OscIDNSObserver, [361](#)
- ~OscIDNSRequest
  - OscIDNSRequest, [362](#)
- ~OscIDestructDealloc
  - OscIDestructDealloc, [350](#)
- ~OscIExclusiveArrayPtr
  - OscIExclusiveArrayPtr, [381](#)
- ~OscIExclusivePtr
  - OscIExclusivePtr, [384](#)
- ~OscIExclusivePtrA
  - OscIExclusivePtrA, [387](#)

- ~OsciExecSchedulerCommonBase
  - OsciExecSchedulerCommonBase, [395](#)
- ~OsciFileCache
  - OsciFileCache, [402](#)
- ~OsciGetHostByNameMethod
  - OsciGetHostByNameMethod, [413](#)
- ~OsciIPSocketI
  - OsciIPSocketI, [419](#)
- ~OsciJump
  - OsciJump, [421](#)
- ~OsciListenMethod
  - OsciListenMethod, [422](#)
- ~OsciLockBase
  - OsciLockBase, [424](#)
- ~OsciMemAudit
  - OsciMemAudit, [429](#)
- ~OsciMemPoolFixedChunkAllocator
  - OsciMemPoolFixedChunkAllocator, [444](#)
- ~OsciMemPoolFixedChunkAllocatorObserver
  - OsciMemPoolFixedChunkAllocator-Observer, [447](#)
- ~OsciMemPoolResizableAllocator
  - OsciMemPoolResizableAllocator, [449](#)
- ~OsciMemPoolResizableAllocatorMemoryObserver
  - OsciMemPoolResizableAllocatorMemory-Observer, [456](#)
- ~OsciMemPoolResizableAllocatorObserver
  - OsciMemPoolResizableAllocatorObserver, [457](#)
- ~OsciMemStatsNode
  - OsciMemStatsNode, [458](#)
- ~OsciMutex
  - OsciMutex, [459](#)
- ~OsciNativeFile
  - OsciNativeFile, [463](#)
- ~OsciNullLock
  - OsciNullLock, [467](#)
- ~OsciPriorityQueue
  - OsciPriorityQueue, [471](#)
- ~OsciPriorityQueueBase
  - OsciPriorityQueueBase, [474](#)
- ~OsciRecvFromMethod
  - OsciRecvFromMethod, [486](#)
- ~OsciRecvMethod
  - OsciRecvMethod, [490](#)
- ~OsciRefCounter
  - OsciRefCounter, [492](#)
- ~OsciRefCounterDA
  - OsciRefCounterDA, [494](#)
- ~OsciRefCounterMTDA
  - OsciRefCounterMTDA, [498](#)
- ~OsciRefCounterMTSA
  - OsciRefCounterMTSA, [500](#)
- ~OsciRefCounterMemFrag
  - OsciRefCounterMemFrag, [496](#)
- ~OsciRefCounterSA
  - OsciRefCounterSA, [502](#)
- ~OsciRegistryAccessClient
  - OsciRegistryAccessClient, [504](#)
- ~OsciRegistryClient
  - OsciRegistryClient, [509](#)
- ~OsciRegistryServTlsImpl
  - OsciRegistryServTlsImpl, [515](#)
- ~OsciSchedulerObserver
  - OsciSchedulerObserver, [517](#)
- ~OsciScopedLock
  - OsciScopedLock, [518](#)
- ~OsciSemaphore
  - OsciSemaphore, [521](#)
- ~OsciSendMethod
  - OsciSendMethod, [523](#)
- ~OsciSendToMethod
  - OsciSendToMethod, [525](#)
- ~OsciSharedPtr
  - OsciSharedPtr, [528](#)
- ~OsciShutdownMethod
  - OsciShutdownMethod, [530](#)
- ~OsciSingleton
  - OsciSingleton, [532](#)
- ~OsciSocketI
  - OsciSocketI, [536](#)
- ~OsciSocketIBase
  - OsciSocketIBase, [541](#)
- ~OsciSocketMethod
  - OsciSocketMethod, [546](#)
- ~OsciSocketObserver
  - OsciSocketObserver, [548](#)
- ~OsciSocketRequestAO
  - OsciSocketRequestAO, [551](#)
- ~OsciSocketServ
  - OsciSocketServ, [554](#)
- ~OsciSocketServIBase
  - OsciSocketServIBase, [559](#)
- ~OsciTCPSocket
  - OsciTCPSocket, [566](#)
- ~OsciTCPSocketI
  - OsciTCPSocketI, [573](#)
- ~OsciTLS
  - OsciTLS, [592](#)
- ~OsciTLSEx
  - OsciTLSEx, [594](#)
- ~OsciThread
  - OsciThread, [575](#)
- ~OsciThreadLock
  - OsciThreadLock, [579](#)
- ~OsciTimer
  - OsciTimer, [583](#)
- ~OsciTimerObject

- OscTimerObject, [587](#)
- ~OscTimerObserver
  - OscTimerObserver, [590](#)
- ~OscUDPSocket
  - OscUDPSocket, [602](#)
- ~OscUDPSocketI
  - OscUDPSocketI, [608](#)
- ~Osc\_Alloc
  - Osc\_Alloc, [170](#)
- ~Osc\_Dealloc
  - Osc\_Dealloc, [171](#)
- ~Osc\_File
  - Osc\_File, [181](#)
- ~Osc\_FileFind
  - Osc\_FileFind, [190](#)
- ~Osc\_FileServer
  - Osc\_FileServer, [193](#)
- ~Osc\_Linked\_List
  - Osc\_Linked\_List, [206](#)
- ~Osc\_Linked\_List\_Base
  - Osc\_Linked\_List\_Base, [212](#)
- ~Osc\_MTLinked\_List
  - Osc\_MTLinked\_List, [225](#)
- ~Osc\_Opaque\_Type\_Alloc
  - Osc\_Opaque\_Type\_Alloc, [229](#)
- ~Osc\_Opaque\_Type\_Alloc\_LL
  - Osc\_Opaque\_Type\_Alloc\_LL, [231](#)
- ~Osc\_Opaque\_Type\_Compare
  - Osc\_Opaque\_Type\_Compare, [233](#)
- ~Osc\_Queue
  - Osc\_Queue, [237](#)
- ~Osc\_Queue\_Base
  - Osc\_Queue\_Base, [239](#)
- ~Osc\_Rb\_Tree
  - Osc\_Rb\_Tree, [244](#)
- ~Osc\_TAlloc
  - Osc\_TAlloc, [282](#)
- ~Osc\_Tag
  - Osc\_Tag, [265](#)
- ~Osc\_TagTree
  - Osc\_TagTree, [270](#)
- ~Osc\_Vector
  - Osc\_Vector, [286](#)
- ~Osc\_Vector\_Base
  - Osc\_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](#)
- ~\_OscBasicAllocator
  - \_OscBasicAllocator, [109](#)
- ~\_OscHeapBase
  - \_OscHeapBase, [111](#)
- \_OSCL\_Abort
  - osclbase, [36](#)
- \_OSCL\_CLEANUP\_BASE\_CLASS
  - osclmemory, [50](#)
- \_OSCL\_TRAP\_NEW
  - osclmemory, [50](#)
- \_OscBasicAllocator, [108](#)
- \_OscBasicAllocator
  - ~\_OscBasicAllocator, [109](#)
  - allocate, [109](#)
  - deallocate, [109](#)
- \_OscHeapBase, [110](#)
- \_OscHeapBase, [111](#)
- \_OscHeapBase
  - ~\_OscHeapBase, [111](#)
  - \_OscHeapBase, [111](#)
  - PVCleanupStack, [111](#)
- \_OscInteger64Transport
  - 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
  - OscExclusiveArrayPtr, [382](#)
  - OscExclusivePtr, [385](#)
  - OscExclusivePtrA, [388](#)

- OscSingleton, [533](#)
- OscTLS, [593](#)
- OscTLSEx, [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\_\_TOscConditionObject\_\_defined\_\_
  - osclconfig\_proc\_check.h, [831](#)
- \_\_verify\_\_TOscFileOffset\_\_defined\_\_
  - osclconfig\_io\_check.h, [822](#)
- \_\_verify\_\_TOscMutexObject\_\_defined\_\_
  - osclconfig\_proc\_check.h, [831](#)
- \_\_verify\_\_TOscSemaphoreObject\_\_defined\_\_
  - osclconfig\_proc\_check.h, [831](#)
- \_\_verify\_\_TOscThreadFuncArg\_\_defined\_\_
  - osclconfig\_proc\_check.h, [831](#)
- \_\_verify\_\_TOscThreadFuncRet\_\_defined\_\_
  - osclconfig\_proc\_check.h, [831](#)
- \_\_verify\_\_TOscThreadId\_\_defined\_\_
  - osclconfig\_proc\_check.h, [831](#)
- \_\_verify\_\_TOscThreadObject\_\_defined\_\_
  - osclconfig\_proc\_check.h, [831](#)
- \_fixedCaches
  - OscFileCache, [402](#)
- \_movableCache
  - OscFileCache, [402](#)
- \_oscl\_audit\_calloc
  - osclmemory, [59](#)
- \_oscl\_audit\_free
  - osclmemory, [59](#)
- \_oscl\_audit\_malloc
  - osclmemory, [59](#)
- \_oscl\_audit\_new
  - osclmemory, [59](#)
- \_oscl\_audit\_realloc
  - osclmemory, [60](#)
- \_oscl\_calloc
  - osclmemory, [60](#)
- \_oscl\_default\_audit\_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
  - OscDNSMethod, [359](#)
  - OscDNSRequestAO, [364](#)
  - OscSocketMethod, [546](#)
  - OscSocketRequestAO, [551](#)
- AbortAll
  - OscDNSMethod, [359](#)
  - OscSocketMethod, [546](#)
- Accept
  - OscAcceptMethod, [308](#)
  - OscAcceptRequest, [309](#)
  - OscSocketI, [536](#)
  - OscSocketIBase, [541](#)
  - OscTCPSocket, [566](#)
  - OscTCPSocketI, [573](#)
- AcceptParam, [112](#)
  - AcceptParam, [112](#)
- AcceptParam
  - AcceptParam, [112](#)
  - iBlankSocket, [112](#)
- AcceptRequest
  - OscAcceptMethod, [308](#)
- Activate
  - OscDNSRequest, [362](#)
  - OscSocketRequest, [549](#)
  - PVActiveBase, [613](#)
- Add
  - OscSocketServRequestList, [560](#)
  - OscTimerQ, [591](#)

- add\_element
  - OscL\_Linked\_List, 207
  - OscL\_Linked\_List\_Base, 212
  - OscL\_MTLLinked\_List, 226
- add\_ref
  - CHepRep, 130
- add\_to\_front
  - OscL\_Linked\_List, 207
  - OscL\_Linked\_List\_Base, 212
  - OscL\_MTLLinked\_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\_DefAllocWithRefCount, 173
  - OscLMemPoolFixedChunkAllocator, 444
  - OscLMemPoolResizableAllocator, 449
  - OscLRefCount, 492
  - OscLRefCountDA, 495
  - OscLRefCountMTDA, 499
  - OscLRefCountMTSA, 501
  - OscLRefCountSA, 503
- address
  - OscL\_TAlloc, 282
- addressListCapacity
  - GetHostByNameParam, 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](#)
- CHeapRep, [130](#)
- CStackRep, [132](#)
- APPEND\_MEDIA\_AT\_END
  - osclutil, [84](#)
- append\_rep
  - CHeapRep, [130](#)
  - OSCL\_String, [261](#)
  - OSCL\_wString, [305](#)
- AppendBuffers
  - PVLoggerAppender, [623](#)
- AppendNext
  - BufFragGroup, [121](#)
- AppendString
  - PVLoggerAppender, [623](#)
- assign
  - CHeapRep, [130](#)
- assign\_vector
  - OscI\_Vector\_Base, [291](#)
- asyncfilereadcancel\_test
  - OscI\_File, [186](#)
- asyncfilereadwrite\_test
  - OscI\_File, [186](#)
- Attach
  - OscIBinStream, [337](#)
- audit\_type
  - OscIMemGlobalAuditObject, [441](#)
- available\_localbuf
  - MediaData, [144](#)
- back
  - OscI\_Queue, [237](#)
  - OscI\_Vector, [287](#)
- BAD\_THREADID\_ADDR\_ERROR
  - OscIProcStatus, [475](#)
- base\_link\_type
  - OscI\_Rb\_Tree\_Base, [246](#)
  - OscI\_Rb\_Tree\_Const\_Iterator, [248](#)
  - OscI\_Rb\_Tree\_Iterator, [251](#)
  - OscI\_Rb\_Tree\_Node\_Base, [254](#)
- begin
  - OscI\_Map, [219](#)
  - OscI\_Rb\_Tree, [244](#)
  - OscI\_TagTree, [270](#)
  - OscI\_Vector, [287](#)
- BeginScheduling
  - OscIExecSchedulerCommonBase, [395](#)
- BeginStats
  - OscIExecSchedulerCommonBase, [395](#)
- BFG\_SUCCESS
  - BufFragStatusClass, [123](#)
- big\_endian\_to\_host
  - osclbase, [36](#)
- Bind
  - osclbase, [36](#)
  - OscIBindMethod, [322](#)
  - OscIBindRequest, [323](#)
  - OscIIPSocketI, [419](#)
  - OscISocketI, [536](#)
  - OscISocketIBase, [541](#)
  - OscITCPSocket, [566](#)
  - OscIUDPSocket, [602](#)
- bind
  - BufferState, [119](#)
- BindAsync
  - OscISocketIBase, [541](#)
  - OscITCPSocket, [566](#)
  - OscITCPSocketI, [573](#)
  - OscIUDPSocket, [602](#)
  - OscIUDPSocketI, [608](#)
- BindParam, [116](#)
  - BindParam, [116](#)
- BindParam
  - BindParam, [116](#)
  - iAddr, [116](#)
- BindRequest
  - OscIBindMethod, [322](#)
- black
  - OscI\_Rb\_Tree\_Node\_Base, [254](#)
- BlockingLoopL
  - OscIExecSchedulerCommonBase, [395](#)
- bSetFailure
  - MM\_AllocInfo, [150](#)
- Buffer
  - OscIAsyncFileBuffer, [320](#)
- buffer
  - CFastRep, [128](#)
  - CHeapRep, [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
- BufFragGroup, 120
  - BufFragGroup, 121
- BufFragGroup
  - ~BufFragGroup, 121
  - AddFragment, 121
  - AppendNext, 121
  - buffer\_states, 122
  - BufFragGroup, 121
  - Clear, 121
  - fragments, 122
  - GetLength, 121
  - GetMaxFrag, 122
  - GetNext, 122
  - GetNumFrag, 122
  - length, 122
  - next, 122
  - num\_fragments, 122
- BufFragStatusClass, 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
- BufFragStatusClass
  - status\_t, 123
- bufsize
  - Osl\_Queue\_Base, 241
  - Osl\_Vector\_Base, 293
- BYTES\_IN\_UUID\_ARRAY
  - osl\_uuid.h, 799
- c
  - OslPriorityQueue, 473
- c\_bool
  - osclbase, 34
- c\_str
  - StrPtrLen, 648
  - WStrPtrLen, 659
- Callback
  - OslReadyQ, 485
- callback\_timer\_type
  - OslTimer, 583
- CallbackTimer, 124
  - CallbackTimer, 124
- CallbackTimer
  - ~CallbackTimer, 124
  - CallbackTimer, 124
- Run, 124
- CallbackTimer< Alloc >
  - OslTimer, 584
- CallbackTimerObserver, 126
- CallbackTimerObserver
  - ~CallbackTimerObserver, 126
  - TimerBaseElapsed, 126
- CallRunExec
  - OslExecSchedulerCommonBase, 395
- Cancel
  - OslActiveObject, 311
  - OslTimer, 583
  - OslTimerObject, 587
  - PVActiveBase, 613
- CancelAccept
  - OslSocketIBase, 542
  - OslTCPSocket, 567
  - OslTCPSocketI, 573
- CancelBind
  - OslSocketIBase, 542
  - OslTCPSocket, 567
  - OslTCPSocketI, 573
  - OslUDPSocket, 602
  - OslUDPSocketI, 608
- CancelConnect
  - OslSocketIBase, 542
  - OslTCPSocket, 567
  - OslTCPSocketI, 573
- CancelFreeChunkAvailableCallback
  - OslMemPoolFixedChunkAllocator, 444
  - OslMemPoolResizableAllocator, 450
- CancelFreeMemoryAvailableCallback
  - OslMemPoolResizableAllocator, 450
- CancelFxn
  - OslDNSIBase, 356
  - OslSocketIBase, 542
- CancelGetHostByName
  - OslDNS, 351
  - OslDNSIBase, 356
- Cancelled
  - OslDNSRequestAO, 364
- CancelListen
  - OslSocketIBase, 542
  - OslTCPSocket, 567
  - OslTCPSocketI, 573
- CancelMethod
  - OslDNSMethod, 359
  - OslSocketMethod, 546
- CancelRecv
  - OslSocketIBase, 542
  - OslTCPSocket, 567
  - OslTCPSocketI, 573
- CancelRecvFrom
  - OslSocketIBase, 542

- OscUDPSocket, [602](#)
- OscUDPSocketI, [608](#)
- CancelRequest
  - OscDNSRequest, [362](#)
  - OscSocketRequest, [549](#)
- CancelSend
  - OscSocketIBase, [542](#)
  - OscTCPSocket, [567](#)
  - OscTCPSocketI, [573](#)
- CancelSendTo
  - OscSocketIBase, [542](#)
  - OscUDPSocket, [602](#)
  - OscUDPSocketI, [608](#)
- CancelShutdown
  - OscSocketIBase, [542](#)
  - OscTCPSocket, [567](#)
  - OscTCPSocketI, [573](#)
- canPersistMoreHostAddresses
  - GetHostByNameParam, [136](#)
- CanTerminate
  - OscThread, [575](#)
- capacity
  - Osc\_Queue\_Base, [240](#)
  - Osc\_Vector\_Base, [291](#)
  - OscFileCacheBuffer, [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](#)
- CHepRep, [129](#)
  - CHepRep, [130](#)
- CHepRep
  - add\_ref, [130](#)
  - append, [130](#)
  - append\_rep, [130](#)
  - assign, [130](#)
  - buffer, [130](#)
  - CHepRep, [130](#)
  - maxsize, [130](#)
  - refcount, [130](#)
  - remove\_ref, [130](#)
  - set, [130](#)
  - set\_rep, [130](#)
  - size, [130](#)
- check\_fence
  - MM\_AllocBlockFence, [147](#)
- check\_list
  - Osc\_Linked\_List, [207](#)
  - Osc\_Linked\_List\_Base, [212](#)
- checkSum
  - StrCSumPtrLen, [645](#)
- ChecksumType
  - StrCSumPtrLen, [645](#)
- children
  - Osc\_TagTree::Node, [280](#)
- children\_type
  - Osc\_TagTree, [270](#)
  - Osc\_TagTree::Node, [280](#)
- ChooseCurCache
  - Osc\_File::OscCacheObserver, [187](#)
- CleanInUse
  - OscAsyncFileBuffer, [320](#)
- Cleanup
  - OscErrorTrap, [376](#)
  - OscInit, [415](#)
  - OscMem, [425](#)
  - OscScheduler, [516](#)
  - PVLogger, [619](#)
- CleanupExecQ
  - OscExecSchedulerCommonBase, [395](#)
- CleanupParam
  - OscSocketRequestAO, [551](#)
- CleanupStatQ
  - OscExecSchedulerCommonBase, [395](#)
- Clear
  - BufFragGroup, [121](#)
  - MediaData, [143](#)
  - OscTimer, [583](#)
- clear
  - Osc\_Linked\_List, [207](#)
  - Osc\_Map, [219](#)
  - Osc\_Queue, [237](#)
  - Osc\_Queue\_Base, [240](#)
  - Osc\_Rb\_Tree, [244](#)
  - Osc\_TagTree, [271](#)
  - Osc\_Vector, [287](#)
- ClearTOS
  - OscSocketTOS, [564](#)
- Close

- OscI\_File, 181
- OscI\_FileFind, 190
- OscI\_FileServer, 193
- OscIAsyncFile, 317
- OscIDNSI, 353
- OscIDNSIBase, 356
- OscIFileCache, 402
- OscIIPSocketI, 419
- OscIMutex, 459
- OscINativeFile, 463
- OscIRegistryAccessClient, 504
- OscIRegistryClient, 509
- OscIRegistryClientImpl, 512
- OscIRegistryServTlsImpl, 515
- OscISemaphore, 521
- OscISocketI, 536
- OscISocketIBase, 542
- OscISocketServ, 554
- OscISocketServI, 556
- OscISocketServIBase, 559
- OscISocketServRequestList, 560
- OscITCPSocket, 568
- OscITCPSocketI, 573
- OscIUDPSocket, 603
- OscIUDPSocketI, 608
- CloseSession
  - OscIComponentRegistry, 344
- color
  - OscI\_Rb\_Tree\_Node\_Base, 255
- color\_type
  - OscI\_Rb\_Tree\_Node\_Base, 254
- comp
  - OscI\_Map::value\_compare, 223
  - OscIPriorityQueue, 473
- compare
  - OscICompareLess, 342
  - OscIReadyCompare, 483
  - OscITimerCompare, 585
- compare\_data
  - OscI\_Opaque\_Type\_Alloc\_LL, 231
- compare\_EQ
  - OscI\_Opaque\_Type\_Compare, 233
  - OscIPriorityQueue, 471
- compare\_LT
  - OscI\_Opaque\_Type\_Compare, 233
  - OscIPriorityQueue, 471
- CompareId
  - OscIThread, 576
- Complete
  - OscIDNSRequest, 362
  - OscISocketRequest, 549
- COMPUTE\_MEM\_ALIGN\_SIZE
  - osclmemory, 51
- Connect
  - OscI\_FileServer, 193
  - OscIConnectMethod, 348
  - OscIConnectRequest, 349
  - OscIRegistryAccessClient, 504
  - OscIRegistryClient, 509
  - OscIRegistryClientImpl, 512
  - OscIRegistryServTlsImpl, 515
  - OscISocketI, 536
  - OscISocketIBase, 542
  - OscISocketServ, 554
  - OscISocketServI, 556
  - OscISocketServIBase, 559
  - OscITCPSocket, 568
  - OscITCPSocketI, 573
- ConnectParam, 131
  - ConnectParam, 131
- ConnectParam
  - ConnectParam, 131
  - iAddr, 131
- ConnectRequest
  - OscIConnectMethod, 348
- const\_iterator
  - OscI\_Map, 218
  - OscI\_Rb\_Tree, 244
  - OscI\_Rb\_Tree\_Const\_Iterator, 248
  - OscI\_TagTree::const\_iterator, 274
  - OscI\_Vector, 286
- const\_pointer
  - OscI\_Rb\_Tree, 244
  - OscI\_TAlloc, 282
- const\_reference
  - OscI\_Map, 218
  - OscI\_Queue, 237
  - OscI\_Rb\_Tree, 244
  - OscI\_TAlloc, 282
  - OscI\_Vector, 286
  - OscIPriorityQueue, 471
- Construct
  - OscIReadyQ, 485
  - OscITimerQ, 591
- construct
  - OscI\_Linked\_List\_Base, 212
  - OscI\_Opaque\_Type\_Alloc, 229
  - OscI\_Opaque\_Type\_Alloc\_LL, 231
  - OscI\_Queue\_Base, 240
  - OscI\_TAlloc, 282
  - OscI\_Vector\_Base, 291
  - OscIPriorityQueueBase, 474
- ConstructL
  - OscIDNSMethod, 359
  - OscIDNSRequestAO, 364
  - OscIExecSchedulerCommonBase, 395
  - OscIIPSocketI, 419
  - OscISocketMethod, 546

- OscSocketRequestAO, 551
- ConstructStatQ
  - OscExecSchedulerCommonBase, 395
- container\_type
  - OscPriorityQueue, 471
- Contains
  - Osc\_File::OscFixedCacheParam, 188
  - OscFileCacheBuffer, 404
- count
  - Osc\_Map, 219
  - Osc\_Rb\_Tree, 244
  - Osc\_TagTree, 271
- CPVInterfaceProxy
  - OscErrorTrapImp, 378
- Create
  - GetHostByNameParam, 136
  - OscMutex, 459
  - OscSemaphore, 521
  - OscThread, 576
- createmempool
  - OscMemPoolFixedChunkAllocator, 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
  - OscExecScheduler, 389
- currentPos
  - OscFileCacheBuffer, 404
- data
  - LinkedListElement, 140
- data1
  - OscUuid, 611
- data2
  - OscUuid, 611
- data3
  - OscUuid, 611
- data4
  - OscUuid, 611
- deallocate
  - \_OscBasicAllocator, 109
  - MemAllocator, 146
  - Osc\_Dealloc, 171
  - Osc\_DefAlloc, 172
  - Osc\_Opaque\_Type\_Alloc, 229
  - Osc\_Opaque\_Type\_Alloc\_LL, 231
  - Osc\_TAlloc, 282
  - OscErrorAllocator, 374
  - OscMemAllocator, 426
  - OscMemAllocDestructDealloc, 427
  - OSCLMemAutoPtr, 437
  - OscMemBasicAllocator, 439
  - OscMemBasicAllocDestructDealloc, 440
  - OscMemPoolFixedChunkAllocator, 445
  - OscMemPoolResizableAllocator, 450
  - OscReadyAlloc, 482
- deallocateblock
  - OscMemPoolResizableAllocator, 450
- decrement\_refcnt
  - BufferState, 119
- DEFAULT\_MM\_AUDIT\_MODE
  - osclmemory, 52
- DEFAULT\_POSTFILL\_PATTERN
  - osclmemory, 52
- DEFAULT\_PREFILL\_PATTERN
  - osclmemory, 52
- Delete
  - Osc\_DefAllocWithRefCounter, 173
  - OscAsyncFile, 317
  - OscBuf, 341
- Depth
  - OscReadyQ, 485
- depth
  - Osc\_TagTree::Node, 280
- dequeue\_element
  - Osc\_Linked\_List, 207
  - Osc\_MTLlinked\_List, 226
- Des
  - OscBuf, 341
- DesC
  - OscBuf, 341
- Destroy
  - DNSRequestParam, 133
  - GetHostByNameParam, 136
  - PVActiveBase, 613
- destroy
  - Osc\_Linked\_List\_Base, 212
  - Osc\_Opaque\_Type\_Alloc, 229
  - Osc\_Opaque\_Type\_Alloc\_LL, 232
  - Osc\_Queue\_Base, 240
  - Osc\_TAlloc, 282
  - Osc\_Vector, 287
  - Osc\_Vector\_Base, 291
- destroyallmempoolbuffers

- OscMemPoolResizableAllocator, [450](#)
- destroymempool
  - OscMemPoolFixedChunkAllocator, [445](#)
- destruct\_and\_dealloc
  - Osc\_TAlloc, [282](#)
  - OscDestructDealloc, [350](#)
  - OscMemAllocDestructDealloc, [427](#)
  - OscMemBasicAllocDestructDealloc, [440](#)
- difference\_type
  - Osc\_Rb\_Tree, [244](#)
- DIR\_TYPE
  - Osc\_FileFind, [189](#)
- DisableAppenderInheritance
  - PVLogger, [619](#)
- DiscardAcceptedSocket
  - OscAcceptMethod, [308](#)
- DNSRequestParam, [133](#)
  - DNSRequestParam, [133](#)
  - OscDNSI, [354](#)
- DNSRequestParam
  - ~DNSRequestParam, [133](#)
  - Destroy, [133](#)
  - DNSRequestParam, [133](#)
  - iDNSRequest, [134](#)
  - iFxn, [134](#)
  - InThread, [133](#)
  - iRefCount, [134](#)
  - RemoveRef, [134](#)
- DoCancel
  - OscActiveObject, [312](#)
  - OscDNSRequestAO, [364](#)
  - OscSocketRequestAO, [551](#)
  - OscTimerObject, [587](#)
  - PVActiveBase, [613](#)
- E\_BUFFER\_TOO\_SMALL
  - Osc\_FileFind, [190](#)
- E\_INVALID\_ARG
  - Osc\_FileFind, [189](#)
- E\_INVALID\_STATE
  - Osc\_FileFind, [189](#)
- E\_MEMORY\_ERROR
  - Osc\_FileFind, [190](#)
- E\_NO\_MATCH
  - Osc\_FileFind, [190](#)
- E\_NOT\_IMPLEMENTED
  - Osc\_FileFind, [190](#)
- E\_OK
  - Osc\_FileFind, [189](#)
- E\_OTHER
  - Osc\_FileFind, [190](#)
- E\_PATH\_NOT\_FOUND
  - Osc\_FileFind, [189](#)
- E\_PATH\_TOO\_LONG
  - Osc\_FileFind, [189](#)
- element\_type
  - Osc\_FileFind, [189](#)
- elems
  - Osc\_Queue\_Base, [241](#)
  - Osc\_Vector\_Base, [293](#)
- empty
  - Osc\_Map, [219](#)
  - Osc\_Queue\_Base, [240](#)
  - Osc\_Rb\_Tree, [244](#)
  - Osc\_TagTree, [271](#)
  - Osc\_Vector\_Base, [291](#)
  - OscPriorityQueue, [472](#)
- EMPTY\_FRAGMENT
  - BufFragStatusClass, [123](#)
- EMPTY\_UUID
  - oscl\_uuid.h, [799](#)
- enablenullpointerreturn
  - OscMemPoolFixedChunkAllocator, [445](#)
  - OscMemPoolResizableAllocator, [450](#)
- End
  - OscFileStats, [411](#)
- end
  - Osc\_Map, [219](#)
  - Osc\_Rb\_Tree, [244](#)
  - Osc\_TagTree, [271](#)
  - Osc\_Vector, [287](#)
- EndOfFile
  - Osc\_File, [181](#)
  - OscAsyncFile, [317](#)
  - OscFileCache, [402](#)
  - OscNativeFile, [463](#)
- endPos
  - OscFileCacheBuffer, [404](#)
- EndScheduling
  - OscExecSchedulerCommonBase, [395](#)
- EndStats
  - OscExecSchedulerCommonBase, [395](#)
- EnterThreadContext
  - PVThreadContext, [634](#)
- eof
  - OscBinStream, [337](#)
- EOF\_STATE
  - OscBinStream, [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](#)

- EOsclFileOp\_EndOfFile
  - osclio, [98](#)
- EOsclFileOp\_Flush
  - osclio, [98](#)
- EOsclFileOp\_Last
  - osclio, [99](#)
- EOsclFileOp\_NativeClose
  - osclio, [98](#)
- EOsclFileOp\_NativeEndOfFile
  - osclio, [99](#)
- EOsclFileOp\_NativeFlush
  - osclio, [99](#)
- EOsclFileOp\_NativeOpen
  - osclio, [98](#)
- EOsclFileOp\_NativeRead
  - osclio, [98](#)
- EOsclFileOp\_NativeSeek
  - osclio, [99](#)
- EOsclFileOp\_NativeSetSize
  - osclio, [99](#)
- EOsclFileOp\_NativeSize
  - osclio, [99](#)
- EOsclFileOp\_NativeTell
  - osclio, [99](#)
- EOsclFileOp\_NativeWrite
  - osclio, [99](#)
- EOsclFileOp\_Open
  - osclio, [98](#)
- EOsclFileOp\_Read
  - osclio, [98](#)
- EOsclFileOp\_Seek
  - osclio, [98](#)
- EOsclFileOp\_SetSize
  - osclio, [98](#)
- EOsclFileOp\_Size
  - osclio, [98](#)
- EOsclFileOp\_Tell
  - osclio, [98](#)
- EOsclFileOp\_Write
  - osclio, [98](#)
- eOscProcError
  - OscProcStatus, [475](#)
- EOsclSocket\_DataRecv
  - oscl\_socket\_stats.h, [771](#)
- EOsclSocket\_DataSent
  - oscl\_socket\_stats.h, [771](#)
- EOsclSocket\_Except
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocket\_OS
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocket\_Readable
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocket\_RequestAO\_Canceled
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocket\_RequestAO\_Error
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocket\_RequestAO\_Success
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocket\_RequestAO\_Timeout
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocket\_ServPoll
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocket\_ServRequestCancelIssued
  - oscl\_socket\_stats.h, [771](#)
- EOsclSocket\_ServRequestComplete
  - oscl\_socket\_stats.h, [771](#)
- EOsclSocket\_ServRequestIssued
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocket\_Writable
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocketServ\_LastEvent
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocketServ\_LoopsockError
  - oscl\_socket\_stats.h, [771](#)
- EOsclSocketServ\_LoopsockOk
  - oscl\_socket\_stats.h, [771](#)
- EOsclSocketServ\_SelectActivity
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocketServ\_SelectNoActivity
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocketServ\_SelectRescheduleAsap
  - oscl\_socket\_stats.h, [770](#)
- EOsclSocketServ\_SelectReschedulePoll
  - oscl\_socket\_stats.h, [770](#)
- EOsclThreadTerminate\_Join
  - oscl\_thread.h, [789](#)
- EOsclThreadTerminate\_Kill
  - oscl\_thread.h, [789](#)
- EOsclThreadTerminate\_NOP
  - oscl\_thread.h, [789](#)
- EOtherExecStats\_Last
  - OscExecSchedulerCommonBase, [394](#)
- EOtherExecStats\_NativeOS
  - OscExecSchedulerCommonBase, [394](#)
- EOtherExecStats\_QueueTime
  - OscExecSchedulerCommonBase, [394](#)
- EOtherExecStats\_ReleaseTime
  - OscExecSchedulerCommonBase, [394](#)
- EOtherExecStats\_WaitTime
  - OscExecSchedulerCommonBase, [394](#)
- EPriorityHigh
  - OscActiveObject, [311](#)
- EPriorityHighest
  - OscActiveObject, [311](#)
- EPriorityIdle
  - OscActiveObject, [311](#)
- EPriorityLow
  - OscActiveObject, [311](#)



EPriorityNominal  
     OsciActiveObject, [311](#)  
 EPV\_ARM\_GNUC  
     osclbase, [32](#)  
 EPV\_ARM\_MSEVC  
     osclbase, [32](#)  
 EPV\_ARM\_RVCT  
     osclbase, [32](#)  
 EPVCritic\_Ecp  
     OsciSocketTOS, [563](#)  
 EPVDNSCancel  
     osclio, [99](#)  
 EPVDNSFailure  
     osclio, [99](#)  
 EPVDNSGetHostByName  
     osclio, [99](#)  
 EPVDNSPending  
     osclio, [99](#)  
 EPVDNSSuccess  
     osclio, [99](#)  
 EPVDNSTimeout  
     osclio, [99](#)  
 EPVFlash  
     OsciSocketTOS, [563](#)  
 EPVHiRel  
     OsciSocketTOS, [563](#)  
 EPVHiThrpt  
     OsciSocketTOS, [563](#)  
 EPVImmediate  
     OsciSocketTOS, [563](#)  
 EPVInetControl  
     OsciSocketTOS, [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  
     OsciSocketTOS, [563](#)  
 EPVNetControl  
     OsciSocketTOS, [563](#)  
 EPVNoTOS  
     OsciSocketTOS, [563](#)  
 EPVOverrideFlash  
     OsciSocketTOS, [563](#)  
 EPVPriority  
     OsciSocketTOS, [563](#)  
 EPVRoutine  
     OsciSocketTOS, [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\_NonOsciThread  
     osclproc, [106](#)  
 EPVThreadContext\_OsciThread  
     osclproc, [106](#)  
 EPVThreadContext\_Undetermined  
     osclproc, [106](#)  
 equal\_range  
     Osci\_Map, [219](#)  
     Osci\_Rb\_Tree, [244](#)  
 erase



- OscI\_Map, 220
- OscI\_Rb\_Tree, 244
- OscI\_TagTree, 271
- OscI\_Vector, 287
- OscI\_Vector\_Base, 291, 292
- Error
  - OscIExecSchedulerCommonBase, 395
- error\_type
  - OscI\_FileFind, 189
- ESocketServ\_Connected
  - OscISocketServIBase, 558
- ESocketServ\_Error
  - OscISocketServIBase, 559
- ESocketServ\_Idle
  - OscISocketServIBase, 558
- ESymbianAccessMode\_Rfile
  - OscI\_File, 180
- ESymbianAccessMode\_RfileBuf
  - OscI\_File, 180
- EXCEED\_MAX\_COUNT\_VARIABLE\_ - ERROR
  - OscIProcStatus, 476
- EXCEED\_MAX\_SEM\_COUNT\_ERROR
  - OscIProcStatus, 476
- Exit
  - OscIThread, 576
- ExitThreadContext
  - PVThreadContext, 634
- extract\_string
  - osclutil, 70
- fail
  - OscIBinStream, 338
- FAIL\_STATE
  - OscIBinStream, 337
- Failure
  - OscIDNSRequestAO, 364
- FENCE\_PATTERN
  - osclmemory, 52
- FILE\_TYPE
  - OscI\_FileFind, 189
- fileName
  - MM\_AllocQueryInfo, 152
- filePosition
  - OscIFileCacheBuffer, 404
- FileSize
  - OscIFileCache, 402
- fill\_fence
  - MM\_AllocBlockFence, 147
- FillFromFile
  - OscIFileCacheBuffer, 404
- filter\_status\_type
  - AllPassFilter, 114
  - PVLogger, 618
- PVLoggerFilter, 624
- FilterOpaqueMessge
  - AllPassFilter, 115
  - PVLoggerFilter, 625
- FilterString
  - AllPassFilter, 115
  - PVLoggerFilter, 625
- Find
  - OscIComponentRegistryData, 345
- find
  - OscI\_Map, 220
  - OscI\_Rb\_Tree, 244
  - OscI\_TagTree, 271
- find\_heap
  - OscIPriorityQueue, 472
  - OscIPriorityQueueBase, 474
- FindExact
  - OscIComponentRegistry, 344
- FindFirst
  - OscI\_FileFind, 190
- findfreeblock
  - OscIMemPoolResizableAllocator, 451
- FindHierarchical
  - OscIComponentRegistry, 344
- FindNext
  - OscI\_FileFind, 191
- FindPVBase
  - OscIExecSchedulerCommonBase, 395
- first
  - OscI\_Pair, 235
- firstFragPtr
  - OscIBinStream, 339
- FIXED\_FRAG\_LOC\_FULL
  - BufFragStatusClass, 123
- Flush
  - OscI\_File, 182
  - OscIAsyncFile, 317
  - OscIFileCache, 402
  - OscINativeFile, 463
- FormatOpaqueMessage
  - PVLoggerLayout, 626
- FormatString
  - PVLoggerLayout, 626
- fragments
  - BufFragGroup, 122
- fragsLeft
  - OscIBinStream, 339
- freeblockavailable
  - OscIMemPoolResizableAllocatorObserver, 457
- freebytes
  - oscl\_fsstat, 195
- freechunkavailable

- OscMemPoolFixedChunkAllocator-
  - Observer, [447](#)
- freememoryavailable
  - OscMemPoolResizableAllocatorMemory-
    - Observer, [456](#)
- front
  - Osc\_Queue, [238](#)
  - Osc\_Vector, [288](#)
- Fxn
  - OscSocketRequest, [549](#)
- get
  - OscBinIStream, [324](#)
  - OscExclusiveArrayPtr, [381](#)
  - OscExclusivePtr, [384](#)
  - OscExclusivePtrA, [387](#)
  - OSCLMemAutoPtr, [437](#)
- get\_buf\_mgr
  - BufferState, [119](#)
- get\_count
  - OscSharedPtr, [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
  - Osc\_Opaque\_Type\_Alloc\_LL, [232](#)
- get\_element
  - Osc\_Linked\_List, [207](#)
  - Osc\_Linked\_List\_Base, [212](#)
  - Osc\_MTLinked\_List, [226](#)
- get\_first
  - Osc\_Linked\_List, [208](#)
  - Osc\_Linked\_List\_Base, [213](#)
- get\_free\_function
  - BufferState, [119](#)
- get\_index
  - Osc\_Linked\_List, [208](#)
  - Osc\_Linked\_List\_Base, [213](#)
  - Osc\_MTLinked\_List, [226](#)
- get\_int64\_lower32
  - Osc\_Int64\_Utils, [204](#)
- get\_int64\_middle32
  - Osc\_Int64\_Utils, [204](#)
- get\_int64\_upper32
  - Osc\_Int64\_Utils, [204](#)
- get\_ISO8601\_str\_time
  - TimeValue, [652](#)
- get\_local\_time
  - TimeValue, [652](#)
- get\_lower32
  - NTPTTime, [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
  - NTPTTime, [168](#)
- get\_next
  - Osc\_Linked\_List, [208](#)
  - Osc\_Linked\_List\_Base, [213](#)
  - Osc\_Opaque\_Type\_Alloc\_LL, [232](#)
- get\_num\_elements
  - Osc\_Linked\_List, [208](#)
- get\_ptr
  - BufferState, [119](#)
- get\_pv8601\_str\_time
  - TimeValue, [652](#)
- get\_refcount
  - BufferState, [119](#)
- get\_registry
  - TLSSStorageOps, [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
  - Osc\_Int64\_Utils, [204](#)

- get\_uint64\_middle32
  - OscI\_Int64\_Utils, [204](#)
- get\_uint64\_upper32
  - OscI\_Int64\_Utils, [204](#)
- get\_upper32
  - NTPTIME, [168](#)
- get\_usec
  - TimeValue, [653](#)
- get\_value
  - NTPTIME, [168](#)
- GetAcceptedSocket
  - OscI\_AcceptMethod, [308](#)
- GetAcceptedSocketL
  - OscITCPSocket, [568](#)
  - OscITCPSocketI, [573](#)
- getAllocatedSize
  - OscI\_MemPoolResizableAllocator, [451](#)
- getAuditRoot
  - MM\_Audit\_Imp, [154](#)
- GetAvailableBufferSize
  - MediaData, [143](#)
- getAvailableSize
  - OscI\_MemPoolResizableAllocator, [451](#)
- getBufferSize
  - OscI\_MemPoolResizableAllocator, [451](#)
- GetBufferState
  - osclutil, [72](#)
- getCapacity
  - OscI\_RefCounterMemFrag, [497](#)
- getChecksum
  - StrCSumPtrLen, [645](#)
- getCount
  - OscI\_DefAllocWithRefCounter, [173](#)
  - OscI\_RefCounter, [492](#)
  - OscI\_RefCounterDA, [495](#)
  - OscI\_RefCounterMemFrag, [497](#)
  - OscI\_RefCounterMTDA, [499](#)
  - OscI\_RefCounterMTSA, [501](#)
  - OscI\_RefCounterSA, [503](#)
- GetElementType
  - OscI\_FileFind, [191](#)
- GetError
  - OscI\_File, [182](#)
  - OscI\_NativeFile, [463](#)
- GetErrorTrapImp
  - OscI\_ErrorTrap, [376](#)
- GetFactories
  - OscI\_RegistryAccessClient, [504](#)
  - OscI\_RegistryClientImpl, [512](#)
  - OscI\_RegistryServTlsImpl, [515](#)
- GetFactory
  - OscI\_RegistryAccessClient, [504](#)
  - OscI\_RegistryClientImpl, [512](#)
  - OscI\_RegistryServTlsImpl, [515](#)
- GetFragment
  - osclutil, [72](#)
- getGlobalMemAuditObject
  - OscI\_MemGlobalAuditObject, [441](#)
- getHead
  - OscI\_DoubleListBase, [369](#)
- GetHostByName
  - OscI\_DNS, [352](#)
  - OscI\_DNSI, [353](#)
  - OscI\_DNSIBase, [356](#)
  - OscI\_GetHostByNameMethod, [413](#)
- GetHostByNameParam, [135](#)
  - addressListCapacity, [135](#)
  - OscI\_DNSRequestAO, [365](#)
- GetHostByNameParam
  - ~GetHostByNameParam, [136](#)
  - canPersistMoreHostAddresses, [136](#)
  - Create, [136](#)
  - Destroy, [136](#)
  - iAddr, [136](#)
  - iAddressList, [136](#)
  - iName, [136](#)
  - PersistHostAddress, [136](#)
- GetHostByNameResponseContainsAliasInfo
  - OscI\_DNSI, [354](#)
  - OscI\_DNSIBase, [356](#)
- GetHostByNameSuccess
  - OscI\_DNSI, [354](#)
  - OscI\_DNSIBase, [356](#)
- GetId
  - OscI\_ExecSchedulerCommonBase, [395](#)
  - OscI\_Thread, [576](#)
- getInstance
  - OscI\_SingletonRegistry, [534](#)
  - OscI\_TLSRegistry, [596](#)
  - OscI\_TLSRegistryEx, [597](#)
- getLargestContiguousFreeBlockSize
  - OscI\_MemPoolResizableAllocator, [451](#)
- GetLastError
  - OscI\_FileFind, [191](#)
- getLeaveCode
  - OscI\_Exception, [379](#)
- GetLength
  - BufFragGroup, [121](#)
- GetLocalBufsize
  - MediaData, [144](#)
- GetLocalFragment
  - MediaData, [144](#)
- GetLock
  - OscI\_MemAudit, [430](#)
- GetLoggerObject
  - PVLogger, [619](#)
- GetLogLevel
  - PVLogger, [619](#)

- GetMaxFrag
  - BufFragGroup, 122
- GetMediaFragment
  - MediaData, 144
- GetMediaSize
  - MediaData, 144
- getMemFrag
  - OscIRefCounterMemFrag, 497
- getMemFragPtr
  - OscIRefCounterMemFrag, 497
- getMemFragSize
  - OscIRefCounterMemFrag, 497
- getMemPoolBufferAllocatedSize
  - OscIMemPoolResizableAllocator, 451
- getMemPoolBufferSize
  - OscIMemPoolResizableAllocator, 451
- GetName
  - OscIExecSchedulerCommonBase, 395
- GetNext
  - BufFragGroup, 122
- GetNextHost
  - OscIDNSI, 354
  - OscIDNSIBase, 356
- GetNextHostSuccess
  - OscIDNSI, 354
  - OscIDNSIBase, 356
- GetNumAppenders
  - PVLogger, 619
- GetNumFrag
  - BufFragGroup, 122
- GetNumMediaFrag
  - MediaData, 144
- getOffset
  - OscIDoubleListBase, 369
- GetParent
  - PVLogger, 620
- GetPeerName
  - OscIIPSocketI, 419
  - OscISocketI, 536
  - OscITCPSocket, 568
  - OscIUDPSocket, 603
- GetPriority
  - OscIThread, 577
- GetPVLoggerObject
  - PVLoggerRegistry, 629
- GetPVLoggerRegistry
  - PVLoggerRegistry, 629
- GetReadAsyncNumElements
  - OscINativeFile, 463
- GetRecvData
  - OscIIPSocketI, 419
  - OscIRecvFromMethod, 486
  - OscIRecvFromRequest, 488
  - OscIRecvMethod, 490
  - OscIRecvRequest, 491
  - OscITCPSocket, 569
  - OscITCPSocketI, 573
  - OscIUDPSocket, 603
  - OscIUDPSocketI, 608
- GetRefCounter
  - OscISharedPtr, 528
- getRefCounter
  - OscIRefCounterMemFrag, 497
- GetRep
  - OscISharedPtr, 528
- GetScheduler
  - OscIExecSchedulerCommonBase, 395
- GetSendData
  - OscIIPSocketI, 419
  - OscISendMethod, 523
  - OscISendRequest, 524
  - OscISendToMethod, 525
  - OscISendToRequest, 526
  - OscITCPSocket, 569
  - OscITCPSocketI, 573
  - OscIUDPSocket, 603
  - OscIUDPSocketI, 608
- GetShutdown
  - OscISocketIBase, 542
- getSize
  - MM\_Audit\_Imp, 154
- GetSocketError
  - OscIDNSRequestAO, 364
  - OscISocketRequestAO, 551
- getTagActualSize
  - MM\_Audit\_Imp, 154
- GetTimestamp
  - MediaData, 144
- GetTOS
  - OscISocketTOS, 564
- good
  - OscIBinStream, 338
- GOOD\_STATE
  - OscIBinStream, 337
- Handle
  - OscI\_File, 182
  - OscIFileHandle, 405
- HandleDNSEvent
  - OscIDNSObserver, 361
- HandleSocketEvent
  - OscISocketObserver, 548
- HasAsyncBind
  - OscISocketIBase, 542
- HasAsyncListen
  - OscISocketIBase, 542
- HasAsyncRead
  - OscINativeFile, 463

- hash
  - OSCL\_String, [262](#)
  - OSCL\_wString, [305](#)
- HasThisOffset
  - OscAsyncFileBuffer, [320](#)
- HaveRoomInCurrentBlock
  - OscBinStream, [338](#)
- Head
  - OscDoubleList, [367](#)
  - OscPriorityList, [469](#)
- head
  - Osc\_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
  - OscDNSRequest, [362](#)
- iAddedNum
  - PVActiveBase, [615](#)
- iAddr
  - BindParam, [116](#)
  - ConnectParam, [131](#)
  - GetHostByNameParam, [136](#)
  - RecvFromParam, [636](#)
  - SendToParam, [640](#)
- iAddress
  - OscIPSocketI, [420](#)
- iAddressList
  - GetHostByNameParam, [136](#)
- iAlloc
  - OscDNSIBase, [357](#)
  - OscDNSMethod, [360](#)
  - OscExecSchedulerCommonBase, [399](#)
  - OscIPSocketI, [420](#)
  - OscSocketIBase, [544](#)
  - OscSocketServIBase, [559](#)
- iAllocatedSz
  - OscMemPoolResizableAllocator::Mem-  
PoolBufferInfo, [455](#)
- iAOPriority
  - TReadyQueueLink, [657](#)
- iAsyncReadBufferSize
  - OscNativeFileParams, [465](#)
- iBlankSocket
  - AcceptParam, [112](#)
- iBlockBuffer
  - OscMemPoolResizableAllocator::Mem-  
PoolBlockInfo, [454](#)
- iBlockInfoAlignedSize
  - OscMemPoolResizableAllocator, [453](#)
- iBlockingMode
  - OscExecSchedulerCommonBase, [399](#)
- iBlockPostFence
  - OscMemPoolResizableAllocator::Mem-  
PoolBlockInfo, [454](#)
- iBlockPreFence
  - OscMemPoolResizableAllocator::Mem-  
PoolBlockInfo, [454](#)
- iBlockSize
  - OscMemPoolResizableAllocator::Mem-  
PoolBlockInfo, [454](#)
- iBuffer
  - OscBuf, [341](#)
- iBufferInfoAlignedSize
  - OscMemPoolResizableAllocator, [453](#)
- iBufferPostFence
  - OscMemPoolResizableAllocator::Mem-  
PoolBufferInfo, [455](#)
- iBufferPreFence
  - OscMemPoolResizableAllocator::Mem-  
PoolBufferInfo, [455](#)
- iBufferSize
  - OscMemPoolResizableAllocator::Mem-  
PoolBufferInfo, [455](#)
- iBufRecv
  - RecvFromParam, [636](#)
  - RecvParam, [638](#)
- iBufSend
  - SendParam, [639](#)
  - SendToParam, [640](#)
- iBusy
  - PVActiveBase, [615](#)
- iCancel
  - OscSocketServRequestQElem, [562](#)
- iCBase
  - OscTrapStackItem, [600](#)
- iCheckFreeMemoryAvailable
  - OscMemPoolResizableAllocator, [453](#)
- iCheckNextAvailable
  - OscMemPoolResizableAllocator, [453](#)
- iCheckNextAvailableFreeChunk
  - OscMemPoolFixedChunkAllocator, [446](#)
- iChunkAlignment
  - OscMemPoolFixedChunkAllocator, [446](#)
- iChunkSize
  - OscMemPoolFixedChunkAllocator, [446](#)
- iChunkSizeMemAligned
  - OscMemPoolFixedChunkAllocator, [446](#)
- iComponentId
  - OscComponentRegistryElement, [346](#)

- iComponentIdCounter
  - OscComponentRegistry, [344](#)
- iContainer
  - OscFileCacheBuffer, [404](#)
  - OscSocketMethod, [547](#)
  - OscSocketRequestAO, [553](#)
- Id
  - OscAsyncFileBuffer, [320](#)
  - OscSocketRequestAO, [552](#)
  - PVThreadContext, [634](#)
- iData
  - OscComponentRegistry, [344](#)
- iDebugger
  - OscExecSchedulerCommonBase, [399](#)
- iDefAlloc
  - OscExecSchedulerCommonBase, [399](#)
- iDelta
  - OscExecSchedulerCommonBase, [399](#)
- iDNSFxn
  - OscDNSMethod, [360](#)
- iDNSI
  - OscDNSRequestAO, [365](#)
- iDNSMethod
  - OscDNSRequestAO, [365](#)
- iDNSObserver
  - OscDNSMethod, [360](#)
- iDNSRequest
  - DNSRequestParam, [134](#)
- iDNSRequestAO
  - OscDNSMethod, [360](#)
  - OscDNSRequest, [362](#)
- iDNSRequestParam
  - OscDNSRequest, [362](#)
- iDoStop
  - OscExecSchedulerCommonBase, [399](#)
- iDoSuspend
  - OscExecSchedulerCommonBase, [399](#)
- iEnableNullPtrReturn
  - OscMemPoolFixedChunkAllocator, [446](#)
  - OscMemPoolResizableAllocator, [453](#)
- iEndAddr
  - OscMemPoolResizableAllocator::Mem-  
PoolBufferInfo, [455](#)
- iErrAlloc
  - OscSelect, [520](#)
- iErrorTrapImp
  - OscExecSchedulerCommonBase, [399](#)
- iExecTimerQ
  - OscExecSchedulerCommonBase, [399](#)
- iExpectedNumBlocksPerBuffer
  - OscMemPoolResizableAllocator, [453](#)
- iFactory
  - OscComponentRegistryElement, [346](#)
  - OscRegistryAccessElement, [508](#)
- iFilePosition
  - Osc\_File::OscFixedCacheParam, [188](#)
- iFlags
  - RecvFromParam, [636](#)
  - RecvParam, [638](#)
  - SendParam, [639](#)
  - SendToParam, [640](#)
- iFreeMemChunkList
  - OscMemPoolFixedChunkAllocator, [446](#)
- iFreeMemContextData
  - OscMemPoolResizableAllocator, [453](#)
- iFreeMemPoolObserver
  - OscMemPoolResizableAllocator, [453](#)
- ifront
  - Osc\_Queue\_Base, [241](#)
- iFxn
  - DNSRequestParam, [134](#)
  - SocketRequestParam, [643](#)
- iGrandTotalTicks
  - OscExecSchedulerCommonBase, [399](#)
- iHead
  - OscDoubleListBase, [369](#)
  - OscDoubleRunner, [370](#)
- iHeapCheck
  - OscSelect, [520](#)
- iHigh
  - OscInteger64Transport, [416](#)
- iHow
  - ShutdownParam, [641](#)
- iId
  - OscComponentRegistryElement, [346](#)
  - OscDNSMethod, [360](#)
  - OscIPSocketI, [420](#)
- iIsIn
  - TReadyQueLink, [657](#)
- iJumpData
  - OscErrorTrapImp, [378](#)
- iLeave
  - OscErrorTrapImp, [378](#)
- iLen
  - PVSockBufRecv, [632](#)
  - PVSockBufSend, [633](#)
- iLength
  - OscBuf, [341](#)
- iLogger
  - OscDNSMethod, [360](#)
  - OscDNSRequestAO, [365](#)
  - OscExecSchedulerCommonBase, [399](#)
  - OscIPSocketI, [420](#)
  - OscSocketServIBase, [559](#)
- iLogPerfIndentStr
  - OscExecSchedulerCommonBase, [399](#)
- iLogPerfIndentStrLen
  - OscExecSchedulerCommonBase, [399](#)

- iLogPerfTotal
  - OscIExecSchedulerCommonBase, 399
- iLow
  - OscInteger64Transport, 416
- iMaxLen
  - PVSockBufRecv, 632
- iMaxLength
  - OscBuf, 341
- iMaxNewMemPoolBufferSz
  - OscMemPoolResizableAllocator, 453
- iMemPool
  - OscMemPoolFixedChunkAllocator, 446
- iMemPoolAligned
  - OscMemPoolFixedChunkAllocator, 446
- iMemPoolAllocator
  - OscMemPoolFixedChunkAllocator, 446
- iMemPoolBufferAllocator
  - OscMemPoolResizableAllocator, 453
- iMemPoolBufferList
  - OscMemPoolResizableAllocator, 453
- iMemPoolBufferNumLimit
  - OscMemPoolResizableAllocator, 453
- iMemPoolBufferSize
  - OscMemPoolResizableAllocator, 453
- iMimeType
  - OscRegistryAccessElement, 508
- iMultiMaxLen
  - RecvFromParam, 636
- iMutex
  - OscComponentRegistry, 344
- iName
  - GetHostByNameParam, 136
  - OscIExecSchedulerCommonBase, 399
  - PVActiveBase, 615
- iNativeAccessMode
  - OscNativeFileParams, 465
- iNativeBufferSize
  - OscNativeFileParams, 465
- iNativeMode
  - OscIExecSchedulerCommonBase, 399
- IncLogPerf
  - OscIExecSchedulerCommonBase, 396
- increment\_refcnt
  - BufferState, 119
- iNext
  - OscDoubleLink, 366
  - OscDoubleRunner, 370
  - OscTrapStackItem, 600
- iNextAvailableContextData
  - OscMemPoolFixedChunkAllocator, 446
  - OscMemPoolResizableAllocator, 453
- iNextFreeBlock
  - OscMemPoolResizableAllocator::Mem-PoolBlockInfo, 454
- OscMemPoolResizableAllocator::Mem-PoolBufferInfo, 455
- Init
  - OscErrorTrap, 376
  - OscInit, 415
  - OscMem, 425
  - OscScheduler, 516
  - PVLogger, 620
- InitExecQ
  - OscIExecSchedulerCommonBase, 396
- Insert
  - OscDoubleListBase, 369
  - OscPriorityList, 469
- insert
  - Osc\_Map, 220
  - Osc\_TagTree, 272
  - Osc\_Vector, 288
  - Osc\_Vector\_Base, 292
- insert\_element
  - Osc\_Linked\_List, 208
  - Osc\_Linked\_List\_Base, 213
- insert\_unique
  - Osc\_Rb\_Tree, 244
- InsertAfter
  - OscDoubleLink, 366
- InsertBefore
  - OscDoubleLink, 366
- InsertHead
  - OscDoubleList, 367
  - OscDoubleListBase, 369
- InsertTail
  - OscDoubleList, 367
  - OscDoubleListBase, 369
- InstallScheduler
  - OscIExecSchedulerCommonBase, 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
  - OscIpMReq, 417
- INTERNAL\_ERROR
  - BufFragStatusClass, 123
- internalLeave, 139
  - osclerror, 88
- internalLeave
  - a, 139
- InThread
  - DNSRequestParam, 133
- iNumAOAdded



- OscExecSchedulerCommonBase, [399](#)
- iNumChunk
  - OscMemPoolFixedChunkAllocator, [446](#)
- iNumOfRun
  - OscAsyncFile, [318](#)
- iNumOfRunErr
  - OscAsyncFile, [318](#)
- iNumOutstanding
  - OscMemPoolResizableAllocator::MemPoolBufferInfo, [455](#)
- iNumSessions
  - OscComponentRegistry, [344](#)
- INVALID\_ACCESS\_ERROR
  - OscProcStatus, [476](#)
- INVALID\_ARGUMENT\_ERROR
  - OscProcStatus, [476](#)
- INVALID\_FUNCTION\_ERROR
  - OscProcStatus, [476](#)
- INVALID\_HANDLE\_ERROR
  - OscProcStatus, [476](#)
- INVALID\_ID
  - BufFragStatusClass, [123](#)
- INVALID\_OPERATION\_ERROR
  - OscProcStatus, [476](#)
- INVALID\_PARAM\_ERROR
  - OscProcStatus, [475](#)
- INVALID\_POINTER\_ERROR
  - OscProcStatus, [476](#)
- INVALID\_PRIORITY\_ERROR
  - OscProcStatus, [475](#)
- INVALID\_THREAD\_ERROR
  - OscProcStatus, [475](#)
- INVALID\_THREAD\_ID\_ERROR
  - OscProcStatus, [475](#)
- INVALID\_TYPE
  - Osc\_FileFind, [189](#)
- iObserver
  - OscIPSocketI, [420](#)
  - OscMemPoolFixedChunkAllocator, [446](#)
  - OscMemPoolResizableAllocator, [453](#)
- iOffset
  - OscDoubleListBase, [369](#)
  - OscDoubleRunner, [370](#)
- iOpCount
  - OscFileStatsItem, [412](#)
- iOscBase
  - OscSelect, [520](#)
- iOscErrorTrap
  - OscSelect, [520](#)
- iOscLogger
  - OscSelect, [520](#)
- iOscMemory
  - OscSelect, [520](#)
- iOscScheduler
  - OscSelect, [520](#)
- iOtherExecStats
  - OscExecSchedulerCommonBase, [399](#)
- iOutputFile
  - OscSelect, [520](#)
- iPacketLen
  - RecvFromParam, [636](#)
- iPacketSource
  - RecvFromParam, [636](#)
- ipAddr
  - OscNetworkAddress, [466](#)
- iParam
  - OscFileStatsItem, [412](#)
  - OscSocketRequest, [549](#)
  - OscSocketRequestAO, [553](#)
- iParam2
  - OscFileStatsItem, [412](#)
- iParamSize
  - OscSocketRequestAO, [553](#)
- iParentBuffer
  - OscMemPoolResizableAllocator::MemPoolBlockInfo, [454](#)
- iPrev
  - OscDoubleLink, [366](#)
- iPrevFreeBlock
  - OscMemPoolResizableAllocator::MemPoolBlockInfo, [454](#)
- iPriority
  - OscPriorityLink, [468](#)
- iPtr
  - PVSockBufRecv, [632](#)
  - PVSockBufSend, [633](#)
- iPVAActiveStats
  - PVActiveBase, [615](#)
- iPVReadyQLink
  - PVActiveBase, [615](#)
- iPVStatQ
  - OscExecSchedulerCommonBase, [399](#)
- iPVStats
  - OscExecSchedulerCommonBase, [399](#)
- iQSize
  - ListenParam, [141](#)
- iReadyQ
  - OscExecSchedulerCommonBase, [399](#)
- irear
  - Osc\_Queue\_Base, [241](#)
- iRefCount
  - DNSRequestParam, [134](#)
  - OscMemPoolFixedChunkAllocator, [446](#)
  - OscMemPoolResizableAllocator, [453](#)
- iRequestedAvailableFreeMemSize
  - OscMemPoolResizableAllocator, [453](#)
- iRequestedNextAvailableSize
  - OscMemPoolResizableAllocator, [453](#)



- iResumeSem
  - OscIExecSchedulerCommonBase, [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
  - OscIActiveObject, [312](#)
  - OscITimerObject, [588](#)
- iSchedulerAlloc
  - OscISelect, [520](#)
- iSchedulerName
  - OscISelect, [520](#)
- iSchedulerReserve
  - OscISelect, [520](#)
- isCIEquivalentTo
  - StrCSumPtrLen, [645](#)
  - StrPtrLen, [648](#)
  - WStrPtrLen, [659](#)
- isCIPrefixOf
  - StrPtrLen, [648](#)
- iSelect
  - OscISocketServRequestQElem, [562](#)
- IsEmpty
  - OscIDoubleListBase, [369](#)
- iSeqNum
  - TReadyQueLink, [657](#)
- iServerError
  - OscISocketServIBase, [559](#)
- iServState
  - OscISocketServIBase, [559](#)
- isFixed
  - OscIFileCacheBuffer, [404](#)
- IsHead
  - OscIDoubleList, [367](#)
  - OscIPriorityList, [469](#)
- IsIn
  - OscIReadyQ, [485](#)
  - OscITimerQ, [591](#)
- IsInAnyQ
  - PVActiveBase, [614](#)
- IsInstalled
  - OscIExecSchedulerCommonBase, [396](#)
- IsInUse
  - OscIAsyncFileBuffer, [320](#)
- iSize
  - OscI\_File::OscIFixedCacheParam, [188](#)
- isLetter
  - StrPtrLen, [648](#)
- IsLocalData
  - MediaData, [144](#)
- ISO8601TIME\_BUFFER\_SIZE
  - osclbase, [46](#)
- ISO8601timeStrBuf
  - osclbase, [35](#)
- ISO8601ToRFC822
  - osclbase, [36](#)
- iSocket
  - OscIIPSocketI, [420](#)
- iSocketError
  - OscIDNSRequestAO, [365](#)
  - OscISocketRequestAO, [553](#)
- iSocketFxn
  - OscISocketMethod, [547](#)
- iSocketI
  - OscISocketRequest, [549](#)
- iSocketRequest
  - OscISocketServRequestQElem, [562](#)
- iSocketRequestAO
  - OscISocketMethod, [547](#)
  - OscISocketRequest, [549](#)
- iSocketServ
  - OscIDNSIBase, [357](#)
  - OscIIPSocketI, [420](#)
  - OscISocketIBase, [544](#)
- IsOpen
  - OscISocketIBase, [542](#)
- IsReady
  - OscIDNSIBase, [356](#)
- IsSameThreadContext
  - PVThreadContext, [634](#)
- IsServConnected
  - OscISocketServIBase, [559](#)
- IsServerThread
  - OscISocketServI, [557](#)
- isSetFailure
  - MM\_Audit\_Imp, [155](#)
- IsStarted
  - OscIExecSchedulerCommonBase, [396](#)
- IsTail
  - OscIDoubleList, [367](#)
  - OscIPriorityList, [469](#)
- iStartAddr
  - OscIMemPoolResizableAllocator::Mem-  
PoolBufferInfo, [455](#)
- iStartTick
  - OscIFileStatsItem, [412](#)
- iStatus
  - PVActiveBase, [615](#)

- iStopper
  - OscExecSchedulerCommonBase, [399](#)
- iStopperCrit
  - OscExecSchedulerCommonBase, [399](#)
- IsUpdated
  - OscFileCacheBuffer, [404](#)
- iSuspended
  - OscExecSchedulerCommonBase, [399](#)
- IsValid
  - OscAsyncFileBuffer, [320](#)
- iTAny
  - OscTrapStackItem, [600](#)
- iterator
  - Osc\_Linked\_List\_Base, [215](#)
  - Osc\_Map, [218](#)
  - Osc\_Rb\_Tree, [244](#)
  - Osc\_Rb\_Tree\_Iterator, [251](#)
  - Osc\_TagTree::iterator, [277](#)
  - Osc\_Vector, [286](#)
  - OscPriorityQueue, [471](#)
- iThreadContext
  - OscExecSchedulerCommonBase, [399](#)
  - PVActiveBase, [615](#)
- iTime
  - OscExecSchedulerCommonBase, [399](#)
- iTimeCompareThreshold
  - OscExecSchedulerCommonBase, [399](#)
- iTimeQueuedTicks
  - TReadyQueueLink, [657](#)
- iTimeToRunTicks
  - TReadyQueueLink, [657](#)
- iTotalPercent
  - OscExecSchedulerCommonBase, [399](#)
- iTotalTicks
  - OscFileStatsItem, [412](#)
- iTotalTicksTemp
  - OscExecSchedulerCommonBase, [399](#)
- iTrapOperation
  - OscTrapStackItem, [600](#)
- iTrapStack
  - OscErrorTrapImp, [378](#)
- iVec
  - OscComponentRegistryData, [345](#)
- iXferLen
  - SendParam, [639](#)
  - SendToParam, [640](#)
- Join
  - OscIPSocketI, [419](#)
  - OscSocketI, [537](#)
  - OscSocketIBase, [542](#)
  - OscUDPSocket, [603](#)
- JoinMulticastGroup
  - OscUDPSocket, [604](#)
- OscUDPSocketI, [608](#)
- Jump
  - OscJump, [421](#)
- key\_comp
  - Osc\_Map, [221](#)
- key\_compare
  - Osc\_Map, [218](#)
- key\_type
  - Osc\_Map, [218](#)
  - Osc\_Rb\_Tree, [244](#)
- largeasyncfilereadwrite\_test
  - Osc\_File, [186](#)
- Leave
  - OscError, [372](#)
- LeaveIfError
  - OscError, [372](#)
- LeaveIfNull
  - OscError, [372](#)
- Left
  - OscPtrC, [480](#)
- left
  - Osc\_Rb\_Tree\_Node\_Base, [255](#)
- len
  - OscMemoryFragment, [442](#)
  - StrPtrLen, [648](#)
  - WStrPtrLen, [659](#)
- Length
  - OscAsyncFileBuffer, [320](#)
  - OscBuf, [341](#)
  - OscPtr, [477](#)
  - OscPtrC, [480](#)
- length
  - BufFragGroup, [122](#)
  - OscBinStream, [339](#)
  - StrPtrLen, [648](#)
  - WStrPtrLen, [659](#)
- lineNo
  - MM\_AllocInfo, [150](#)
  - MM\_AllocQueryInfo, [152](#)
- link\_type
  - Osc\_Rb\_Tree, [244](#)
  - Osc\_Rb\_Tree\_Const\_Iterator, [248](#)
  - Osc\_Rb\_Tree\_Iterator, [251](#)
  - Osc\_Rb\_Tree\_Node, [253](#)
- LinkedListElement, [140](#)
  - LinkedListElement, [140](#)
- LinkedListElement
  - data, [140](#)
  - LinkedListElement, [140](#)
  - next, [140](#)
- Listen
  - OscListenMethod, [422](#)

- OslListenRequest, [423](#)
- OslSocketI, [537](#)
- OslSocketIBase, [542](#)
- OslTCPSocket, [569](#)
- OslTCPSocketI, [573](#)
- ListenAsync
  - OslSocketIBase, [542](#)
  - OslTCPSocket, [569](#)
  - OslTCPSocketI, [574](#)
- ListenParam, [141](#)
  - ListenParam, [141](#)
- ListenParam
  - iQSize, [141](#)
  - ListenParam, [141](#)
- ListenRequest
  - OslListenMethod, [422](#)
- little\_endian\_to\_host
  - osclbase, [37](#)
- localbuf
  - MediaData, [144](#)
- Lock
  - OslLockBase, [424](#)
  - OslMutex, [460](#)
  - OslNullLock, [467](#)
  - OslThreadLock, [579](#)
- lockAndGetInstance
  - OslSingletonRegistry, [534](#)
- Log
  - OslFileStats, [411](#)
- log\_level\_type
  - AllPassFilter, [114](#)
  - PVLogger, [618](#)
  - PVLoggerFilter, [624](#)
  - PVLoggerRegistry, [628](#)
- LogAll
  - OslFileStats, [411](#)
- Logger
  - OslSocketI, [537](#)
- LogMsgBuffers
  - PVLogger, [620](#)
- LogMsgBuffersV
  - PVLogger, [620](#)
- LogMsgString
  - PVLogger, [621](#)
- LogMsgStringV
  - PVLogger, [621](#)
- LoopbackSocket
  - OslSocketServI, [557](#)
- lower\_bound
  - Osl\_Map, [221](#)
  - Osl\_Rb\_Tree, [244](#)
- MakeAddr
  - OslSocketI, [537](#)
- MakeMulticastGroupInformation
  - OslSocketI, [537](#)
- makeValidTag
  - MM\_Audit\_Imp, [155](#)
- map\_type
  - Osl\_TagTree, [270](#)
- mapit
  - Osl\_TagTree::const\_iterator, [274](#)
  - Osl\_TagTree::iterator, [277](#)
- mapiter
  - Osl\_TagTree::const\_iterator, [274](#)
  - Osl\_TagTree::iterator, [277](#)
- Match
  - OslComponentRegistryElement, [346](#)
- MAX\_NUMBER\_OF\_BYTE\_PER\_UTF8
  - osclutil, [69](#)
- max\_size
  - Osl\_Map, [221](#)
  - Osl\_Rb\_Tree, [244](#)
- MAX\_THRDS\_REACHED\_ERROR
  - OslProcStatus, [475](#)
- maximum
  - Osl\_Rb\_Tree\_Node\_Base, [255](#)
- MaxLen
  - OslNameString, [461](#)
- maxsize
  - CFastRep, [128](#)
  - CHepRep, [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](#)
  - GetNumMediaFrgs, [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
  - OscMemPoolResizableAllocator, [451](#)
- message\_id\_type
  - AllPassFilter, [114](#)
  - PVLogger, [618](#)
  - PVLoggerAppender, [623](#)
  - PVLoggerFilter, [624](#)
  - PVLoggerLayout, [626](#)
- MethodDone
  - OscIDNSMethod, [359](#)
  - OscSocketMethod, [546](#)
- MICROSECONDS
  - osclbase, [35](#)
- MILLISECONDS
  - osclbase, [35](#)
- MIN\_FENCE\_SIZE
  - osclmemory, [52](#)
- minimum
  - OscRb\_Tree\_Node\_Base, [255](#)
- MM\_AddTag
  - MM\_Audit\_Imp, [155](#)
  - OscMemAudit, [430](#)
- MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN
  - osclmemory, [52](#)
- MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN
  - osclmemory, [52](#)
- MM\_allocate
  - MM\_Audit\_Imp, [155](#)
  - OscMemAudit, [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\_GetTagNode, [157](#)
- MM\_GetTreeNode, [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](#)
  - OscMemAudit, [430](#)
- MM\_deallocate
  - MM\_Audit\_Imp, [155](#)
  - OscMemAudit, [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](#)
  - OscMemAudit, [430](#)
- MM\_GetAllocNodeInfo
  - MM\_Audit\_Imp, [156](#)
  - OscMemAudit, [430](#)
- MM\_GetExistingTag
  - MM\_Audit\_Imp, [156](#)
  - OscMemAudit, [431](#)
- MM\_GetMode
  - MM\_Audit\_Imp, [156](#)
  - OscMemAudit, [431](#)
- MM\_GetNumAllocNodes
  - MM\_Audit\_Imp, [156](#)
  - OscMemAudit, [431](#)
- MM\_GetOverheadStats
  - MM\_Audit\_Imp, [156](#)
  - OscMemAudit, [431](#)
- MM\_GetPostfillPattern
  - MM\_Audit\_Imp, [156](#)
  - OscMemAudit, [431](#)
- MM\_GetPrefillPattern
  - MM\_Audit\_Imp, [156](#)
  - OscMemAudit, [431](#)
- MM\_GetRefCount
  - OscMemAudit, [431](#)
- MM\_GetRootNode
  - MM\_Audit\_Imp, [157](#)
  - OscMemAudit, [432](#)
- MM\_GetStats
  - MM\_Audit\_Imp, [157](#)
  - OscMemAudit, [432](#)
- MM\_GetStatsInDepth
  - MM\_Audit\_Imp, [157](#)
  - OscMemAudit, [432](#)
- MM\_GetTagNode
  - MM\_Audit\_Imp, [157](#)

- OslMemAudit, [432](#)
- MM\_GetTreeNodes
  - MM\_Audit\_Imp, [157](#)
  - OslMemAudit, [432](#)
- MM\_ReleaseAllocNodeInfo
  - MM\_Audit\_Imp, [157](#)
  - OslMemAudit, [432](#)
- MM\_SetFailurePoint
  - MM\_Audit\_Imp, [158](#)
  - OslMemAudit, [432](#)
- MM\_SetMode
  - MM\_Audit\_Imp, [158](#)
  - OslMemAudit, [433](#)
- MM\_SetPostfillPattern
  - MM\_Audit\_Imp, [158](#)
  - OslMemAudit, [433](#)
- MM\_SetPrefillPattern
  - MM\_Audit\_Imp, [158](#)
  - OslMemAudit, [433](#)
- MM\_SetTagLevel
  - MM\_Audit\_Imp, [158](#)
  - OslMemAudit, [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
  - oslmemory, [59](#)
- MM\_UnsetFailurePoint
  - MM\_Audit\_Imp, [158](#)
  - OslMemAudit, [433](#)
- MM\_Validate
  - MM\_Audit\_Imp, [158](#)
  - OslMemAudit, [433](#)
- MMAuditCharAutoPtr
  - oslmemory, [59](#)
- MMAuditUInt8AutoPtr
  - oslmemory, [59](#)
- Mode
  - OslNativeFile, [463](#)
- mode
  - oscl\_stat\_buf, [259](#)
- MODE\_APPEND
  - Osl\_File, [180](#)
- MODE\_BINARY
  - Osl\_File, [180](#)
- MODE\_READ
  - Osl\_File, [180](#)
- MODE\_READ\_PLUS
  - Osl\_File, [180](#)
- MODE\_READWRITE
  - Osl\_File, [180](#)
- MODE\_TEXT
  - Osl\_File, [180](#)
- mode\_type
  - Osl\_File, [180](#)
- move\_to\_end
  - Osl\_Linked\_List, [209](#)
  - Osl\_Linked\_List\_Base, [213](#)
  - Osl\_MTLlinked\_List, [226](#)
- move\_to\_front
  - Osl\_Linked\_List, [209](#)
  - Osl\_Linked\_List\_Base, [214](#)
  - Osl\_MTLlinked\_List, [227](#)
- MSEC\_PER\_SEC
  - osclbase, [46](#)
- MSEC\_TO\_MICROSEC
  - oscl\_socket\_method.h, [759](#)
- MsecToTicks
  - OslTickCount, [580](#)
- multicastAddr
  - OslIpMReq, [417](#)
- MUTEX\_LOCKED\_ERROR
  - OslProcStatus, [476](#)
- nAllocNum
  - MM\_FailInsertParam, [162](#)
- New
  - Osl\_DefAllocWithRefCount, [174](#)
- NewL
  - OslAcceptMethod, [308](#)
  - OslAsyncFile, [317](#)
  - OslAsyncFileBuffer, [320](#)
  - OslBindMethod, [322](#)
  - OslBuf, [341](#)
  - OslConnectMethod, [348](#)
  - OslDNS, [352](#)
  - OslDNSI, [354](#)
  - OslGetHostByNameMethod, [413](#)
  - OslListenMethod, [422](#)
  - OslRecvFromMethod, [486](#)
  - OslRecvMethod, [490](#)

- OslSendMethod, [523](#)
- OslSendToMethod, [525](#)
- OslShutdownMethod, [530](#)
- OslSocketI, [537](#)
- OslSocketServ, [555](#)
- OslSocketServI, [557](#)
- OslTCPSocket, [569](#)
- OslTCPSocketI, [574](#)
- OslUDPSocket, [604](#)
- OslUDPSocketI, [608](#)
- NewRequest
  - OslDNSRequestAO, [364](#)
  - OslSocketRequestAO, [552](#)
- next
  - BufFragGroup, [122](#)
  - LinkedListElement, [140](#)
- nextFragPtr
  - OslBinStream, [339](#)
- NO\_PERMISSION\_ERROR
  - OslProcStatus, [475](#)
- Node
  - Osl\_TagTree::Node, [280](#)
- node
  - Osl\_Rb\_Tree\_Const\_Iterator, [248](#)
  - Osl\_Rb\_Tree\_Iterator, [251](#)
- node\_ptr
  - Osl\_TagTree, [270](#)
- node\_type
  - Osl\_TagTree, [270](#)
- NOT\_ENOUGH\_MEMORY\_ERROR
  - OslProcStatus, [475](#)
- NOT\_ENOUGH\_RESOURCES\_ERROR
  - OslProcStatus, [475](#)
- NOT\_ENOUGH\_SPACE
  - BufFragStatusClass, [123](#)
- NOT\_IMPLEMENTED
  - OslProcStatus, [476](#)
- NOT\_SUSPENDED\_ERROR
  - OslProcStatus, [475](#)
- notifyfreeblockavailable
  - OslMemPoolResizableAllocator, [451](#)
- notifyfreechunkavailable
  - OslMemPoolFixedChunkAllocator, [445](#)
- notifyfreememoryavailable
  - OslMemPoolResizableAllocator, [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](#)
  - set\_from\_system\_time, [169](#)
  - set\_to\_current\_time, [169](#)
  - TimeValue, [655](#)
  - to\_system\_time, [169](#)
- NULL
  - osclbase, [32](#)
- NULL\_INPUT
  - BufFragStatusClass, [123](#)
- NULL\_TERM\_CHAR
  - osclbase, [32](#)
- num\_child\_nodes
  - MM\_Stats\_CB, [163](#)
- num\_elements
  - Osl\_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
  - Osl\_Queue\_Base, [241](#)
  - Osl\_Vector\_Base, [293](#)
- numFrag
  - OslBinStream, [339](#)
- octet
  - osclbase, [35](#)
- Offset
  - OslAsyncFileBuffer, [320](#)
- Open
  - Osl\_File, [182](#)
  - OslAsyncFile, [317](#), [318](#)
  - OslDNSI, [354](#)
  - OslDNSIBase, [356](#)
  - OslFileCache, [402](#)
  - OslNativeFile, [463](#)
  - OslSocketI, [537](#)
  - OslSocketIBase, [543](#)
  - OslSocketServRequestList, [560](#)
- OpenSession
  - OslComponentRegistry, [344](#)
- operator \*
  - Osl\_Rb\_Tree\_Const\_Iterator, [248](#)
  - Osl\_Rb\_Tree\_Iterator, [251](#)
  - Osl\_TagTree::const\_iterator, [274](#)
  - Osl\_TagTree::iterator, [277](#)
  - OslExclusiveArrayPtr, [381](#)
  - OslExclusivePtr, [384](#)
  - OslExclusivePtrA, [387](#)



- OSCLMemAutoPtr, [437](#)
- OscISharedPtr, [528](#)
- OscISingleton, [532](#)
- OscITLS, [592](#)
- OscITLSEx, [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](#)
  - OscIErrorAllocator, [375](#)
  - osclmemory, [60](#)
  - OscIMemStatsNode, [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](#)
  - OscIErrorAllocator, [375](#)
  - osclmemory, [60](#)
  - OscIMemStatsNode, [458](#)
- operator new[]
  - osclmemory, [60](#)
- operator T \*
  - OscIDoubleRunner, [370](#)
- operator TheClass \*
  - OscISharedPtr, [529](#)
- operator !=
  - OscI\_Rb\_Tree\_Const\_Iterator, [248](#)
  - OscI\_Rb\_Tree\_Iterator, [251](#)
  - OSCL\_String, [262](#)
  - OscI\_TagTree::const\_iterator, [274](#)
  - OscI\_TagTree::iterator, [277](#)
  - OSCL\_wString, [306](#)
  - OscIAOStatus, [315](#)
  - OscIUuid, [611](#)
  - StrCSumPtrLen, [645](#)
  - StrPtrLen, [648](#)
  - TimeValue, [655](#)
  - WStrPtrLen, [659](#)
- operator()
  - OscI\_Less, [205](#)
  - OscI\_Map::value\_compare, [223](#)
  - OscI\_Select1st, [256](#)
  - OscI\_Tag\_Base, [268](#)
- operator+
  - osclbase, [37](#)
- operator++
  - OscI\_Rb\_Tree\_Const\_Iterator, [248](#)
  - OscI\_Rb\_Tree\_Iterator, [251](#)
  - OscI\_TagTree::const\_iterator, [274](#)
  - OscI\_TagTree::iterator, [277](#)
  - OscIDoubleRunner, [370](#)
- operator+=
  - NTPTIME, [168](#)
  - OSCL\_String, [262](#)
  - OSCL\_wString, [306](#)
  - TimeValue, [654](#)
- operator-
  - NTPTIME, [168](#)
  - osclbase, [37](#)
- operator-
  - OscI\_Rb\_Tree\_Const\_Iterator, [248](#)
  - OscI\_Rb\_Tree\_Iterator, [251](#)
  - OscI\_TagTree::const\_iterator, [274](#)
  - OscI\_TagTree::iterator, [277](#)
  - OscIDoubleRunner, [370](#)
- operator-=
  - TimeValue, [654](#)
- operator->
  - OscI\_Rb\_Tree\_Const\_Iterator, [248](#)
  - OscI\_Rb\_Tree\_Iterator, [251](#)
  - OscI\_TagTree::const\_iterator, [274](#)
  - OscI\_TagTree::iterator, [277](#)
  - OscIExclusiveArrayPtr, [381](#)
  - OscIExclusivePtr, [384](#)
  - OscIExclusivePtrA, [387](#)
  - OSCLMemAutoPtr, [437](#)
  - OscISharedPtr, [529](#)
  - OscISingleton, [532](#)
  - OscITLS, [592](#)
  - OscITLSEx, [594](#)
- operator<
  - OSCL\_String, [262](#)
  - OscI\_Tag, [265](#)
  - OSCL\_wString, [306](#)
  - OscIAOStatus, [315](#)
  - TimeValue, [655](#)
- operator<<
  - OscIBinOStreamBigEndian, [333](#)
  - OscIBinOStreamLittleEndian, [335](#)
- operator<=
  - OSCL\_String, [263](#)
  - OSCL\_wString, [306](#)
  - OscIAOStatus, [315](#)
  - TimeValue, [655](#)
- operator=
  - NTPTIME, [168](#), [169](#)
  - OSCL\_FastString, [177](#)



- OSCL\_HeapStringA, 201
- OscI\_Map, 221
- OscI\_Rb\_Tree, 244
- OSCL\_String, 263
- OscI\_TagTree, 272
- OscI\_Vector, 288
- OSCL\_wFastString, 295
- OSCL\_wHeapStringA, 301
- OSCL\_wString, 306
- OscIAOStatus, 315
- OscIComponentRegistryElement, 346
- OscIExclusiveArrayPtr, 381
- OscIExclusivePtr, 384
- OscIExclusivePtrA, 387
- OSCLMemAutoPtr, 437
- OscIRefCounterMemFrag, 497
- OscISharedPtr, 529
- osclutil, 72–74
- OscIUuid, 611
- StrCSumPtrLen, 645
- StrPtrLen, 648
- TimeValue, 654
- WStrPtrLen, 659
- operator==
  - OscI\_Rb\_Tree\_Const\_Iterator, 248
  - OscI\_Rb\_Tree\_Iterator, 251
  - OSCL\_String, 263
  - OscI\_TagTree::const\_iterator, 274
  - OscI\_TagTree::iterator, 277
  - OSCL\_wString, 306
  - OscIAOStatus, 315
  - osclbase, 37
  - OscINetworkAddress, 466
  - OscIUuid, 611
  - StrCSumPtrLen, 645
  - StrPtrLen, 648
  - TimeValue, 655
  - WStrPtrLen, 659
- operator>
  - OSCL\_String, 263
  - OSCL\_wString, 306
  - OscIAOStatus, 315
  - TimeValue, 655
- operator>=
  - OSCL\_String, 263
  - OSCL\_wString, 306
  - OscIAOStatus, 315
  - TimeValue, 655
- operator>>
  - OscIBinIStreamBigEndian, 327
  - OscIBinIStreamLittleEndian, 330
- operator[]
  - OscI\_Map, 221
  - OSCL\_String, 263
  - OscI\_TagTree, 272
  - OscI\_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
- OscI\_Alloc, 170
  - ~OscI\_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
  - osclerror, [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
  - osclerror, [88](#)
- OSCL\_CATCH\_ANY
  - osclerror, [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\_Cistrncmp
  - 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](#)
- Osc\_Dealloc, [171](#)
  - ~Osc\_Dealloc, [171](#)
  - deallocate, [171](#)
- Osc\_DefAlloc, [172](#)
- Osc\_DefAlloc
  - allocate, [172](#)
  - allocate\_fl, [172](#)
  - deallocate, [172](#)
- oscl\_defalloc.h, [667](#)
- Osc\_DefAllocWithRefCounter, [173](#)
- Osc\_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](#)
- Osc\_DeleteFile
  - Osc\_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
  - optype, 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
  - asynclilereadcancel\_test, 186
  - asynclilereadwrite\_test, 186
  - Close, 181
  - EndOfFile, 181
  - Flush, 182
  - GetError, 182
  - Handle, 182
  - largeasynclilereadwrite\_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
  - iFilePosition, 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

- OscFileManager, [406](#)
- OSCL\_FILE\_ATTRIBUTE\_NORMAL
  - OscFileManager, [406](#)
- OSCL\_FILE\_ATTRIBUTE\_READONLY
  - OscFileManager, [406](#)
- OSCL\_FILE\_ATTRIBUTE\_SYSTEM
  - OscFileManager, [406](#)
- OSCL\_FILE\_ATTRIBUTE\_TYPE
  - OscFileManager, [406](#)
- OSCL\_FILE\_BUFFER\_MAX\_SIZE
  - osclconfig\_io.h, [815](#)
- oscl\_file\_cache.h, [692](#)
- OSCL\_FILE\_CHAR\_PATH\_DELIMITER
  - osclio, [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
  - osclio, [97](#)
- oscl\_file\_types.h, [702](#)
- OSCL\_FILE\_WCHAR\_PATH\_DELIMITER
  - osclio, [97](#)
- Osc\_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](#)
- Osc\_FileFind, [189](#)
  - Osc\_FileFind, [190](#)
- Osc\_FileFind
  - ~Osc\_FileFind, [190](#)
  - Close, [190](#)
  - element\_type, [189](#)
  - error\_type, [189](#)
  - FindFirst, [190](#)
  - FindNext, [191](#)
  - GetElementType, [191](#)
  - GetLastError, [191](#)
  - Osc\_FileFind, [190](#)
- OSCL\_FILEMGMT\_E\_ALREADY\_EXISTS
  - osclio, [98](#)
- OSCL\_FILEMGMT\_E\_NO\_MATCH
  - osclio, [98](#)
- OSCL\_FILEMGMT\_E\_NOT\_EMPTY
  - osclio, [98](#)
- OSCL\_FILEMGMT\_E\_NOT\_-IMPLEMENTED
  - osclio, [98](#)
- OSCL\_FILEMGMT\_E\_OK
  - osclio, [98](#)
- OSCL\_FILEMGMT\_E\_PATH\_NOT\_FOUND
  - osclio, [98](#)
- OSCL\_FILEMGMT\_E\_PATH\_TOO\_LONG
  - osclio, [98](#)
- OSCL\_FILEMGMT\_E\_PERMISSION\_-DENIED
  - osclio, [98](#)
- OSCL\_FILEMGMT\_E\_SYS\_SPECIFIC
  - osclio, [98](#)
- OSCL\_FILEMGMT\_E\_UNKNOWN
  - osclio, [98](#)
- OSCL\_FILEMGMT\_ERR\_TYPE
  - osclio, [98](#)
- OSCL\_FILEMGMT\_MODE\_DIR
  - osclio, [98](#)
- OSCL\_FILEMGMT\_MODES
  - osclio, [98](#)
- OSCL\_FILEMGMT\_PERMS
  - osclio, [98](#)
- OSCL\_FILEMGMT\_PERMS\_EXECUTE
  - osclio, [98](#)
- OSCL\_FILEMGMT\_PERMS\_READ
  - osclio, [98](#)
- OSCL\_FILEMGMT\_PERMS\_WRITE
  - osclio, [98](#)
- Osc\_FileServer, [193](#)
  - Osc\_FileServer, [193](#)
- Osc\_FileServer
  - ~Osc\_FileServer, [193](#)
  - Close, [193](#)
  - Connect, [193](#)
  - Osc\_DeleteFile, [193](#), [194](#)
  - Osc\_File, [194](#)
  - Osc\_FileServer, [193](#)
  - OscNativeFile, [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
- Osc\_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
  - \_OscInteger64Transport, 705
- OSCL\_INTEGERS\_WORD\_ALIGNED
  - osclconfig, 23
- OSCL\_IO\_EXTENSION\_MAXLEN
  - osclio, 97
- OSCL\_IO\_FILENAME\_MAXLEN
  - osclio, 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](#)
- OSCL\_IsErrnoSupported
  - osclerror, [93](#)
- oscl\_isLetter
  - osclbase, [38](#)
- OSCL\_JUMP\_MAX\_JUMP\_MARKS
  - osclerror, [89](#)
- OSCL\_LAST\_CATCH
  - osclerror, [89](#)
- OSCL\_LEAVE
  - osclerror, [89](#)
- OscL\_Less, [205](#)
  - operator(), [205](#)
- OSCL\_LIB\_READ\_DEBUG\_LIBS
  - osclconfig\_lib.h, [824](#)
- OscL\_Linked\_List, [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](#)
- oscl\_linked\_list.h, [707](#)
- OscL\_Linked\_List\_Base, [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](#)
- oscl\_lock\_base.h, [708](#)
- oscl\_log
  - osclutil, [75](#)
- oscl\_log10
  - osclutil, [75](#)
- OSCL\_MALLOC
  - osclmemory, [57](#)
- oscl\_malloc
  - osclmemory, [57](#)
- OscL\_Map, [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](#)
- oscl\_map.h, [709](#)
  - OSCL\_DISABLE\_WARNING\_-  
TRUNCATE\_DEBUG\_MESSAGE,  
[709](#)
- OscL\_Map::value\_compare, [223](#)
  - comp, [223](#)
  - operator(), [223](#)
  - OscL\_Map< Key, T, Alloc, Compare >, [223](#)
  - value\_compare, [223](#)
- OscL\_Map< Key, T, Alloc, Compare >
  - OscL\_Map::value\_compare, [223](#)
- oscl\_math.h, [710](#)
- OSCL\_MAX
  - 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
- Osc\_MTLLinked\_List, 225
  - ~Osc\_MTLLinked\_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
  - Osc\_MTLLinked\_List, 225
  - remove\_element, 227
  - the\_list, 227
- oscl\_mutex.h, 723
  - OscNoYieldMutex, 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
- Osc\_Opaque\_Type\_Alloc, 229
  - ~Osc\_Opaque\_Type\_Alloc, 229
  - allocate, 229
  - construct, 229
  - deallocate, 229
  - destroy, 229
- Osc\_Opaque\_Type\_Alloc\_LL, 231
  - ~Osc\_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
- Osc\_Opaque\_Type\_Compare, 233
  - ~Osc\_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
- Osc\_Pair, 235
  - first, 235
  - Osc\_Pair, 235
  - second, 235
- OSCL\_PERF\_SUMMARY\_LOGGING
  - osclproc, 105



- OSCL\_PLACEMENT\_NEW
  - osclmemory, [57](#)
- oscl\_pow
  - osclutil, [75](#)
- oscl\_priqueue.h, [726](#)
- oscl\_priqueue\_test
  - OscPriorityQueue, [473](#)
- oscl\_procstatus.h, [727](#)
- Osc\_Queue, [236](#)
  - ~Osc\_Queue, [237](#)
  - back, [237](#)
  - clear, [237](#)
  - const\_reference, [237](#)
  - front, [238](#)
  - Osc\_Queue, [237](#)
  - pointer, [237](#)
  - pop, [238](#)
  - push, [238](#)
  - reference, [237](#)
  - size\_type, [237](#)
  - value\_type, [237](#)
- oscl\_queue.h, [728](#)
- Osc\_Queue\_Base, [239](#)
  - ~Osc\_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\_RANDOM\_MAX
  - osclconfig\_util.h, [847](#)
- Osc\_Rb\_Tree, [242](#)
  - ~Osc\_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](#)
  - Osc\_Rb\_Tree, [244](#)
  - pointer, [244](#)
  - reference, [244](#)
  - size, [244](#)
  - size\_type, [244](#)
  - upper\_bound, [244](#)
  - value\_type, [244](#)
- Osc\_Rb\_Tree\_Base, [246](#)
  - base\_link\_type, [246](#)
  - rebalance, [246](#)
  - rebalance\_for\_erase, [246](#)
  - rotate\_left, [246](#)
  - rotate\_right, [246](#)
- Osc\_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](#)
  - Osc\_Rb\_Tree\_Const\_Iterator, [248](#)
  - pointer, [248](#)
  - reference, [248](#)
  - self, [248](#)
  - value\_type, [248](#)
- Osc\_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](#)
  - Osc\_Rb\_Tree\_Iterator, [251](#)
  - pointer, [251](#)
  - reference, [251](#)
  - self, [251](#)
  - value\_type, [251](#)

- Osc\_Rb\_Tree\_Node, 253
  - link\_type, 253
  - value, 253
  - value\_type, 253
- Osc\_Rb\_Tree\_Node\_Base
  - black, 254
  - red, 254
- Osc\_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
- Osc\_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\_-PERRORTRAP, 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\_PERRORTRAP
  - 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 DATAGRAM
  - 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\_SOCKET\_NO\_SERV, [757](#)
  - PVSOCK\_ERR\_SOCKET\_NOT\_-
    - CONNECTED, [757](#)
  - PVSOCK\_ERR\_SOCKET\_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
  - EOscSocket\_DataRecv, [771](#)
  - EOscSocket\_DataSent, [771](#)
  - EOscSocket\_Except, [770](#)
  - EOscSocket\_OS, [770](#)
  - EOscSocket\_Readable, [770](#)
  - EOscSocket\_RequestAO\_Canceled, [770](#)
  - EOscSocket\_RequestAO\_Error, [770](#)
  - EOscSocket\_RequestAO\_Success, [770](#)
  - EOscSocket\_RequestAO\_Timeout, [770](#)
  - EOscSocket\_ServPoll, [770](#)
  - EOscSocket\_ServRequestCancelIssued, [771](#)
  - EOscSocket\_ServRequestComplete, [771](#)
  - EOscSocket\_ServRequestIssued, [770](#)
  - EOscSocket\_Writable, [770](#)
  - EOscSocketServ\_LastEvent, [770](#)
  - EOscSocketServ\_LoopsockError, [771](#)
  - EOscSocketServ\_LoopsockOk, [771](#)
  - EOscSocketServ\_SelectActivity, [770](#)
  - EOscSocketServ\_SelectNoActivity, [770](#)
  - EOscSocketServ\_SelectRescheduleAsap, [770](#)
  - EOscSocketServ\_SelectReschedulePoll, [770](#)
- oscl\_socket\_stats.h, [770](#)
  - TOscSocketServStatEvent, [770](#)
  - TOscSocketStatEvent, [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](#)
- PV\_SOCKET\_SERVER\_SELECT, [773](#)
- PV\_SOCKET\_SERVER\_SELECT\_-  
LOOPBACK\_SOCKET, [773](#)
- PV\_SOCKET\_SERVER\_SELECT\_-  
TIMEOUT\_MSEC, [773](#)
- PV\_SOCKET\_SERVER\_THREAD\_-  
PRIORITY, [773](#)
- PV\_SOCKET\_SERVI\_STATS, [773](#)
- oscl\_socket\_types.h
  - EPVIPAddMembership, [775](#)
  - EPVIPMulticastTTL, [775](#)
  - EPVIPProtoIP, [775](#)
  - EPVIPProtoTCP, [775](#)
  - EPVIPTOS, [775](#)
  - EPVSocket, [775](#)
  - EPVSocket\_Last, [775](#)
  - EPVSocketAccept, [775](#)
  - EPVSocketBind, [775](#)
  - EPVSocketBothShutdown, [775](#)
  - EPVSocketCancel, [774](#)
  - EPVSocketConnect, [775](#)
  - EPVSocketFailure, [774](#)
  - EPVSocketListen, [775](#)
  - EPVSocketNotImplemented, [775](#)
  - EPVSocketPending, [774](#)
  - EPVSocketRecv, [775](#)
  - EPVSocketRecvFrom, [775](#)
  - EPVSocketRecvShutdown, [775](#)
  - EPVSocketSend, [775](#)
  - EPVSocketSendShutdown, [775](#)
  - EPVSocketSendTo, [775](#)
  - 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\_SOCKET\_IP\_ADDMEMBERSHIP  
osclconfig\_io.h, [815](#)
- OSCL\_SOCKET\_IP\_MULTICAST\_TTL  
osclconfig\_io.h, [815](#)
- OSCL\_SOCKET\_IP\_TOS  
osclconfig\_io.h, [815](#)
- OSCL\_SOCKET\_SOL\_REUSEADDR  
osclconfig\_io.h, [815](#)
- OSCL\_SOL\_IP  
osclconfig\_io.h, [815](#)
- OSCL\_SOL\_SOCKET  
osclconfig\_io.h, [815](#)
- OSCL\_SOL\_TCP  
osclconfig\_io.h, [815](#)
- OSCL\_SOL\_UDP  
osclconfig\_io.h, [815](#)
- oscl\_sqrt  
osclutil, [76](#)
- OSCL\_StackString, [257](#)  
osclutil, [76](#), [77](#)
- OSCL\_StackString  
chartype, [258](#)  
optype, [258](#)  
OSCL\_String, [258](#)  
other\_chartype, [258](#)
- oscl\_stat  
osclio, [101](#), [102](#)
- OSCL\_STAT\_BUF  
osclio, [97](#)
- oscl\_stat\_buf, [259](#)  
mode, [259](#)  
perms, [259](#)
- oscl\_statfs  
osclio, [102](#)
- OSCL\_STATIC\_CAST  
osclbase, [34](#)
- oscl\_stdstring.h, [776](#)
- oscl\_str\_escape\_xml  
osclutil, [77](#)
- oscl\_str\_is\_valid\_utf8  
osclutil, [77](#)
- oscl\_str\_need\_escape\_xml  
osclutil, [78](#)
- oscl\_str\_ptr\_len.h, [778](#)
- oscl\_str\_truncate\_utf8  
osclutil, [78](#)
- oscl\_str\_unescape\_uri  
osclutil, [78](#), [79](#)
- oscl\_strcat  
osclbase, [39](#)
- oscl\_strchr  
osclbase, [39](#), [40](#)
- oscl\_strcmp  
osclbase, [40](#)
- OSCL\_StrError  
osclerror, [93](#)
- OSCL\_String, [260](#)  
~OSCL\_String, [261](#)  
append\_rep, [261](#)  
chartype, [261](#)  
get\_cstr, [261](#)  
get\_maxsize, [261](#)  
get\_size, [262](#)  
get\_str, [262](#)

- 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\_strncmp
  - osclbase, 41, 42
- oscl\_strncpy
  - osclbase, 42
- oscl\_strchr
  - 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](#)
- OscI\_TagTree::Node, [279](#)
- OscI\_TagTree::Node
  - children, [280](#)
  - children\_type, [280](#)
  - depth, [280](#)
  - Node, [280](#)
  - parent, [280](#)
  - sort\_children, [280](#)
  - tag, [280](#)
  - value, [280](#)
- OscI\_TAlloc, [281](#)
  - ~OscI\_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](#)
- OscI\_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](#)
  - OscIThread\_State, [788](#)
  - OscIThreadPriority, [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](#)
  - OscUuid32, [799](#)
- oscl\_uuid\_utils.h, [800](#)
  - PV\_CHAR\_CLOSE\_BRACKET, [800](#)
  - PV\_CHAR\_COMMA, [800](#)
- Osc\_Vector, [285](#)
  - ~Osc\_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](#)
  - Osc\_Vector, [286](#)
  - pointer, [286](#)
  - pop\_back, [288](#)
  - push\_back, [289](#)
  - push\_front, [289](#)
  - reference, [286](#)
  - value\_type, [286](#)
- oscl\_vector.h, [801](#)
- Osc\_Vector\_Base, [290](#)
  - ~Osc\_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](#)
  - OscPriorityQueueBase, [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
- OscIAccept
  - osclconfig\_io.h, 815
- OscIAcceptMethod, 308
- OscIAcceptMethod
  - ~OscIAcceptMethod, 308
  - Accept, 308
  - AcceptRequest, 308
  - DiscardAcceptedSocket, 308
  - GetAcceptedSocket, 308
  - NewL, 308
- OscIAcceptRequest, 309
  - OscIAcceptRequest, 309
  - OscISocketI, 539
- OscIAcceptRequest
  - Accept, 309
  - OscIAcceptRequest, 309
- OscIActiveObject, 310
  - EPriorityHigh, 311
  - EPriorityHighest, 311
  - EPriorityIdle, 311
  - EPriorityLow, 311
  - EPriorityNominal, 311
  - OscIActiveObject, 311
  - OscIExecSchedulerCommonBase, 397
  - PVActiveBase, 615
  - PVActiveStats, 616
  - PVThreadContext, 635
- OscIActiveObject
  - ~OscIActiveObject, 311
  - AddToScheduler, 311
  - Cancel, 311
  - DoCancel, 312
  - IsBusy, 312
  - OscIActiveObject, 311
  - OscIActivePriority, 311



- PendComplete, [312](#)
- PendForExec, [312](#)
- Priority, [312](#)
- RemoveFromScheduler, [312](#)
- RunError, [312](#)
- RunIfNotReady, [313](#)
- SetBusy, [313](#)
- SetStatus, [313](#)
- Status, [313](#)
- StatusRef, [313](#)
- OslActivePriority
  - OslActiveObject, [311](#)
- OslAllocDestructDealloc, [314](#)
- OslAllocDestructDealloc
  - ~OslAllocDestructDealloc, [314](#)
- OslAny
  - osclbase, [35](#)
- OslAOSStatus, [315](#)
  - OslAOSStatus, [315](#)
- OslAOSStatus
  - operator!=, [315](#)
  - operator<, [315](#)
  - operator<=, [315](#)
  - operator=, [315](#)
  - operator==, [315](#)
  - operator>, [315](#)
  - operator>=, [315](#)
  - OslAOSStatus, [315](#)
  - Value, [315](#)
- OslAsyncFile, [316](#)
- OslAsyncFile
  - ~OslAsyncFile, [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](#)
- OslAsyncFileBuffer, [319](#)
- OslAsyncFileBuffer
  - ~OslAsyncFileBuffer, [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](#)
- OslAuditCB, [321](#)
  - OslAuditCB, [321](#)
- OslAuditCB
  - OslAuditCB, [321](#)
  - pAudit, [321](#)
  - pStatsNode, [321](#)
- OslBase
  - OslSingletonRegistry, [534](#)
  - OslTLSRegistry, [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\_strcat, [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\_PERRORTRAP, [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](#)
- TOsclTlsKey, [35](#)
- uint, [35](#)
- uint64, [35](#)
- unix\_ntp\_offset, [46](#)
- USEC\_PER\_SEC, [46](#)
- OsclBasicDateTimeStruct  
    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](#)

- OslBinOStream, [331](#)
- OslBinOStream
  - ~OslBinOStream, [331](#)
  - OslBinOStream, [331](#)
  - write, [331](#)
- OslBinOStreamBigEndian, [332](#)
  - OslBinOStreamBigEndian, [333](#)
- OslBinOStreamBigEndian
  - operator<<, [333](#)
  - OslBinOStreamBigEndian, [333](#)
  - WriteUnsignedLong, [333](#)
  - WriteUnsignedShort, [333](#)
- OslBinOStreamLittleEndian, [334](#)
  - OslBinOStreamLittleEndian, [335](#)
- OslBinOStreamLittleEndian
  - operator<<, [335](#)
  - OslBinOStreamLittleEndian, [335](#)
  - WriteUnsignedLong, [335](#)
  - WriteUnsignedShort, [335](#)
- OslBinStream, [336](#)
  - EOF\_STATE, [337](#)
  - FAIL\_STATE, [337](#)
  - GOOD\_STATE, [337](#)
  - OslBinStream, [337](#)
- OslBinStream
  - Attach, [337](#)
  - eof, [337](#)
  - fail, [338](#)
  - firstFragPtr, [339](#)
  - fragsLeft, [339](#)
  - good, [338](#)
  - HaveRoomInCurrentBlock, [338](#)
  - length, [339](#)
  - nextFragPtr, [339](#)
  - numFrag, [339](#)
  - OslBinStream, [337](#)
  - pBasePosition, [339](#)
  - PositionInBlock, [338](#)
  - pPosition, [339](#)
  - ReserveSpace, [338](#)
  - Seek, [338](#)
  - seekFromCurrentPosition, [338](#)
  - specialFragBuffer, [339](#)
  - state, [339](#)
  - state\_t, [337](#)
  - tellg, [338](#)
- OslBuf, [340](#)
  - OslBuf, [341](#)
- OslBuf
  - Delete, [341](#)
  - Des, [341](#)
  - DesC, [341](#)
  - iBuffer, [341](#)
  - iLength, [341](#)
  - iMaxLength, [341](#)
  - Length, [341](#)
  - NewL, [341](#)
  - OslBuf, [341](#)
- OslCloseSocket
  - osclconfig\_io.h, [816](#)
- OslCoeActiveScheduler
  - OslExecSchedulerBase, [391](#)
  - OslExecSchedulerCommonBase, [397](#)
  - PVThreadContext, [635](#)
- OslCoeActiveSchedulerBase
  - PVThreadContext, [635](#)
- OslCompareLess, [342](#)
- OslCompareLess
  - compare, [342](#)
- OslComponentFactory
  - osclutil, [69](#)
- OslComponentRegistry, [343](#)
  - OslComponentRegistry, [344](#)
- OslComponentRegistry
  - ~OslComponentRegistry, [344](#)
  - CloseSession, [344](#)
  - FindExact, [344](#)
  - FindHierarchical, [344](#)
  - iComponentIdCounter, [344](#)
  - iData, [344](#)
  - iMutex, [344](#)
  - iNumSessions, [344](#)
  - OpenSession, [344](#)
  - OslComponentRegistry, [344](#)
  - Register, [344](#)
  - Unregister, [344](#)
- OslComponentRegistryData, [345](#)
- OslComponentRegistryData
  - Find, [345](#)
  - iVec, [345](#)
- OslComponentRegistryElement, [346](#)
  - OslComponentRegistryElement, [346](#)
- OslComponentRegistryElement
  - ~OslComponentRegistryElement, [346](#)
  - iComponentId, [346](#)
  - iFactory, [346](#)
  - iId, [346](#)
  - Match, [346](#)
  - operator=, [346](#)
  - OslComponentRegistryElement, [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\_SOCKET\_IP\_-
  - ADDMEMBERSHIP, [815](#)
- OSCL\_SOCKET\_IP\_MULTICAST\_-
  - TTL, [815](#)
- OSCL\_SOCKET\_IP\_TOS, [815](#)
- OSCL\_SOCKET\_SOL\_REUSEADDR,
  - [815](#)
- OSCL\_SOL\_IP, [815](#)
- OSCL\_SOL\_SOCKET, [815](#)
- OSCL\_SOL\_TCP, [815](#)
- OSCL\_SOL\_UDP, [815](#)
- OscIAccept, [815](#)
- OscIBind, [816](#)
- OscICloseSocket, [816](#)
- OscIConnect, [816](#)
- OscIConnectComplete, [816](#)
- OscIGetAsyncSockErr, [816](#)
- OscIGetDottedAddr, [816](#)
- OscIGetDottedAddrVector, [817](#)
- OscIGethostbyname, [817](#)
- OscIGetPeerName, [817](#)
- OscIJoin, [817](#)
- OscIListen, [818](#)
- OscIMakeInAddr, [818](#)
- OscIMakeSockAddr, [818](#)
- OscIPipe, [818](#)
- OscIReadFD, [818](#)
- OscIRecv, [818](#)
- OscIRecvFrom, [818](#)
- OscISend, [819](#)
- OscISendTo, [819](#)
- OscISetNonBlocking, [819](#)
- OscISetRecvBufferSize, [819](#)
- OscISetSockOpt, [819](#)
- OscIShutdown, [819](#)
- OscISocket, [820](#)
- OscISocketCleanup, [820](#)
- OscISocketSelect, [820](#)
- OscISocketStartup, [820](#)
- OscIUnMakeInAddr, [820](#)
- OscIUnMakeSockAddr, [821](#)
- OscIValidInetAddr, [821](#)
- OscIWriteFD, [821](#)
- TIpMReq, [821](#)
- TOscIFileOffset, [821](#)
- TOscIHostent, [821](#)
- TOscISockAddr, [821](#)
- TOscISockAddrLen, [821](#)
- TOscISocket, [821](#)
- osclconfig\_io\_check.h, [822](#)
  - \_\_verify\_\_TOscIFileOffset\_\_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\_\_TOscIConditionObject\_\_defined\_\_, [831](#)
  - \_\_verify\_\_TOscIMutexObject\_\_defined\_\_, [831](#)
  - \_\_verify\_\_TOscISemaphoreObject\_\_defined\_\_, [831](#)
  - \_\_verify\_\_TOscIThreadFuncArg\_\_defined\_\_, [831](#)
  - \_\_verify\_\_TOscIThreadFuncRet\_\_defined\_\_, [831](#)
  - \_\_verify\_\_TOscIThreadId\_\_defined\_\_, [831](#)
  - \_\_verify\_\_TOscIThreadObject\_\_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](#)
  - TOscIConditionObject, [834](#)
  - TOscIMutexObject, [834](#)
  - TOscISemaphoreObject, [834](#)
  - TOscIThreadFuncArg, [834](#)
  - TOscIThreadFuncRet, [834](#)
  - TOscIThreadId, [834](#)
  - TOscIThreadObject, [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
  - OscBasicDateTimeStruct, 837
  - OscBasicTimeStruct, 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\_RANDOM\_MAX, [847](#)
- SLEEP\_ONE\_SEC, [847](#)
- osclconfig\_util\_check.h, [848](#)
- OscConnect
  - osclconfig\_io.h, [816](#)
- OscConnectComplete
  - osclconfig\_io.h, [816](#)
- OscConnectMethod, [348](#)
- OscConnectMethod
  - ~OscConnectMethod, [348](#)
  - Connect, [348](#)
  - ConnectRequest, [348](#)
  - NewL, [348](#)
- OscConnectRequest, [349](#)
  - OscConnectRequest, [349](#)
  - OscSocketI, [539](#)
- OscConnectRequest
  - Connect, [349](#)
  - OscConnectRequest, [349](#)
- OscDestructDealloc, [350](#)
- OscDestructDealloc
  - ~OscDestructDealloc, [350](#)
  - destruct\_and\_dealloc, [350](#)
- OscDNS, [351](#)
  - OscSocketServ, [555](#)
- OscDNS
  - ~OscDNS, [351](#)
  - CancelGetHostByName, [351](#)
  - GetHostByName, [352](#)
  - NewL, [352](#)
  - OscDNSRequestAO, [352](#)
- OscDNSI, [353](#)
  - OscDNSRequestAO, [365](#)
  - OscSocketServI, [557](#)
- OscDNSI
  - ~OscDNSI, [353](#)
  - Close, [353](#)
  - DNSRequestParam, [354](#)
  - GetHostByName, [353](#)
  - GetHostByNameResponseContainsAlias-Info, [354](#)
  - GetHostByNameSuccess, [354](#)
  - GetNextHost, [354](#)
  - GetNextHostSuccess, [354](#)
  - NewL, [354](#)
  - Open, [354](#)
  - OscDNSRequest, [354](#)
  - OscGetHostByNameRequest, [354](#)
- OscDNSIBase, [355](#)
  - OscDNSIBase, [356](#)
- OscDNSIBase
  - ~OscDNSIBase, [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](#)
  - OscDNSIBase, [356](#)
  - OscDNSRequest, [357](#)
  - OscGetHostByNameRequest, [357](#)
- OscDNSMethod, [358](#)
  - OscDNSMethod, [359](#)
  - OscDNSRequestAO, [365](#)
- OscDNSMethod
  - Abort, [359](#)
  - AbortAll, [359](#)
  - CancelMethod, [359](#)
  - ConstructL, [359](#)
  - iAlloc, [360](#)
  - iDNSFxn, [360](#)
  - iDNSObserver, [360](#)
  - iDNSRequestAO, [360](#)
  - iId, [360](#)
  - iLogger, [360](#)
  - MethodDone, [359](#)
  - OscDNSMethod, [359](#)
  - Run, [359](#)
  - StartMethod, [359](#)
- OscDNSObserver, [361](#)
- OscDNSObserver
  - ~OscDNSObserver, [361](#)
  - HandleDNSEvent, [361](#)
- OscDNSRequest, [362](#)
  - OscDNSI, [354](#)
  - OscDNSIBase, [357](#)
  - OscDNSRequest, [362](#)
  - OscDNSRequestAO, [365](#)
- OscDNSRequest
  - ~OscDNSRequest, [362](#)
  - Activate, [362](#)
  - CancelRequest, [362](#)
  - Complete, [362](#)
  - iActive, [362](#)
  - iDNSRequestAO, [362](#)
  - iDNSRequestParam, [362](#)
  - OscDNSRequest, [362](#)

- OscIDNSRequestAO, [363](#)
  - OscIDNS, [352](#)
  - OscIDNSRequestAO, [364](#)
- OscIDNSRequestAO
  - Abort, [364](#)
  - Cancelled, [364](#)
  - ConstructL, [364](#)
  - DoCancel, [364](#)
  - Failure, [364](#)
  - GetHostByNameParam, [365](#)
  - GetSocketError, [364](#)
  - iDNSI, [365](#)
  - iDNSMethod, [365](#)
  - iLogger, [365](#)
  - iSocketError, [365](#)
  - NewRequest, [364](#)
  - OscIDNSI, [365](#)
  - OscIDNSMethod, [365](#)
  - OscIDNSRequest, [365](#)
  - OscIDNSRequestAO, [364](#)
  - RequestDone, [364](#)
  - Run, [364](#)
  - Serv, [365](#)
  - Success, [365](#)
- OscDoubleLink, [366](#)
  - OscDoubleLink, [366](#)
- OscDoubleLink
  - iNext, [366](#)
  - InsertAfter, [366](#)
  - InsertBefore, [366](#)
  - iPrev, [366](#)
  - OscDoubleLink, [366](#)
  - Remove, [366](#)
- OscDoubleList, [367](#)
  - OscDoubleList, [367](#)
- OscDoubleList
  - Head, [367](#)
  - InsertHead, [367](#)
  - InsertTail, [367](#)
  - IsHead, [367](#)
  - IsTail, [367](#)
  - OscDoubleList, [367](#)
  - Tail, [367](#)
- OscDoubleListBase, [368](#)
  - OscDoubleListBase, [369](#)
- OscDoubleListBase
  - getHead, [369](#)
  - getOffset, [369](#)
  - iHead, [369](#)
  - Insert, [369](#)
  - InsertHead, [369](#)
  - InsertTail, [369](#)
  - iOffset, [369](#)
  - IsEmpty, [369](#)
- OscDoubleListBase, [369](#)
  - Reset, [369](#)
  - SetOffset, [369](#)
- OscDoubleRunner, [370](#)
  - OscDoubleRunner, [370](#)
- OscDoubleRunner
  - iHead, [370](#)
  - iNext, [370](#)
  - iOffset, [370](#)
  - operator T \*, [370](#)
  - operator++, [370](#)
  - operator--, [370](#)
  - OscDoubleRunner, [370](#)
  - Set, [370](#)
  - SetToHead, [370](#)
  - SetToTail, [370](#)
- OscErrAlreadyExists
  - osclerror, [92](#)
- OscErrAlreadyInstalled
  - osclerror, [92](#)
- OscErrArgument
  - osclerror, [92](#)
- OscErrBadHandle
  - osclerror, [92](#)
- OscErrBusy
  - osclerror, [92](#)
- OscErrCancelled
  - osclerror, [92](#)
- OscErrCorrupt
  - osclerror, [92](#)
- OscErrGeneral
  - osclerror, [92](#)
- OscErrInvalidState
  - osclerror, [92](#)
- OscErrNoHandler
  - osclerror, [92](#)
- OscErrNoMemory
  - osclerror, [92](#)
- OscErrNone
  - osclerror, [92](#)
- OscErrNoResources
  - osclerror, [92](#)
- OscErrNotInstalled
  - osclerror, [92](#)
- OscErrNotReady
  - osclerror, [92](#)
- OscErrNotSupported
  - osclerror, [92](#)
- OscError, [372](#)
  - OscErrorTrapImp, [378](#)
  - OscExecSchedulerCommonBase, [397](#)
  - OscTrapStack, [599](#)
- OscError
  - 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](#)
  - OscErrAlreadyExists, [92](#)
  - OscErrAlreadyInstalled, [92](#)
  - OscErrArgument, [92](#)
  - OscErrBadHandle, [92](#)
  - OscErrBusy, [92](#)
  - OscErrCancelled, [92](#)
  - OscErrCorrupt, [92](#)
  - OscErrGeneral, [92](#)
  - OscErrInvalidState, [92](#)
  - OscErrNoHandler, [92](#)
  - OscErrNoMemory, [92](#)
  - OscErrNone, [92](#)
  - OscErrNoResources, [92](#)
  - OscErrNotInstalled, [92](#)
  - OscErrNotReady, [92](#)
  - OscErrNotSupported, [92](#)
  - OscErrOverflow, [92](#)
  - OscErrSystemCallFailed, [92](#)
  - OscErrThreadContextIncorrect, [92](#)
  - OscErrTimeout, [92](#)
  - OscErrUnderflow, [92](#)
  - OscFailure, [92](#)
  - OscLeaveCode, [93](#)
  - OscPending, [92](#)
  - OscReturnCode, [93](#)
  - OscSuccess, [92](#)
  - OscTrapOperation, [93](#)
  - PVError\_DoLeave, [92](#)
  - PVEERROR\_IMP\_JUMPS, [92](#)
  - PVEERRORTRAP\_REGISTRY, [92](#)
  - PVEERRORTRAP\_REGISTRY\_ID, [93](#)
- OscErrorAllocator, [374](#)
  - OscErrorAllocator, [374](#)
- OscErrorAllocator
  - allocate, [374](#)
  - deallocate, [374](#)
  - operator delete, [375](#)
  - operator new, [375](#)
  - OscErrorAllocator, [374](#)
- OscErrorTrap, [376](#)
  - OscErrorTrapImp, [378](#)
  - OscTrapStack, [599](#)
- OscErrorTrap
  - Cleanup, [376](#)
  - GetErrorTrapImp, [376](#)
  - Init, [376](#)
- OscErrorTrapImp, [377](#)
  - OscJump, [421](#)
  - OscTrapStack, [599](#)
- OscErrorTrapImp
  - CPVInterfaceProxy, [378](#)
  - iJumpData, [378](#)
  - iLeave, [378](#)
  - iTrapStack, [378](#)
  - OscError, [378](#)
  - OscErrorTrap, [378](#)
  - OscExecScheduler, [378](#)
  - OscExecSchedulerCommonBase, [378](#)
  - OscJump, [378](#)
  - OscJumpMark, [378](#)
  - OscScheduler, [378](#)
  - OscTrapStack, [378](#)
  - Trap, [377](#)
  - TrapNoTls, [377](#)
  - UnTrap, [377](#)
- OscErrOverflow
  - osclerror, [92](#)
- OscErrSystemCallFailed
  - osclerror, [92](#)
- OscErrThreadContextIncorrect
  - osclerror, [92](#)
- OscErrTimeout
  - osclerror, [92](#)
- OscErrUnderflow
  - osclerror, [92](#)
- OscException, [379](#)
  - OscException, [379](#)
- OscException
  - getLeaveCode, [379](#)

- OslException, 379
- OslExclusiveArrayPtr, 380
  - OslExclusiveArrayPtr, 381
- OslExclusiveArrayPtr
  - ~OslExclusiveArrayPtr, 381
  - \_Ptr, 382
  - get, 381
  - operator \*, 381
  - operator->, 381
  - operator=, 381
  - OslExclusiveArrayPtr, 381
  - release, 382
  - set, 382
- OslExclusivePtr, 383
  - OslExclusivePtr, 384
- OslExclusivePtr
  - ~OslExclusivePtr, 384
  - \_Ptr, 385
  - get, 384
  - operator \*, 384
  - operator->, 384
  - operator=, 384
  - OslExclusivePtr, 384
  - release, 385
  - set, 385
- OslExclusivePtrA, 386
  - OslExclusivePtrA, 387
- OslExclusivePtrA
  - ~OslExclusivePtrA, 387
  - \_Ptr, 388
  - get, 387
  - operator \*, 387
  - operator->, 387
  - operator=, 387
  - OslExclusivePtrA, 387
  - release, 388
  - set, 388
- OslExecScheduler, 389
  - OslErrorTrapImp, 378
  - OslExecSchedulerBase, 391
  - OslExecSchedulerCommonBase, 397
  - PVActiveBase, 615
  - PVActiveStats, 616
  - PVThreadContext, 635
- OslExecScheduler
  - Current, 389
  - OslScheduler, 390
  - RegisterForCallback, 389
  - RunSchedulerNonBlocking, 389
- OslExecSchedulerBase, 391
  - PVThreadContext, 635
- OslExecSchedulerBase
  - OslCoeActiveScheduler, 391
  - OslExecScheduler, 391
- PVActiveBase, 391
- OslExecSchedulerCommonBase, 392
  - EOtherExecStats\_Last, 394
  - EOtherExecStats\_NativeOS, 394
  - EOtherExecStats\_QueueTime, 394
  - EOtherExecStats\_ReleaseTime, 394
  - EOtherExecStats\_WaitTime, 394
  - OslErrorTrapImp, 378
  - OslExecSchedulerCommonBase, 395
  - PVActiveStats, 616
  - PVThreadContext, 635
- OslExecSchedulerCommonBase
  - ~OslExecSchedulerCommonBase, 395
  - AddToExecTimerQ, 395
  - BeginScheduling, 395
  - BeginStats, 395
  - BlockingLoopL, 395
  - CallRunExec, 395
  - CleanupExecQ, 395
  - CleanupStatQ, 395
  - ConstructL, 395
  - ConstructStatQ, 395
  - EndScheduling, 395
  - EndStats, 395
  - Error, 395
  - FindPVBBase, 395
  - GetId, 395
  - GetName, 395
  - GetScheduler, 395
  - iAlloc, 399
  - iBlockingMode, 399
  - iDebugLogger, 399
  - iDefAlloc, 399
  - iDelta, 399
  - iDoStop, 399
  - iDoSuspend, 399
  - iErrorTrapImp, 399
  - iExecTimerQ, 399
  - iGrandTotalTicks, 399
  - iLogger, 399
  - iLogPerfIndentStr, 399
  - iLogPerfIndentStrLen, 399
  - iLogPerfTotal, 399
  - iName, 399
  - iNativeMode, 399
  - IncLogPerf, 396
  - InitExecQ, 396
  - InstallScheduler, 396
  - iNumAOAdded, 399
  - iOtherExecStats, 399
  - iPVStatQ, 399
  - iPVStats, 399
  - iReadyQ, 399
  - iResumeSem, 399

- IsInstalled, 396
- IsStarted, 396
- iStopper, 399
- iStopperCrit, 399
- iSuspended, 399
- iThreadContext, 399
- iTime, 399
- iTimeCompareThreshold, 399
- iTotalPercent, 399
- iTotalTicksTemp, 399
- OscActiveObject, 397
- OscCoeActiveScheduler, 397
- OscError, 397
- OscExecScheduler, 397
- OscExecSchedulerCommonBase, 395
- OscReadyQ, 397
- OscScheduler, 397
- OscTimerCompare, 397
- OscTimerObject, 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
- OscExtractFilenameFromFullpath
  - OscFileManager, 407
- OscFailure
  - osclerror, 92
- OscFileCache, 401
  - Osc\_File, 186
  - OscFileCache, 402
- OscFileCache
  - ~OscFileCache, 402
  - \_fixedCaches, 402
  - \_movableCache, 402
  - AddFixedCache, 402
  - Close, 402
  - EndOfFile, 402
  - FileSize, 402
  - Flush, 402
  - Open, 402
  - OscFileCache, 402
  - OscFileCacheBuffer, 402
  - Read, 402
  - Seek, 402
  - Tell, 402
  - Write, 402
- OscFileCacheBuffer, 403
  - Osc\_File, 186
  - OscFileCache, 402
  - OscFileCacheBuffer, 404
- OscFileCacheBuffer
  - capacity, 404
  - Contains, 404
  - currentPos, 404
  - endPos, 404
  - filePosition, 404
  - FillFromFile, 404
  - iContainer, 404
  - isFixed, 404
  - IsUpdated, 404
  - OscFileCacheBuffer, 404
  - pBuffer, 404
  - Preceeds, 404
  - PrepRead, 404
  - PrepWrite, 404
  - SetPosition, 404
  - updateEnd, 404
  - updateStart, 404
  - usableSize, 404
  - WriteUpdatesToFile, 404
- OscFileHandle, 405
  - OscFileHandle, 405
- OscFileHandle
  - Handle, 405
  - Osc\_File, 405
  - OscFileHandle, 405
- OscFileManager, 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
- OscFileManager
  - OSCL\_FILE\_ATTRIBUTE\_TYPE, 406
  - OscExtractFilenameFromFullpath, 407
  - OscGetFileAttributes, 407

- OscGetFileCreationTime, [407, 408](#)
  - OscGetFileLastAccessTime, [408](#)
  - OscGetFileLastWriteTime, [409](#)
  - OscGetFileSize, [409](#)
- OscFileStats, [411](#)
  - OscFileStats, [411](#)
- OscFileStats
  - End, [411](#)
  - Log, [411](#)
  - LogAll, [411](#)
  - OscFileStats, [411](#)
  - Start, [411](#)
- OscFileStatsItem, [412](#)
- OscFileStatsItem
  - iOpCount, [412](#)
  - iParam, [412](#)
  - iParam2, [412](#)
  - iStartTick, [412](#)
  - iTotalTicks, [412](#)
- OscFloat
  - osclbase, [35](#)
- OscGetAsyncSockErr
  - osclconfig\_io.h, [816](#)
- OscGetDottedAddr
  - osclconfig\_io.h, [816](#)
- OscGetDottedAddrVector
  - osclconfig\_io.h, [817](#)
- OscGetFileAttributes
  - OscFileManager, [407](#)
- OscGetFileCreationTime
  - OscFileManager, [407, 408](#)
- OscGetFileLastAccessTime
  - OscFileManager, [408](#)
- OscGetFileLastWriteTime
  - OscFileManager, [409](#)
- OscGetFileSize
  - OscFileManager, [409](#)
- OscGethostbyname
  - osclconfig\_io.h, [817](#)
- OscGetHostByNameMethod, [413](#)
  - OscGetHostByNameRequest, [414](#)
- OscGetHostByNameMethod
  - ~OscGetHostByNameMethod, [413](#)
  - GetHostByName, [413](#)
  - NewL, [413](#)
- OscGetHostByNameRequest, [414](#)
  - OscIDNSI, [354](#)
  - OscIDNSIBase, [357](#)
- OscGetHostByNameRequest
  - OscGetHostByNameMethod, [414](#)
- OscGetPeerName
  - osclconfig\_io.h, [817](#)
- OscInit, [415](#)
- OscInit
  - Cleanup, [415](#)
  - Init, [415](#)
- OscInteger64Transport, [416](#)
- OscInteger64Transport
  - iHigh, [416](#)
  - iLow, [416](#)
- osclio
  - EOscFileOp\_Close, [98](#)
  - EOscFileOp\_EndOfFile, [98](#)
  - EOscFileOp\_Flush, [98](#)
  - EOscFileOp\_Last, [99](#)
  - EOscFileOp\_NativeClose, [98](#)
  - EOscFileOp\_NativeEndOfFile, [99](#)
  - EOscFileOp\_NativeFlush, [99](#)
  - EOscFileOp\_NativeOpen, [98](#)
  - EOscFileOp\_NativeRead, [98](#)
  - EOscFileOp\_NativeSeek, [99](#)
  - EOscFileOp\_NativeSetSize, [99](#)
  - EOscFileOp\_NativeSize, [99](#)
  - EOscFileOp\_NativeTell, [99](#)
  - EOscFileOp\_NativeWrite, [99](#)
  - EOscFileOp\_Open, [98](#)
  - EOscFileOp\_Read, [98](#)
  - EOscFileOp\_Seek, [98](#)
  - EOscFileOp\_SetSize, [98](#)
  - EOscFileOp\_Size, [98](#)
  - EOscFileOp\_Tell, [98](#)
  - EOscFileOp\_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
- TOscIFileHandle, 97
- TOscIFileOffsetInt32, 97
- TOscIFileOp, 98
- TPVDNSEvent, 99
- TPVDNSFxn, 99
- OscIPMReq, 417
  - OscIPMReq, 417
- OscIPMReq
  - interfaceAddr, 417
  - multicastAddr, 417
  - OscIPMReq, 417
- OscIPSocketI, 418
  - OscIPSocketI, 419
- OscIPSocketI
  - ~OscIPSocketI, 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
  - OscIPSocketI, 419
  - OscIsocketMethod, 420
  - OscIsocketRequestAO, 420
  - SetOptionToReuseAddress, 419
  - SetRecvBufferSize, 419
  - SetTOS, 419
  - SocketServ, 419
  - ThreadLogoff, 419
  - ThreadLogon, 419
- OscIJoin
  - osclconfig\_io.h, 817
- OscIJump, 421
  - OscIErrorTrapImp, 378
- OscIJump
  - ~OscIJump, 421
  - Jump, 421
  - OscIErrorTrapImp, 421
  - StaticJump, 421
  - Top, 421
- OscIJumpMark
  - OscIErrorTrapImp, 378
- OscILeaveCode
  - osclerror, 93
- OscIListen
  - osclconfig\_io.h, 818
- OscIListenMethod, 422
- OscIListenMethod
  - ~OscIListenMethod, 422
  - Listen, 422
  - ListenRequest, 422
  - NewL, 422
- OscIListenRequest, 423
  - OscIListenRequest, 423
- OscIListenRequest
  - Listen, 423
  - OscIListenRequest, 423
- OscILockBase, 424
- OscILockBase
  - ~OscILockBase, 424
  - Lock, 424
  - Unlock, 424
- OscIMakeInAddr
  - osclconfig\_io.h, 818
- OscIMakeSockAddr
  - osclconfig\_io.h, 818
- OscIMem, 425
  - OscIMemGlobalAuditObject, 441
- OscIMem
  - Cleanup, 425
  - Init, 425
- OscIMemAllocator, 426
- OscIMemAllocator
  - allocate, 426
  - allocate\_fl, 426
  - deallocate, 426
- OscIMemAllocDestructDealloc, 427
- OscIMemAllocDestructDealloc

- allocate, [427](#)
- allocate\_fl, [427](#)
- deallocate, [427](#)
- destruct\_and\_dealloc, [427](#)
- OscMemAudit, [429](#)
  - OscMemAudit, [429](#)
- OscMemAudit
  - ~OscMemAudit, [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\_GetTagNode, [432](#)
  - MM\_GetTreeNode, [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](#)
  - OscMemAudit, [429](#)
  - OscMemGlobalAuditObject, [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](#)
- OscMemBasicAllocator, [439](#)
- OscMemBasicAllocator
  - allocate, [439](#)
  - deallocate, [439](#)
- OscMemBasicAllocDestructDealloc, [440](#)
- OscMemBasicAllocDestructDealloc
  - allocate, [440](#)
  - deallocate, [440](#)
  - destruct\_and\_dealloc, [440](#)
- OscMemGlobalAuditObject, [441](#)
  - OscMemAudit, [434](#)
- OscMemGlobalAuditObject
  - audit\_type, [441](#)
  - getGlobalMemAuditObject, [441](#)
  - OscMem, [441](#)
- OscMemInit
  - 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
- OscMemInit, 62
- OscMemStatsNodeAutoPtr, 59
- OscTagTreeType, 59
- TagTree\_Allocator, 59
- OscMemoryFragment, 442
- OscMemoryFragment
  - len, 442
  - ptr, 442
- OscMemPoolFixedChunkAllocator, 443
  - OscMemPoolFixedChunkAllocator, 444
- OscMemPoolFixedChunkAllocator
  - ~OscMemPoolFixedChunkAllocator, 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
  - OscMemPoolFixedChunkAllocator, 444
  - removeRef, 445
- OscMemPoolFixedChunkAllocatorObserver, 447
- OscMemPoolFixedChunkAllocatorObserver
  - ~OscMemPoolFixedChunkAllocatorObserver, 447
  - freechunkavailable, 447
- OscMemPoolResizableAllocator, 448
  - OscMemPoolResizableAllocator, 449
- OscMemPoolResizableAllocator
  - ~OscMemPoolResizableAllocator, 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](#)
- OsciMemPoolResizableAllocator, [449](#)
- removeRef, [452](#)
- setMaxSzForNewMemPoolBuffer, [452](#)
- trim, [452](#)
- validateblock, [452](#)
- OsciMemPoolResizableAllocator::MemPoolBlockInfo, [454](#)
- OsciMemPoolResizableAllocator::MemPool-BlockInfo
  - iBlockBuffer, [454](#)
  - iBlockPostFence, [454](#)
  - iBlockPreFence, [454](#)
  - iBlockSize, [454](#)
  - iNextFreeBlock, [454](#)
  - iParentBuffer, [454](#)
  - iPrevFreeBlock, [454](#)
- OsciMemPoolResizableAllocator::MemPoolBufferInfo, [455](#)
- OsciMemPoolResizableAllocator::MemPool-BufferInfo
  - iAllocatedSz, [455](#)
  - iBufferPostFence, [455](#)
  - iBufferPreFence, [455](#)
  - iBufferSize, [455](#)
  - iEndAddr, [455](#)
  - iNextFreeBlock, [455](#)
  - iNumOutstanding, [455](#)
  - iStartAddr, [455](#)
- OsciMemPoolResizableAllocatorMemoryObserver, [456](#)
- OsciMemPoolResizableAllocatorMemory-Observer
  - ~OsciMemPoolResizableAllocatorMemoryObserver, [456](#)
  - freememoryavailable, [456](#)
- OsciMemPoolResizableAllocatorObserver, [457](#)
- OsciMemPoolResizableAllocatorObserver
  - ~OsciMemPoolResizableAllocatorObserver, [457](#)
  - freeblockavailable, [457](#)
- OsciMemStatsNode, [458](#)
- OsciMemStatsNode
  - OsciMemStatsNode, [458](#)
- OsciMemStatsNode
  - ~OsciMemStatsNode, [458](#)
  - operator delete, [458](#)
  - operator new, [458](#)
  - OsciMemStatsNode, [458](#)
  - pMMFIPParam, [458](#)
  - pMMStats, [458](#)
  - reset, [458](#)
  - tag, [458](#)
- OsciMemStatsNodeAutoPtr
  - osclmemory, [59](#)
- OsciMutex, [459](#)
- OsciMutex
  - OsciMutex, [459](#)
- OsciMutex
  - ~OsciMutex, [459](#)
  - Close, [459](#)
  - Create, [459](#)
  - Lock, [460](#)
  - OsciMutex, [459](#)
  - TryLock, [460](#)
  - Unlock, [460](#)
- OsciNameString, [461](#)
- OsciNameString
  - OsciNameString, [461](#)
- OsciNameString
  - MaxLen, [461](#)
  - OsciNameString, [461](#)
  - Set, [461](#)
  - Str, [461](#)
- OsciNativeFile, [462](#)
- OsciNativeFile
  - Osci\_FileServer, [194](#)
  - OsciNativeFile, [463](#)
- OsciNativeFile
  - ~OsciNativeFile, [463](#)
  - Close, [463](#)
  - EndOfFile, [463](#)
  - Flush, [463](#)



- GetError, [463](#)
- GetReadAsyncNumElements, [463](#)
- HasAsyncRead, [463](#)
- Mode, [463](#)
- Open, [463](#)
- OscNativeFile, [463](#)
- Read, [463](#)
- ReadAsync, [463](#)
- ReadAsyncCancel, [463](#)
- Seek, [464](#)
- SetSize, [464](#)
- Size, [464](#)
- Tell, [464](#)
- Write, [464](#)
- OscNativeFileParams, [465](#)
  - OscNativeFileParams, [465](#)
- OscNativeFileParams
  - iAsyncReadBufferSize, [465](#)
  - iNativeAccessMode, [465](#)
  - iNativeBufferSize, [465](#)
  - OscNativeFileParams, [465](#)
- OscNetworkAddress, [466](#)
  - OscNetworkAddress, [466](#)
- OscNetworkAddress
  - ipAddr, [466](#)
  - operator==, [466](#)
  - OscNetworkAddress, [466](#)
  - port, [466](#)
- OscNoYieldMutex
  - oscl\_mutex.h, [723](#)
- OscNullLock, [467](#)
- OscNullLock
  - ~OscNullLock, [467](#)
  - Lock, [467](#)
  - Unlock, [467](#)
- OscPending
  - osclerror, [92](#)
- OscPipe
  - osclconfig\_io.h, [818](#)
- OscPriorityLink, [468](#)
- OscPriorityLink
  - iPriority, [468](#)
- OscPriorityList, [469](#)
  - OscPriorityList, [469](#)
- OscPriorityList
  - Head, [469](#)
  - Insert, [469](#)
  - IsHead, [469](#)
  - IsTail, [469](#)
  - OscPriorityList, [469](#)
  - Tail, [469](#)
- OscPriorityQueue, [470](#)
  - OscPriorityQueue, [471](#)
- OscPriorityQueue
  - ~OscPriorityQueue, [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\_priqueue\_test, [473](#)
  - OscPriorityQueue, [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](#)
- OscPriorityQueueBase, [474](#)
  - Osc\_Vector\_Base, [293](#)
- OscPriorityQueueBase
  - ~OscPriorityQueueBase, [474](#)
  - construct, [474](#)
  - find\_heap, [474](#)
  - pop\_heap, [474](#)
  - push\_heap, [474](#)
  - remove, [474](#)
- osclproc
  - EPVThreadContext\_InThread, [106](#)
  - EPVThreadContext\_NonOscThread, [106](#)
  - EPVThreadContext\_OscThread, [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](#)
  - OscPtrAdd, [106](#)
  - OscPtrSub, [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](#)
- PVEXECNAMELEN, [105](#)
- PVSCHEDNAMELEN, [105](#)
- QUE\_ITER\_BEGIN, [105](#)
- QUE\_ITER\_END, [105](#)
- TOscIReady, [106](#)
- TPVThreadContext, [106](#)
- OscIProcStatus, [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](#)
  - OUTOFMEMORY\_ERROR, [475](#)
  - PSHARED\_ATTRIBUTE\_SETTING\_-  
ERROR, [476](#)
  - PSHARED\_NOT\_ZERO\_ERROR, [476](#)
  - RELOCK\_MUTEX\_ERROR, [476](#)
  - SEM\_NOT\_SIGNALED\_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](#)
- OscIProcStatus
  - eOscIProcError, [475](#)
  - OscIPtr, [477](#)
    - OscIPtr, [477](#)
  - OscIPtr
    - Append, [477](#)
    - Length, [477](#)
    - OscIPtr, [477](#)
    - Ptr, [477](#)
    - Set, [477](#)
    - SetLength, [477](#)
    - Zero, [477](#)
  - OscIPtrAdd
    - osclproc, [106](#)
  - OscIPtrC, [479](#)
    - OscIPtrC, [480](#)
  - OscIPtrC
    - Left, [480](#)
    - Length, [480](#)
    - OscIPtrC, [480](#)
    - Ptr, [480](#)
    - Right, [480](#)
    - Set, [480](#)
    - SetLength, [480](#)
    - Zero, [480](#)
  - OscIPtrSub
    - osclproc, [106](#)
  - OscIRand, [481](#)
  - OscIRand
    - Rand, [481](#)
    - Seed, [481](#)
  - OscIReadFD
    - osclconfig\_io.h, [818](#)
  - OscIReadyAlloc, [482](#)
  - OscIReadyAlloc
    - allocate, [482](#)
    - allocate\_fl, [482](#)
    - deallocate, [482](#)
  - OscIReadyCompare, [483](#)
    - PVActiveBase, [615](#)
  - OscIReadyCompare
    - compare, [483](#)
  - OscIReadyQ, [484](#)
    - OscIExecSchedulerCommonBase, [397](#)
    - PVActiveBase, [615](#)
    - PVActiveStats, [616](#)
  - OscIReadyQ
    - 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](#)
- OslReadySetPosition
  - PVActiveBase, [615](#)
- OslRecv
  - oslconfig\_io.h, [818](#)
- OslRecvFrom
  - oslconfig\_io.h, [818](#)
- OslRecvFromMethod, [486](#)
- OslRecvFromMethod
  - ~OslRecvFromMethod, [486](#)
  - GetRecvData, [486](#)
  - NewL, [486](#)
  - RecvFrom, [486](#)
  - RecvFromRequest, [486](#)
- OslRecvFromRequest, [488](#)
  - OslRecvFromRequest, [488](#)
  - OslSocketI, [539](#)
- OslRecvFromRequest
  - GetRecvData, [488](#)
  - OslRecvFromRequest, [488](#)
  - RecvFrom, [488](#)
  - Success, [488](#)
- OslRecvMethod, [490](#)
- OslRecvMethod
  - ~OslRecvMethod, [490](#)
  - GetRecvData, [490](#)
  - NewL, [490](#)
  - Recv, [490](#)
  - RecvRequest, [490](#)
- OslRecvRequest, [491](#)
  - OslRecvRequest, [491](#)
  - OslSocketI, [539](#)
- OslRecvRequest
  - GetRecvData, [491](#)
  - OslRecvRequest, [491](#)
  - Recv, [491](#)
  - Success, [491](#)
- OslRefCount, [492](#)
- OslRefCount
  - ~OslRefCount, [492](#)
  - addRef, [492](#)
  - getCount, [492](#)
  - removeRef, [492](#)
- OslRefCountDA, [494](#)
  - OslRefCountDA, [494](#)
- OslRefCountDA
  - ~OslRefCountDA, [494](#)
  - addRef, [495](#)
  - getCount, [495](#)
  - OslRefCountDA, [494](#)
  - removeRef, [495](#)
- OslRefCountMemFrag, [496](#)
  - OslRefCountMemFrag, [496](#)
- OslRefCountMemFrag
  - ~OslRefCountMemFrag, [496](#)
  - getCapacity, [497](#)
  - getCount, [497](#)
  - getMemFrag, [497](#)
  - getMemFragPtr, [497](#)
  - getMemFragSize, [497](#)
  - getRefCount, [497](#)
  - operator=, [497](#)
  - OslRefCountMemFrag, [496](#)
- OslRefCountMTDA, [498](#)
  - OslRefCountMTDA, [498](#)
- OslRefCountMTDA
  - ~OslRefCountMTDA, [498](#)
  - addRef, [499](#)
  - getCount, [499](#)
  - OslRefCountMTDA, [498](#)
  - removeRef, [499](#)
- OslRefCountMTSA, [500](#)
  - OslRefCountMTSA, [500](#)
- OslRefCountMTSA
  - ~OslRefCountMTSA, [500](#)
  - addRef, [501](#)
  - getCount, [501](#)
  - OslRefCountMTSA, [500](#)
  - removeRef, [501](#)
- OslRefCountSA, [502](#)
  - OslRefCountSA, [502](#)
- OslRefCountSA
  - ~OslRefCountSA, [502](#)
  - addRef, [503](#)
  - getCount, [503](#)
  - OslRefCountSA, [502](#)
  - removeRef, [503](#)
- OslRegistryAccessClient, [504](#)
  - OslRegistryAccessClient, [504](#)
  - OslRegistryClientImpl, [512](#)
  - OslRegistryServTlsImpl, [515](#)
- OslRegistryAccessClient
  - ~OslRegistryAccessClient, [504](#)
  - Close, [504](#)
  - Connect, [504](#)
  - GetFactories, [504](#)
  - GetFactory, [504](#)
  - OslRegistryAccessClient, [504](#)
- OslRegistryAccessClientImpl, [506](#)
- OslRegistryAccessClientTlsImpl, [507](#)
- OslRegistryAccessElement, [508](#)
- OslRegistryAccessElement
  - iFactory, [508](#)
  - iMimeType, [508](#)

- OscRegistryClient, 509
  - OscRegistryClient, 509
  - OscRegistryClientImpl, 512
  - OscRegistryServTlsImpl, 515
- OscRegistryClient
  - ~OscRegistryClient, 509
  - Close, 509
  - Connect, 509
  - OscRegistryClient, 509
  - Register, 509
  - UnRegister, 510
- OscRegistryClientImpl, 511
- OscRegistryClientImpl
  - Close, 512
  - Connect, 512
  - GetFactories, 512
  - GetFactory, 512
  - OscRegistryAccessClient, 512
  - OscRegistryClient, 512
  - Register, 512
  - UnRegister, 512
- OscRegistryClientTlsImpl, 513
- OscRegistryServTlsImpl, 514
  - OscRegistryServTlsImpl, 515
- OscRegistryServTlsImpl
  - ~OscRegistryServTlsImpl, 515
  - Close, 515
  - Connect, 515
  - GetFactories, 515
  - GetFactory, 515
  - OscRegistryAccessClient, 515
  - OscRegistryClient, 515
  - OscRegistryServTlsImpl, 515
  - Register, 515
  - UnRegister, 515
- OscReturnCode
  - osclerror, 93
- OscScheduler, 516
  - OscErrorTrapImp, 378
  - OscExecScheduler, 390
  - OscExecSchedulerCommonBase, 397
- OscScheduler
  - Cleanup, 516
  - Init, 516
- OscSchedulerCommonBase
  - PVActiveBase, 615
- OscSchedulerObserver, 517
- OscSchedulerObserver
  - ~OscSchedulerObserver, 517
  - OscSchedulerReadyCallback, 517
  - OscSchedulerTimerCallback, 517
- OscSchedulerReadyCallback
  - OscSchedulerObserver, 517
- OscSchedulerTimerCallback
  - OscSchedulerObserver, 517
- OscSchedulerObserver, 517
- OscScopedLock, 518
  - OscScopedLock, 518
- OscScopedLock
  - ~OscScopedLock, 518
  - OscScopedLock, 518
- OscSelect, 519
  - OscSelect, 520
- OscSelect
  - iErrAlloc, 520
  - iHeapCheck, 520
  - iOscBase, 520
  - iOscErrorTrap, 520
  - iOscLogger, 520
  - iOscMemory, 520
  - iOscScheduler, 520
  - iOutputFile, 520
  - iSchedulerAlloc, 520
  - iSchedulerName, 520
  - iSchedulerReserve, 520
  - OscSelect, 520
- OscSemaphore, 521
  - OscSemaphore, 521
- OscSemaphore
  - ~OscSemaphore, 521
  - Close, 521
  - Create, 521
  - OscSemaphore, 521
  - Signal, 522
  - TryWait, 522
  - Wait, 522
- OscSend
  - osclconfig\_io.h, 819
- OscSendMethod, 523
- OscSendMethod
  - ~OscSendMethod, 523
  - GetSendData, 523
  - NewL, 523
  - Send, 523
  - SendRequest, 523
- OscSendRequest, 524
  - OscSendRequest, 524
  - OscSocketI, 539
- OscSendRequest
  - GetSendData, 524
  - OscSendRequest, 524
  - Send, 524
  - Success, 524
- OscSendTo
  - osclconfig\_io.h, 819
- OscSendToMethod, 525
- OscSendToMethod
  - ~OscSendToMethod, 525
  - GetSendData, 525

- NewL, [525](#)
- SendTo, [525](#)
- SendToRequest, [525](#)
- OscSendToRequest, [526](#)
- OscSendToRequest, [526](#)
- OscSocketI, [539](#)
- OscSendToRequest
  - GetSendData, [526](#)
  - OscSendToRequest, [526](#)
  - SendTo, [526](#)
  - Success, [526](#)
- OscSetNonBlocking
  - osclconfig\_io.h, [819](#)
- OscSetRecvBufferSize
  - osclconfig\_io.h, [819](#)
- OscSetSockOpt
  - osclconfig\_io.h, [819](#)
- OscSharedPtr, [527](#)
- OscSharedPtr, [528](#)
- OscSharedPtr
  - ~OscSharedPtr, [528](#)
  - get\_count, [528](#)
  - GetRefCounter, [528](#)
  - GetRep, [528](#)
  - operator \*, [528](#)
  - operator TheClass \*, [529](#)
  - operator->, [529](#)
  - operator=, [529](#)
  - OscSharedPtr, [528](#)
  - Unbind, [529](#)
- OscShutdown
  - osclconfig\_io.h, [819](#)
- OscShutdownMethod, [530](#)
- OscShutdownMethod
  - ~OscShutdownMethod, [530](#)
  - NewL, [530](#)
  - Shutdown, [530](#)
  - ShutdownRequest, [530](#)
- OscShutdownRequest, [531](#)
- OscShutdownRequest, [531](#)
- OscSocketI, [539](#)
- OscShutdownRequest
  - OscShutdownRequest, [531](#)
  - Shutdown, [531](#)
- OscSingleton, [532](#)
- OscSingleton, [532](#)
- OscSingleton
  - ~OscSingleton, [532](#)
  - \_Ptr, [533](#)
  - operator \*, [532](#)
  - operator->, [532](#)
  - OscSingleton, [532](#)
  - set, [532](#)
- OscSingletonRegistry, [534](#)
- OscSingletonRegistry
  - getInstance, [534](#)
  - lockAndGetInstance, [534](#)
  - OscBase, [534](#)
  - registerInstance, [534](#)
  - registerInstanceAndUnlock, [534](#)
- OscSocket
  - osclconfig\_io.h, [820](#)
- OscSocketCleanup
  - osclconfig\_io.h, [820](#)
- OscSocketI, [535](#)
- OscSocketRequestAO, [553](#)
- OscSocketServI, [557](#)
- OscSocketI
  - ~OscSocketI, [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](#)
  - OscAcceptRequest, [539](#)
  - OscConnectRequest, [539](#)
  - OscRecvFromRequest, [539](#)
  - OscRecvRequest, [539](#)
  - OscSendRequest, [539](#)
  - OscSendToRequest, [539](#)
  - OscShutdownRequest, [539](#)
  - OscTCPSocket, [539](#)
  - OscUDPSocket, [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](#)
- OscSocketIBase, [540](#)
  - OscSocketIBase, [541](#)
- OscSocketIBase
  - ~OscSocketIBase, [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](#)
  - OscSocketIBase, [541](#)
  - OscSocketMethod, [544](#)
  - OscSocketRequest, [544](#)
  - OscSocketRequestAO, [544](#)
  - OscTCPSocket, [544](#)
  - OscUDPSocket, [544](#)
  - Recv, [543](#)
  - RecvFrom, [543](#)
  - RecvFromSuccess, [543](#)
  - RecvSuccess, [543](#)
  - Send, [543](#)
  - SendSuccess, [543](#)
  - SendTo, [543](#)
  - SendToSuccess, [543](#)
  - Shutdown, [544](#)
- OscSocketMethod, [545](#)
  - OscIPSocketI, [420](#)
  - OscSocketIBase, [544](#)
  - OscSocketMethod, [546](#)
  - OscSocketRequestAO, [553](#)
- OscSocketMethod
  - ~OscSocketMethod, [546](#)
- Abort, [546](#)
- AbortAll, [546](#)
- Alloc, [546](#)
- CancelMethod, [546](#)
- ConstructL, [546](#)
- iContainer, [547](#)
- iSocketFxn, [547](#)
- iSocketRequestAO, [547](#)
- MethodDone, [546](#)
- OscSocketMethod, [546](#)
- Run, [546](#)
- StartMethod, [547](#)
- ThreadLogoff, [547](#)
- ThreadLogon, [547](#)
- OscSocketObserver, [548](#)
- OscSocketObserver
  - ~OscSocketObserver, [548](#)
  - HandleSocketEvent, [548](#)
- OscSocketRequest, [549](#)
  - OscSocketIBase, [544](#)
  - OscSocketRequest, [549](#)
  - OscSocketRequestAO, [553](#)
  - OscSocketServI, [557](#)
- OscSocketRequest
  - Activate, [549](#)
  - CancelRequest, [549](#)
  - Complete, [549](#)
  - Fxn, [549](#)
  - iParam, [549](#)
  - iSocketI, [549](#)
  - iSocketRequestAO, [549](#)
  - OscSocketRequest, [549](#)
- OscSocketRequestAO, [550](#)
  - OscIPSocketI, [420](#)
  - OscSocketIBase, [544](#)
  - OscSocketRequestAO, [551](#)
- OscSocketRequestAO
  - ~OscSocketRequestAO, [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](#)
  - OscSocketI, [553](#)
  - OscSocketMethod, [553](#)
  - OscSocketRequest, [553](#)
  - OscSocketRequestAO, [551](#)

- RequestDone, [552](#)
- Run, [552](#)
- SocketI, [552](#)
- SocketObserver, [552](#)
- Success, [552](#)
- OscSocketSelect
  - osclconfig\_io.h, [820](#)
- OscSocketServ, [554](#)
  - OscSocketServI, [557](#)
- OscSocketServ
  - ~OscSocketServ, [554](#)
  - Close, [554](#)
  - Connect, [554](#)
  - NewL, [555](#)
  - OscIDNS, [555](#)
  - OscTCPSocket, [555](#)
  - OscUDPSocket, [555](#)
- OscSocketServI, [556](#)
  - OscSocketServRequestList, [560](#)
- OscSocketServI
  - Close, [556](#)
  - Connect, [556](#)
  - IsServerThread, [557](#)
  - LoopbackSocket, [557](#)
  - NewL, [557](#)
  - OscIDNSI, [557](#)
  - OscSocketI, [557](#)
  - OscSocketRequest, [557](#)
  - OscSocketServ, [557](#)
  - OscSocketServRequestList, [557](#)
  - OscTCPSocketI, [557](#)
  - OscUDPSocketI, [557](#)
- OscSocketServIBase, [558](#)
  - ESocketServ\_Connected, [558](#)
  - ESocketServ\_Error, [559](#)
  - ESocketServ\_Idle, [558](#)
  - OscSocketServIBase, [559](#)
- OscSocketServIBase
  - ~OscSocketServIBase, [559](#)
  - Close, [559](#)
  - Connect, [559](#)
  - iAlloc, [559](#)
  - iLogger, [559](#)
  - iServError, [559](#)
  - iServState, [559](#)
  - IsServConnected, [559](#)
  - OscSocketServIBase, [559](#)
  - State, [559](#)
  - TSocketServState, [558](#)
- OscSocketServRequestList, [560](#)
  - OscSocketServI, [557](#)
  - OscSocketServRequestList, [560](#)
- OscSocketServRequestList
  - Add, [560](#)
  - Close, [560](#)
  - Open, [560](#)
  - OscSocketServI, [560](#)
  - OscSocketServRequestList, [560](#)
  - Remove, [560](#)
  - StartCancel, [560](#)
  - WaitOnRequests, [560](#)
  - Wakeup, [560](#)
- OscSocketServRequestQElem, [562](#)
  - OscSocketServRequestQElem, [562](#)
- OscSocketServRequestQElem
  - iCancel, [562](#)
  - iSelect, [562](#)
  - iSocketRequest, [562](#)
  - OscSocketServRequestQElem, [562](#)
- OscSocketStartup
  - osclconfig\_io.h, [820](#)
- OscSocketTOS, [563](#)
  - EPVCritic\_Ecp, [563](#)
  - EPVFlash, [563](#)
  - EPVHiRel, [563](#)
  - EPVHiThrtpt, [563](#)
  - EPVImmediate, [563](#)
  - EPVInetControl, [563](#)
  - EPVLDelay, [563](#)
  - EPVNetControl, [563](#)
  - EPVNoTOS, [563](#)
  - EPVOverrideFlash, [563](#)
  - EPVPriority, [563](#)
  - EPVRoutine, [563](#)
  - OscSocketTOS, [564](#)
- OscSocketTOS
  - ClearTOS, [564](#)
  - GetTOS, [564](#)
  - OscSocketTOS, [564](#)
  - SetPrecedence, [564](#)
  - SetPriority, [564](#)
  - TPVServicePrecedence, [563](#)
  - TPVServicePriority, [563](#)
- OscSuccess
  - osclerror, [92](#)
- OscTagTreeType
  - osclmemory, [59](#)
- OscTCPSocket, [565](#)
  - OscSocketI, [539](#)
  - OscSocketIBase, [544](#)
  - OscSocketServ, [555](#)
- OscTCPSocket
  - ~OscTCPSocket, [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](#)
  - OscIThread, [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](#)
  - OscIThreadLock, [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](#)
  - OscIExecSchedulerCommonBase, [397](#)
- OscITimerCompare
  - compare, [585](#)
- OscITimerObject, [586](#)
  - OscIExecSchedulerCommonBase, [399](#)
  - OscITimerObject, [587](#)
  - PVActiveBase, [615](#)
  - PVActiveStats, [616](#)
  - PVThreadContext, [635](#)
- OscITimerObject



- ~OscTimerObject, [587](#)
- AddToScheduler, [587](#)
- After, [587](#)
- Cancel, [587](#)
- DoCancel, [587](#)
- IsBusy, [588](#)
- OscTimerObject, [587](#)
- Priority, [588](#)
- RemoveFromScheduler, [588](#)
- RunError, [588](#)
- RunIfNotReady, [588](#)
- SetBusy, [588](#)
- SetStatus, [588](#)
- Status, [589](#)
- StatusRef, [589](#)
- OscTimerObserver, [590](#)
- OscTimerObserver
  - ~OscTimerObserver, [590](#)
  - TimeoutOccurred, [590](#)
- OscTimerQ, [591](#)
- OscTimerQ
  - Add, [591](#)
  - Construct, [591](#)
  - IsIn, [591](#)
  - Pop, [591](#)
  - PopTop, [591](#)
  - Remove, [591](#)
  - Top, [591](#)
- OscTLS, [592](#)
  - OscTLS, [592](#)
- OscTLS
  - ~OscTLS, [592](#)
  - \_Ptr, [593](#)
  - operator \*, [592](#)
  - operator->, [592](#)
  - OscTLS, [592](#)
  - set, [592](#)
- OscTLSEx, [594](#)
  - OscTLSEx, [594](#)
- OscTLSEx
  - ~OscTLSEx, [594](#)
  - \_Ptr, [595](#)
  - operator \*, [594](#)
  - operator->, [594](#)
  - OscTLSEx, [594](#)
  - set, [594](#)
- OscTLSRegistry, [596](#)
- OscTLSRegistry
  - getInstance, [596](#)
  - OscBase, [596](#)
  - registerInstance, [596](#)
- OscTLSRegistryEx, [597](#)
- OscTLSRegistryEx
  - getInstance, [597](#)
  - registerInstance, [597](#)
- OscTrapItem, [598](#)
  - OscTrapItem, [598](#)
- OscTrapItem
  - OscTrapItem, [598](#)
  - OscTrapStack, [598](#)
  - OscTrapStackItem, [598](#)
- OscTrapOperation
  - osclerror, [93](#)
- OscTrapStack, [599](#)
  - OscErrorTrapImp, [378](#)
  - OscTrapItem, [598](#)
- OscTrapStack
  - OscError, [599](#)
  - OscErrorTrap, [599](#)
  - OscErrorTrapImp, [599](#)
- OscTrapStackItem, [600](#)
  - OscTrapItem, [598](#)
  - OscTrapStackItem, [600](#)
- OscTrapStackItem
  - iCBase, [600](#)
  - iNext, [600](#)
  - iTAny, [600](#)
  - iTrapOperation, [600](#)
  - OscTrapStackItem, [600](#)
- OscUDPSocket, [601](#)
  - OscSocketI, [539](#)
  - OscSocketIBase, [544](#)
  - OscSocketServ, [555](#)
- OscUDPSocket
  - ~OscUDPSocket, [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](#)
- OscUDPSocketI, [607](#)
  - OscSocketServI, [557](#)
- OscUDPSocketI

- ~OscIUDPSocketI, 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
- OscIUid32
  - oscl\_uuid.h, 799
- OscIUnMakeInAddr
  - osclconfig\_io.h, 820
- OscIUnMakeSockAddr
  - 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\_UnicodeToUTF8, 79
  - oscl\_UTF8ToUnicode, 80
  - oscl\_vsnprintf, 80, 82
  - OSCL\_wHeapString, 82
  - OSCL\_wStackString, 82
  - OscIComponentFactory, 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
- OscIUid, 610
  - OscIUid, 611
- OscIUid
  - data1, 611
  - data2, 611
  - data3, 611
  - data4, 611
  - operator!=, 611
  - operator=, 611
  - operator==, 611
  - OscIUid, 611
- OscIValidInetAddr
  - osclconfig\_io.h, 821
- OscIWriteFD
  - osclconfig\_io.h, 821
- other
  - OscI\_TAlloc::rebind, 284
- other\_chatype
  - 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
  - OscProcStatus, 475
- OUTOFMEMORY\_ERROR
  - OscProcStatus, 475
- overwrite
  - CFastRep, 128
- pad
  - MM\_AllocBlockFence, 147
  - MM\_AllocBlockHdr, 148
- pair\_citerator\_citerator
  - Osc\_Map, 218
- pair\_iterator\_bool
  - Osc\_Map, 218
  - Osc\_TagTree, 270
- pair\_iterator\_iterator
  - Osc\_Map, 218
- pAllocInfo
  - MM\_AllocNode, 151
- parent
  - Osc\_Rb\_Tree\_Node\_Base, 255
  - Osc\_TagTree::Node, 280
- pAudit
  - OscAuditCB, 321
- pBasePosition
  - OscBinStream, 339
- pBuffer
  - OscFileCacheBuffer, 404
- peakNumAllocs
  - MM\_Stats\_t, 165
- peakNumBytes
  - MM\_Stats\_t, 165
- PendComplete
  - OscActiveObject, 312
  - OscExecSchedulerCommonBase, 396
  - OscReadyQ, 485
- PendForExec
  - OscActiveObject, 312
- per\_allocation\_overhead
  - MM\_AuditOverheadStats, 161
- perms
  - oscl\_stat\_buf, 259
- PersistHostAddress
  - GetHostByNameParam, 136
- pFileName
  - MM\_AllocInfo, 150
- pMemBlock
  - MM\_AllocInfo, 150
  - MM\_AllocQueryInfo, 152
- pMMFIParam
  - OscMemStatsNode, 458
- pMMStats
  - OscMemStatsNode, 458
- pNext
  - MM\_AllocNode, 151
- pNode
  - MM\_AllocBlockHdr, 148
- pointer
  - MemAllocator, 146
  - Osc\_Map, 218
  - Osc\_Queue, 237
  - Osc\_Rb\_Tree, 244
  - Osc\_Rb\_Tree\_Const\_Iterator, 248
  - Osc\_Rb\_Tree\_Iterator, 251
  - Osc\_TagTree::const\_iterator, 274
  - Osc\_TagTree::iterator, 277
  - Osc\_TAlloc, 282
  - Osc\_Vector, 286
- Pop
  - OscError, 372
  - OscTimerQ, 591
- pop
  - Osc\_Queue, 238
  - Osc\_Queue\_Base, 240
  - OscPriorityQueue, 472
- pop\_back
  - Osc\_Vector, 288
  - Osc\_Vector\_Base, 292
- pop\_heap
  - OscPriorityQueue, 472
  - OscPriorityQueueBase, 474
- PopDealloc
  - OscError, 372, 373
- PopTop
  - OscReadyQ, 485
  - OscTimerQ, 591
- port
  - OscNetworkAddress, 466
- PositionInBlock
  - OscBinStream, 338
- pPosition
  - OscBinStream, 339
- pPrev
  - MM\_AllocNode, 151
- Preceeds
  - OscFileCacheBuffer, 404
- PrepRead
  - OscFileCacheBuffer, 404
- PrepWrite
  - OscFileCacheBuffer, 404
- Priority
  - OscActiveObject, 312
  - OscTimerObject, 588
- ProcessAccept
  - OscSocketI, 537

ProcessConnect  
     OscSocketI, [538](#)  
 ProcessRecv  
     OscSocketI, [538](#)  
 ProcessRecvFrom  
     OscSocketI, [538](#)  
 ProcessSend  
     OscSocketI, [538](#)  
 ProcessSendTo  
     OscSocketI, [538](#)  
 ProcessShutdown  
     OscSocketI, [538](#)  
 pRootNode  
     MM\_AllocBlockHdr, [148](#)  
 pruneSubtree  
     MM\_Audit\_Imp, [159](#)  
 PSHARED\_ATTRIBUTE\_SETTING\_ERROR  
     OscProcStatus, [476](#)  
 PSHARED\_NOT\_ZERO\_ERROR  
     OscProcStatus, [476](#)  
 pStats  
     MM\_Stats\_CB, [163](#)  
 pStatsNode  
     MM\_AllocInfo, [150](#)  
     OscAuditCB, [321](#)  
 Ptr  
     OscIPtr, [477](#)  
     OscIPtrC, [480](#)  
 ptr  
     OscMemoryFragment, [442](#)  
     StrPtrLen, [648](#)  
     WStrPtrLen, [659](#)  
 push  
     Osc\_Queue, [238](#)  
     Osc\_Queue\_Base, [240](#)  
     OscPriorityQueue, [472](#)  
 push\_back  
     Osc\_Vector, [289](#)  
     Osc\_Vector\_Base, [292](#)  
 push\_front  
     Osc\_Vector, [289](#)  
     Osc\_Vector\_Base, [293](#)  
 push\_heap  
     OscPriorityQueue, [472](#)  
     OscPriorityQueueBase, [474](#)  
 PushL  
     OscError, [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
  - iPVActiveStats, 615
  - iPVReadyQLink, 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
- PVERROR\_IMP\_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
  - GetNumAppenders, 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\_HLDBG, [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\_HLDBG, [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\_HLDBG, [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\_HLDBG, [854](#)  
 PVLOGMSG\_INST\_LLDBG, [854](#)  
 PVLOGMSG\_INST\_MLDBG, [854](#)  
 PVLOGMSG\_INST\_PROF, [855](#)  
 PVLOGMSG\_INST\_REL, [855](#)  
 PVLOGMSG\_NONFATAL\_ERROR, [856](#)  
 PVLOGMSG\_NOTICE, [856](#)  
 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\_HLDBG, [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](#)
- PVNETWORKADDRESS\_LEN
  - oscl\_socket\_types.h, [774](#)
- PVOsclBase\_Cleanup
  - osclbase, [45](#)
- PVOsclBase\_Init
  - osclbase, [45](#)
- PVSCHEDNAMELEN
  - osclproc, [105](#)
- PVSchedulerStopper, [631](#)
  - OscExecSchedulerCommonBase, [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](#)
  - OscExecSchedulerCommonBase, [399](#)
  - PVThreadContext, [634](#)
- PVThreadContext
  - ~PVThreadContext, [634](#)
  - EnterThreadContext, [634](#)



- ExitThreadContext, [634](#)
- Id, [634](#)
- IsSameThreadContext, [634](#)
- OscActiveObject, [635](#)
- OscCoeActiveScheduler, [635](#)
- OscCoeActiveSchedulerBase, [635](#)
- OscExecScheduler, [635](#)
- OscExecSchedulerBase, [635](#)
- OscExecSchedulerCommonBase, [635](#)
- OscTimerObject, [635](#)
- PVActiveBase, [635](#)
- PVThreadContext, [634](#)
- ThreadHasScheduler, [635](#)
  
- QUE\_ITER\_BEGIN
  - osclproc, [105](#)
- QUE\_ITER\_END
  - osclproc, [105](#)
  
- Rand
  - OscRand, [481](#)
- Read
  - Osc\_File, [183](#)
  - OscAsyncFile, [318](#)
  - OscBinIStreamBigEndian, [327](#)
  - OscFileCache, [402](#)
  - OscNativeFile, [463](#)
- read
  - OSCL\_String, [263](#)
  - OSCL\_wString, [306](#)
- Read\_uint16
  - OscBinIStreamBigEndian, [327](#)
  - OscBinIStreamLittleEndian, [330](#)
- Read\_uint32
  - OscBinIStreamBigEndian, [327](#)
  - OscBinIStreamLittleEndian, [330](#)
- Read\_uint8
  - OscBinIStream, [324](#)
- ReadAsync
  - OscNativeFile, [463](#)
- ReadAsyncCancel
  - OscNativeFile, [463](#)
- rebalance
  - Osc\_Rb\_Tree\_Base, [246](#)
- rebalance\_for\_erase
  - Osc\_Rb\_Tree\_Base, [246](#)
- Recv
  - OscRecvMethod, [490](#)
  - OscRecvRequest, [491](#)
  - OscSocketI, [538](#)
  - OscSocketIBase, [543](#)
  - OscTCPSocket, [570](#)
  - OscTCPSocketI, [574](#)
- RecvFrom
  - OscRecvFromMethod, [486](#)
  - OscRecvFromRequest, [488](#)
  - OscSocketI, [538](#)
  - OscSocketIBase, [543](#)
  - OscUDPSocket, [604](#)
  - OscUDPSocketI, [608](#)
- RecvFromParam, [636](#)
  - RecvFromParam, [636](#)
- RecvFromParam
  - iAddr, [636](#)
  - iBufRecv, [636](#)
  - iFlags, [636](#)
  - iMultiMaxLen, [636](#)
  - iPacketLen, [636](#)
  - iPacketSource, [636](#)
  - RecvFromParam, [636](#)
- RecvFromRequest
  - OscRecvFromMethod, [486](#)
- RecvFromSuccess
  - OscSocketI, [538](#)
  - OscSocketIBase, [543](#)
- RecvParam, [638](#)
  - RecvParam, [638](#)
- RecvParam
  - iBufRecv, [638](#)
  - iFlags, [638](#)
  - RecvParam, [638](#)
- RecvRequest
  - OscRecvMethod, [490](#)
- RecvSuccess
  - OscSocketI, [538](#)
  - OscSocketIBase, [543](#)
- red
  - Osc\_Rb\_Tree\_Node\_Base, [254](#)
- RedBl
  - Osc\_Rb\_Tree\_Node\_Base, [254](#)
- refcount
  - CHepRep, [130](#)
- reference
  - Osc\_Map, [218](#)
  - Osc\_Queue, [237](#)
  - Osc\_Rb\_Tree, [244](#)
  - Osc\_Rb\_Tree\_Const\_Iterator, [248](#)
  - Osc\_Rb\_Tree\_Iterator, [251](#)
  - Osc\_TagTree::const\_iterator, [274](#)
  - Osc\_TagTree::iterator, [277](#)
  - Osc\_TAlloc, [282](#)
  - Osc\_Vector, [286](#)
- Register
  - OscComponentRegistry, [344](#)
  - OscRegistryClient, [509](#)
  - OscRegistryClientImpl, [512](#)
  - OscRegistryServTlsImpl, [515](#)
- RegisterForCallback

- OscExecScheduler, 389
- OscReadyQ, 485
- registerInstance
  - OscSingletonRegistry, 534
  - OscTLSRegistry, 596
  - OscTLSRegistryEx, 597
- registerInstanceAndUnlock
  - OscSingletonRegistry, 534
- release
  - OscExclusiveArrayPtr, 382
  - OscExclusivePtr, 385
  - OscExclusivePtrA, 388
  - OSCLMemAutoPtr, 437
- RELOCK\_MUTEX\_ERROR
  - OscProcStatus, 476
- Remove
  - OscDoubleLink, 366
  - OscReadyQ, 485
  - OscSocketServRequestList, 560
  - OscTimerQ, 591
- remove
  - OscPriorityQueue, 472
  - OscPriorityQueueBase, 474
- remove\_element
  - Osc\_Linked\_List, 209
  - Osc\_Linked\_List\_Base, 214
  - Osc\_MTLlinked\_List, 227
- remove\_ref
  - CHepRep, 130
- removeALLAllocNodes
  - MM\_Audit\_Imp, 159
- removeAllocNode
  - MM\_Audit\_Imp, 159
- RemoveAppender
  - PVLogger, 621
- RemoveFixedCache
  - Osc\_File, 183
- RemoveFromScheduler
  - OscActiveObject, 312
  - OscTimerObject, 588
  - PVActiveBase, 614
- RemoveRef
  - DNSRequestParam, 134
- removeRef
  - Osc\_DefAllocWithRefCount, 174
  - OscMemPoolFixedChunkAllocator, 445
  - OscMemPoolResizableAllocator, 452
  - OscRefCount, 492
  - OscRefCountDA, 495
  - OscRefCountMTDA, 499
  - OscRefCountMTSA, 501
  - OscRefCountSA, 503
- Request
  - OscTimer, 583
- RequestCanceled
  - OscExecSchedulerCommonBase, 396
- RequestDone
  - OscDNSRequestAO, 364
  - OscSocketRequestAO, 552
- reserve
  - Osc\_Queue\_Base, 240
  - Osc\_Vector\_Base, 293
  - OscPriorityQueue, 472
- ReserveSpace
  - OscBinStream, 338
- Reset
  - OscDoubleListBase, 369
- reset
  - BufferState, 119
  - MM\_FailInsertParam, 162
  - MM\_Stats\_t, 165
  - OscMemStatsNode, 458
- ResetLogPerf
  - OscExecSchedulerCommonBase, 396
- Resume
  - OscThread, 577
- ResumeScheduler
  - OscExecSchedulerCommonBase, 396
- retrieveParentTag
  - MM\_Audit\_Imp, 159
- retrieveParentTagLength
  - MM\_Audit\_Imp, 159
- RFC822ToPV8601
  - osclbase, 45
- Right
  - OscPtrC, 480
- right
  - Osc\_Rb\_Tree\_Node\_Base, 255
- rotate\_left
  - Osc\_Rb\_Tree\_Base, 246
- rotate\_right
  - Osc\_Rb\_Tree\_Base, 246
- Run
  - CallbackTimer, 124
  - OscDNSMethod, 359
  - OscDNSRequestAO, 364
  - OscSocketMethod, 546
  - OscSocketRequestAO, 552
  - PVActiveBase, 614
- RunError
  - OscActiveObject, 312
  - OscTimerObject, 588
  - PVActiveBase, 614
- RunIfNotReady
  - OscActiveObject, 313
  - OscTimerObject, 588
- RunSchedulerNonBlocking
  - OscExecScheduler, 389

- save\_registry
  - TLSSStorageOps, 656
- second
  - OscI\_Pair, 235
- SECONDS
  - osclbase, 35
- Seed
  - OscIRand, 481
- Seek
  - OscI\_File, 183
  - OscIAsyncFile, 318
  - OscIBinStream, 338
  - OscIFileCache, 402
  - OscINativeFile, 464
- seek\_type
  - OscI\_File, 180
- SEEKCUR
  - OscI\_File, 180
- SEEKEND
  - OscI\_File, 180
- seekFromCurrentPosition
  - OscIBinStream, 338
- SEEKSET
  - OscI\_File, 180
- self
  - OscI\_Map, 218
  - OscI\_Rb\_Tree\_Const\_Iterator, 248
  - OscI\_Rb\_Tree\_Iterator, 251
  - OscI\_TagTree::const\_iterator, 274
  - OscI\_TagTree::iterator, 277
- SEM\_NOT\_SIGNED\_ERROR
  - OscIProcStatus, 476
- Send
  - OscISendMethod, 523
  - OscISendRequest, 524
  - OscISocketI, 538
  - OscISocketIBase, 543
  - OscITCPSocket, 570
  - OscITCPSocketI, 574
- SendParam, 639
  - SendParam, 639
- SendParam
  - iBufSend, 639
  - iFlags, 639
  - iXferLen, 639
  - SendParam, 639
- SendRequest
  - OscISendMethod, 523
- SendSuccess
  - OscISocketI, 538
  - OscISocketIBase, 543
- SendTo
  - OscISendToMethod, 525
  - OscISendToRequest, 526
  - OscISocketI, 538
  - OscISocketIBase, 543
  - OscIUDPSocket, 605
  - OscIUDPSocketI, 608
- SendToParam, 640
  - SendToParam, 640
- SendToParam
  - ~SendToParam, 640
  - iAddr, 640
  - iBufSend, 640
  - iFlags, 640
  - iXferLen, 640
  - SendToParam, 640
- SendToRequest
  - OscISendToMethod, 525
- SendToSuccess
  - OscISocketI, 538
  - OscISocketIBase, 543
- Serv
  - OscIDNSRequestAO, 365
- Set
  - OscIDoubleRunner, 370
  - OscINameString, 461
  - OscIPtr, 477
  - OscIPtrC, 480
- set
  - CHeapRep, 130
  - CStackRep, 132
  - OSCL\_FastString, 177, 178
  - OSCL\_HeapStringA, 201, 202
  - OSCL\_wFastString, 296
  - OSCL\_wHeapStringA, 301
  - OscIExclusiveArrayPtr, 382
  - OscIExclusivePtr, 385
  - OscIExclusivePtrA, 388
  - OscISingleton, 532
  - OscITLS, 592
  - OscITLSEx, 594
  - osclutil, 82–84
- set\_from\_ntp\_time
  - TimeValue, 654
- set\_from\_system\_time
  - NTPTIME, 169
- set\_int64
  - OscI\_Int64\_Utills, 204
- set\_len
  - OSCL\_String, 263
  - OSCL\_wString, 307
- set\_length
  - OSCL\_FastString, 178
  - OSCL\_wFastString, 296
- set\_next
  - OscI\_Opaque\_Type\_Alloc\_LL, 232
- set\_r

- CFastRep, [128](#)
- set\_rep
  - CHearRep, [130](#)
  - OSCL\_String, [263](#), [264](#)
  - OSCL\_wString, [307](#)
- set\_to\_current\_time
  - NTPTIME, [169](#)
  - TimeValue, [654](#)
- set\_to\_zero
  - TimeValue, [655](#)
- set\_uint64
  - OscInt64\_Utils, [204](#)
- set\_w
  - CFastRep, [128](#)
- set\_zulu
  - TimeValue, [655](#)
- setAllocNodeFlag
  - MM\_AllocBlockHdr, [148](#)
- SetAsyncReadBufferSize
  - OscFile, [183](#)
- SetBusy
  - OscActiveObject, [313](#)
  - OscTimerObject, [588](#)
- SetCacheObserver
  - OscFile, [184](#)
- setChecksum
  - StrCSumPtrLen, [645](#)
- SetExactFrequency
  - OscTimer, [583](#)
- SetFileHandle
  - OscFile, [184](#)
- SetFrequency
  - OscTimer, [584](#)
- SetInUse
  - OscAsyncFileBuffer, [320](#)
- SetLength
  - OscIPtr, [477](#)
  - OscIPtrC, [480](#)
- SetLoggingEnable
  - OscFile, [184](#)
- SetLogLevel
  - PVLogger, [621](#)
- SetLogLevelAndPropagate
  - PVLogger, [622](#)
- setMaxSzForNewMemPoolBuffer
  - OscMemPoolResizableAllocator, [452](#)
- SetMulticastTTL
  - OscUDPSocket, [605](#)
  - OscUDPSocketI, [608](#)
- SetNativeAccessMode
  - OscFile, [184](#)
- SetNativeBufferSize
  - OscFile, [185](#)
- SetNodeLogLevelExplicit
  - PVLoggerRegistry, [629](#)
- SetObserver
  - OscTimer, [584](#)
- SetOffset
  - OscAsyncFileBuffer, [320](#)
  - OscDoubleListBase, [369](#)
- SetOptionToReuseAddress
  - OscIPSocketI, [419](#)
  - OscTCPSocket, [570](#)
  - OscUDPSocket, [605](#)
- SetParent
  - PVLogger, [622](#)
- SetPosition
  - OscFileCacheBuffer, [404](#)
- SetPrecedence
  - OscSocketTOS, [564](#)
- SetPriority
  - OscSocketTOS, [564](#)
  - OscThread, [577](#)
- setPtrLen
  - StrCSumPtrLen, [645](#)
  - StrPtrLen, [648](#)
  - WStrPtrLen, [659](#)
- SetPVCacheSize
  - OscFile, [185](#)
- SetRecvBufferSize
  - OscIPSocketI, [419](#)
  - OscSocketI, [538](#)
  - OscUDPSocket, [606](#)
- setrep\_to\_char
  - OSCL\_String, [264](#)
- setrep\_to\_wide\_char
  - OSCL\_wString, [307](#)
- SetScheduler
  - OscExecSchedulerCommonBase, [396](#)
- SetSize
  - OscFile, [185](#)
  - OscNativeFile, [464](#)
- SetSockOpt
  - OscSocketI, [539](#)
- SetStatus
  - OscActiveObject, [313](#)
  - OscTimerObject, [588](#)
- SetSummaryStatsLoggingEnable
  - OscFile, [185](#)
- SetTimestamp
  - MediaData, [144](#)
- SetToHead
  - OscDoubleRunner, [370](#)
- SetTOS
  - OscIPSocketI, [419](#)
  - OscTCPSocket, [571](#)
  - OscUDPSocket, [606](#)
- SetToTail

- OscIDoubleRunner, 370
- setWithoutOwnership
  - OSCLMemAutoPtr, 437
- ShowStats
  - OscIExecSchedulerCommonBase, 396
- ShowSummaryStats
  - OscIExecSchedulerCommonBase, 396
- Shutdown
  - OscIShutdownMethod, 530
  - OscIShutdownRequest, 531
  - OscISocketI, 539
  - OscISocketIBase, 544
  - OscITCPSocket, 571
  - OscITCPSocketI, 574
- ShutdownParam, 641
  - ShutdownParam, 641
- ShutdownParam
  - iHow, 641
  - ShutdownParam, 641
- ShutdownRequest
  - OscIShutdownMethod, 530
- Signal
  - OscISemaphore, 522
- Size
  - OscI\_File, 185
  - OscIAsyncFile, 318
  - OscINativeFile, 464
- size
  - CFastRep, 128
  - CHepRep, 130
  - CStackRep, 132
  - MM\_AllocBlockHdr, 148
  - MM\_AllocInfo, 150
  - MM\_AllocQueryInfo, 152
  - OscI\_Map, 221
  - OscI\_Queue\_Base, 240
  - OscI\_Rb\_Tree, 244
  - OscI\_TagTree, 272
  - OscI\_Vector\_Base, 293
  - OscIPriorityQueue, 472
  - StrPtrLen, 648
  - WStrPtrLen, 659
- size\_type
  - OscI\_Map, 218
  - OscI\_Queue, 237
  - OscI\_Rb\_Tree, 244
  - OscI\_Tag\_Base, 268
  - OscI\_TagTree, 270
  - OscI\_TAlloc, 282
- sizeof\_T
  - OscI\_Linked\_List\_Base, 215
  - OscI\_Queue\_Base, 241
  - OscI\_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
  - OscIThread, 577
- Socket
  - OscISocketI, 539
- SocketI
  - OscISocketRequestAO, 552
- SocketObserver
  - OscISocketRequestAO, 552
- SocketRequestParam, 642
  - SocketRequestParam, 643
- SocketRequestParam
  - iFxn, 643
  - SocketRequestParam, 643
- SocketServ
  - OscIIPSocketI, 419
- sort\_children
  - OscI\_TagTree::Node, 280
- specialFragBuffer
  - OscIBinStream, 339
- Start
  - OscIFileStats, 411
- Start\_on\_creation
  - oscl\_thread.h, 788
- StartAsyncRead
  - OscIAsyncFileBuffer, 320
- StartCancel
  - OscISocketServRequestList, 560
- StartMethod
  - OscIDNSMethod, 359
  - OscISocketMethod, 547
- StartNativeScheduler
  - OscIExecSchedulerCommonBase, 396
- StartScheduler
  - OscIExecSchedulerCommonBase, 396
- State
  - OscISocketServIBase, 559
- state
  - OscIBinStream, 339
- state\_t
  - OscIBinStream, 337
- StaticJump
  - OscIJump, 421
- stats\_overhead
  - MM\_AuditOverheadStats, 161
- Status

- OslActiveObject, [313](#)
- OslTimerObject, [589](#)
- status\_t
  - BufFragStatusClass, [123](#)
- StatusRef
  - OslActiveObject, [313](#)
  - OslTimerObject, [589](#)
- StopScheduler
  - OslExecSchedulerCommonBase, [396](#)
- Str
  - OslNameString, [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
  - OslDNSRequestAO, [365](#)
  - OslRecvFromRequest, [488](#)
  - OslRecvRequest, [491](#)
  - OslSendRequest, [524](#)
  - OslSendToRequest, [526](#)
  - OslSocketRequestAO, [552](#)
- SUCCESS\_ERROR
  - OslProcStatus, [475](#)
- Suspend
  - OslThread, [578](#)
- Suspend\_on\_creation
  - oscl\_thread.h, [788](#)
- SuspendScheduler
  - OslExecSchedulerCommonBase, [397](#)
- swap
  - Osl\_Opaque\_Type\_Compare, [233](#)
  - OslPriorityQueue, [472](#)
- SYSTEM\_RESOURCES\_UNAVAILABLE\_
  - ERROR
  - OslProcStatus, [476](#)
- tag
  - MM\_AllocQueryInfo, [152](#)
  - MM\_Stats\_CB, [163](#)
  - Osl\_Tag, [265](#)
  - Osl\_TagTree::Node, [280](#)
  - OslMemStatsNode, [458](#)
- tag\_ancestor
  - Osl\_Tag\_Base, [268](#)
- tag\_base\_type
  - Osl\_Tag\_Base, [268](#)
  - Osl\_TagTree, [270](#)
- tag\_base\_unit
  - Osl\_Tag\_Base, [268](#)
- tag\_cmp
  - Osl\_Tag\_Base, [268](#)
- tag\_copy
  - Osl\_Tag\_Base, [268](#)
- tag\_depth
  - Osl\_Tag\_Base, [268](#)
- tag\_len
  - Osl\_Tag\_Base, [268](#)
- tag\_type
  - Osl\_TagTree, [270](#)
- tagAllocator
  - Osl\_Tag, [265](#)
- TagTree\_Allocator
  - osclmemory, [59](#)
- Tail
  - OslDoubleList, [367](#)
  - OslPriorityList, [469](#)
- tail
  - Osl\_Linked\_List\_Base, [215](#)
- takeOwnership
  - OSCLMemAutoPtr, [438](#)
- TDNSRequestParamAllocator
  - oscl\_dns\_param.h, [675](#)
- Tell
  - Osl\_File, [185](#)
  - OslAsyncFile, [318](#)
  - OslFileCache, [402](#)
  - OslNativeFile, [464](#)
- tellg
  - OslBinStream, [338](#)
- Terminate

- OscThread, [578](#)
- the\_list
  - Osc\_MTLLinked\_List, [227](#)
- THREAD\_1\_INACTIVE\_ERROR
  - OscProcStatus, [475](#)
- THREAD\_BLOCK\_ERROR
  - OscProcStatus, [476](#)
- THREAD\_NOT\_OWN\_MUTEX\_ERROR
  - OscProcStatus, [476](#)
- ThreadHasScheduler
  - PVThreadContext, [635](#)
- ThreadLogoff
  - OscIPSocketI, [419](#)
  - OscReadyQ, [485](#)
  - OscSocketI, [539](#)
  - OscSocketMethod, [547](#)
  - OscTCPSocket, [571](#)
  - OscTCPSocketI, [574](#)
  - OscUDPSocket, [606](#)
  - OscUDPSocketI, [608](#)
- ThreadLogon
  - OscIPSocketI, [419](#)
  - OscReadyQ, [485](#)
  - OscSocketI, [539](#)
  - OscSocketMethod, [547](#)
  - OscTCPSocket, [571](#)
  - OscTCPSocketI, [574](#)
  - OscUDPSocket, [606](#)
  - OscUDPSocketI, [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
  - OscTickCount, [580](#)
- TickCountFrequency
  - OscTickCount, [580](#)
- TickCountPeriod
  - OscTickCount, [580](#)
- TicksToMsec
  - OscTickCount, [580](#)
- TimeoutOccurred
  - OscTimerObserver, [590](#)
- TimerBaseElapsed
  - CallbackTimerObserver, [126](#)
  - OscTimer, [584](#)
- TimerCallback
  - OscReadyQ, [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](#)
- TLSSStorageOps, [656](#)
- TLSSStorageOps
  - get\_registry, [656](#)
  - save\_registry, [656](#)
- to\_msec
  - TimeValue, [655](#)
- to\_system\_time
  - NTPTIME, [169](#)
- TOO\_MANY\_FRAGS
  - BufFragStatusClass, [123](#)
- TOO\_MANY\_THREADS\_ERROR
  - OscProcStatus, [475](#)
- Top



- OslJump, [421](#)
- OslReadyQ, [485](#)
- OslTimerQ, [591](#)
- top
  - OslPriorityQueue, [472](#)
- TOSCL\_StringOp
  - osclutil, [70](#)
- TOSCL\_wStringOp
  - osclutil, [70](#)
- TOslBasicLockObject
  - osclconfig\_unix\_android.h, [842](#)
  - osclconfig\_unix\_common.h, [846](#)
- TOslConditionObject
  - osclconfig\_proc\_unix\_android.h, [834](#)
  - osclconfig\_proc\_unix\_common.h, [836](#)
- TOslFileHandle
  - osclio, [97](#)
- TOslFileOffset
  - osclconfig\_io.h, [821](#)
- TOslFileOffsetInt32
  - osclio, [97](#)
- TOslFileOp
  - osclio, [98](#)
- TOslHostent
  - osclconfig\_io.h, [821](#)
- TOslMutexObject
  - osclconfig\_proc\_unix\_android.h, [834](#)
  - osclconfig\_proc\_unix\_common.h, [836](#)
- TOslReady
  - osclproc, [106](#)
- TOslSemaphoreObject
  - osclconfig\_proc\_unix\_android.h, [834](#)
  - osclconfig\_proc\_unix\_common.h, [836](#)
- TOslSockAddr
  - osclconfig\_io.h, [821](#)
- TOslSockAddrLen
  - osclconfig\_io.h, [821](#)
- TOslSocket
  - osclconfig\_io.h, [821](#)
- TOslSocketServStatEvent
  - oscl\_socket\_stats.h, [770](#)
- TOslSocketStatEvent
  - oscl\_socket\_stats.h, [770](#)
- TOslThreadFuncArg
  - osclconfig\_proc\_unix\_android.h, [834](#)
  - osclconfig\_proc\_unix\_common.h, [836](#)
- TOslThreadFuncPtr
  - oscl\_thread.h, [788](#)
- TOslThreadFuncRet
  - osclconfig\_proc\_unix\_android.h, [834](#)
  - osclconfig\_proc\_unix\_common.h, [836](#)
- TOslThreadId
  - osclconfig\_proc\_unix\_android.h, [834](#)
  - osclconfig\_proc\_unix\_common.h, [836](#)
- TOslThreadObject
  - osclconfig\_proc\_unix\_android.h, [834](#)
  - osclconfig\_proc\_unix\_common.h, [836](#)
- TOslThreadTerminate
  - oscl\_thread.h, [789](#)
- TOslTlsKey
  - 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
  - OslExecSchedulerCommonBase, [394](#)
- TPVDNSEvent
  - osclio, [99](#)
- TPVDNSFxn
  - osclio, [99](#)
- TPVServicePrecedence
  - OslSocketTOS, [563](#)
- TPVServicePriority
  - OslSocketTOS, [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
  - OslErrorTrapImp, [377](#)
- TrapNoTls
  - OslErrorTrapImp, [377](#)
- TReadyQueLink, [657](#)
  - TReadyQueLink, [657](#)
- TReadyQueLink
  - iAOPriority, [657](#)
  - iIsIn, [657](#)
  - iSeqNum, [657](#)
  - iTimeQueuedTicks, [657](#)
  - iTimeToRunTicks, [657](#)
  - TReadyQueLink, [657](#)
- trim
  - OslMemPoolResizableAllocator, [452](#)
- TryLock
  - OslMutex, [460](#)



- TryWait
  - OslSemaphore, [522](#)
- TSocketServState
  - OslSocketServIBase, [558](#)
- TSymbianAccessMode
  - Osl\_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
  - OslSharedPtr, [529](#)
- UninstallScheduler
  - OslExecSchedulerCommonBase, [397](#)
- unix\_ntp\_offset
  - osclbase, [46](#)
- Unlock
  - OslLockBase, [424](#)
  - OslMutex, [460](#)
  - OslNullLock, [467](#)
  - OslThreadLock, [579](#)
- UnRegister
  - OslRegistryClient, [510](#)
  - OslRegistryClientImpl, [512](#)
  - OslRegistryServTlsImpl, [515](#)
- Unregister
  - OslComponentRegistry, [344](#)
- UnTrap
  - OslErrorTrapImp, [377](#)
- update
  - MM\_Stats\_t, [165](#)
- UpdateData
  - OslAsyncFileBuffer, [320](#)
- updateEnd
  - OslFileCacheBuffer, [404](#)
- updateStart
  - OslFileCacheBuffer, [404](#)
- updateStatsNode
  - MM\_Audit\_Imp, [159](#)
- updateStatsNodeInFailure
  - MM\_Audit\_Imp, [159](#)
- UpdateTimers
  - OslExecSchedulerCommonBase, [397](#)
- UpdateTimersMsec
  - OslExecSchedulerCommonBase, [397](#)
- upper\_bound
  - Osl\_Map, [221](#), [222](#)
- Osl\_Rb\_Tree, [244](#)
- usableSize
  - OslFileCacheBuffer, [404](#)
- USEC\_PER\_SEC
  - osclbase, [46](#)
- validate
  - MM\_Audit\_Imp, [159](#)
  - OslPriorityQueue, [473](#)
- validate\_all\_heap
  - MM\_Audit\_Imp, [159](#)
- validateblock
  - OslMemPoolResizableAllocator, [452](#)
- Value
  - OslAOSStatus, [315](#)
- value
  - Osl\_Rb\_Tree\_Node, [253](#)
  - Osl\_TagTree::Node, [280](#)
- value\_comp
  - Osl\_Map, [222](#)
- value\_compare
  - Osl\_Map::value\_compare, [223](#)
- value\_type
  - Osl\_Map, [218](#)
  - Osl\_Queue, [237](#)
  - Osl\_Rb\_Tree, [244](#)
  - Osl\_Rb\_Tree\_Const\_Iterator, [248](#)
  - Osl\_Rb\_Tree\_Iterator, [251](#)
  - Osl\_Rb\_Tree\_Node, [253](#)
  - Osl\_TagTree, [270](#)
  - Osl\_TAlloc, [282](#)
  - Osl\_Vector, [286](#)
  - OslPriorityQueue, [471](#)
- vec
  - OslPriorityQueue, [473](#)
- Wait
  - OslSemaphore, [522](#)
- WAIT\_ABANDONED\_ERROR
  - OslProcStatus, [476](#)
- WAIT\_TIMEOUT\_ERROR
  - OslProcStatus, [476](#)
- WaitAndPopTop
  - OslReadyQ, [485](#)
- WaitForReadyAO
  - OslExecSchedulerCommonBase, [397](#)
- WaitForRequestComplete
  - OslReadyQ, [485](#)
- WaitOnRequests
  - OslSocketServRequestList, [560](#)
- Wakeup
  - OslSocketServRequestList, [560](#)
- writable
  - CFastRep, [128](#)

## Write

- Osl\_File, [186](#)
- OslAsyncFile, [318](#)
- OslFileCache, [402](#)
- OslNativeFile, [464](#)

## write

- OSCL\_String, [264](#)
- OSCL\_wString, [307](#)
- OslBinOStream, [331](#)

## WriteUnsignedLong

- OslBinOStreamBigEndian, [333](#)
- OslBinOStreamLittleEndian, [335](#)

## WriteUnsignedShort

- OslBinOStreamBigEndian, [333](#)
- OslBinOStreamLittleEndian, [335](#)

## WriteUpdatesToFile

- OslFileCacheBuffer, [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

- OslPtr, [477](#)
- OslPtrC, [480](#)