



packetvideo™

OSCL API

Build Version: CORE_7.501.1.1

August 6, 2009

Contents

1 oscl Module Index	1
1.1 oscl Modules	1
2 oscl Hierarchical Index	2
2.1 oscl Class Hierarchy	2
3 oscl Data Structure Index	8
3.1 oscl Data Structures	8
4 oscl File Index	14
4.1 oscl File List	14
5 oscl Page Index	19
5.1 oscl Related Pages	19
6 oscl Module Documentation	20
6.1 OSCL config	20
6.2 OSCL Base	24
6.3 OSCL Memory	45
6.4 OSCL Util	61
6.5 OSCL Error	83
6.6 OSCL IO	93
6.7 OSCL Proc	101
6.8 OSCL Init	105
7 oscl Data Structure Documentation	106
7.1 _OsclBasicAllocator Class Reference	106
7.2 _OsclHeapBase Class Reference	108
7.3 AcceptParam Class Reference	110
7.4 allocator Class Reference	111

7.5	AllPassFilter Class Reference	112
7.6	BindParam Class Reference	114
7.7	BufferFragment Class Reference	115
7.8	BufferMgr Class Reference	116
7.9	BufferState Class Reference	117
7.10	BuFragGroup< ChainClass, max_frags > Class Template Reference	118
7.11	BuffFragStatusClass Class Reference	121
7.12	CallbackTimer< Alloc > Class Template Reference	122
7.13	CallbackTimerObserver Class Reference	124
7.14	CFastRep Class Reference	125
7.15	CHheapRep Class Reference	127
7.16	ConnectParam Class Reference	129
7.17	CStackRep Class Reference	130
7.18	DNSRequestParam Class Reference	131
7.19	GetHostByNameParam Class Reference	133
7.20	HeapBase Class Reference	134
7.21	internalLeave Class Reference	136
7.22	LinkedListElement< LLClass > Class Template Reference	137
7.23	ListenParam Class Reference	138
7.24	MediaData< ChainClass, max_frags, local_bufsize > Class Template Reference	139
7.25	MediaStatusClass Class Reference	142
7.26	MemAllocator< T > Class Template Reference	143
7.27	MM_AllocBlockFence Struct Reference	144
7.28	MM_AllocBlockHdr Struct Reference	145
7.29	MM_AllocInfo Struct Reference	146
7.30	MM_AllocNode Struct Reference	148
7.31	MM_AllocQueryInfo Struct Reference	149
7.32	MM_Audit_Imp Class Reference	150
7.33	MM_AuditOverheadStats Struct Reference	158
7.34	MM_FailInsertParam Struct Reference	159
7.35	MM_Stats_CB Struct Reference	160
7.36	MM_Stats_t Struct Reference	161
7.37	NTPTTime Class Reference	163
7.38	Oscl_Alloc Class Reference	167
7.39	Oscl_Dealloc Class Reference	168
7.40	Oscl_DefAlloc Class Reference	169

7.41 Oscl_DefAllocWithRefCounter< DefAlloc > Class Template Reference	170
7.42 OSCL_FastString Class Reference	172
7.43 Oscl_File Class Reference	176
7.44 Oscl_File::OsclCacheObserver Class Reference	184
7.45 Oscl_File::OsclFixedCacheParam Class Reference	185
7.46 Oscl_FileFind Class Reference	186
7.47 Oscl_FileServer Class Reference	190
7.48 oscl_fsstat Struct Reference	192
7.49 OSCL_HeapString< Alloc > Class Template Reference	193
7.50 OSCL_HeapStringA Class Reference	195
7.51 Oscl_Int64_Utils Class Reference	200
7.52 Oscl_Less< T > Struct Template Reference	202
7.53 Oscl_Linked_List< LLClass, Alloc > Class Template Reference	203
7.54 Oscl_Linked_List_Base Class Reference	207
7.55 Oscl_Map< Key, T, Alloc, Compare > Class Template Reference	211
7.56 Oscl_Map< Key, T, Alloc, Compare >::value_compare Class Reference	218
7.57 Oscl_MTLinked_List< LLClass, Alloc, TheLock > Class Template Reference	220
7.58 Oscl_Opaque_Type_Alloc Class Reference	224
7.59 Oscl_Opaque_Type_Alloc_LL Class Reference	225
7.60 Oscl_Opaque_Type_Compare Class Reference	227
7.61 Oscl_Pair< T1, T2 > Struct Template Reference	229
7.62 Oscl_Queue< T, Alloc > Class Template Reference	230
7.63 Oscl_Queue_Base Class Reference	233
7.64 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference	236
7.65 Oscl_Rb_Tree_Base Class Reference	240
7.66 Oscl_Rb_Tree_Const_Iterator< Value > Struct Template Reference	241
7.67 Oscl_Rb_Tree_Iterator< Value > Struct Template Reference	244
7.68 Oscl_Rb_Tree_Node< Value > Struct Template Reference	247
7.69 Oscl_Rb_Tree_Node_Base Struct Reference	248
7.70 Oscl_Select1st< V, U > Struct Template Reference	250
7.71 OSCL_StackString< MaxBufSize > Class Template Reference	251
7.72 oscl_stat_buf Struct Reference	253
7.73 OSCL_String Class Reference	254
7.74 Oscl_Tag< Alloc > Struct Template Reference	260
7.75 Oscl_Tag_Base Struct Reference	262
7.76 Oscl_TagTree< T, Alloc > Class Template Reference	264

7.77 Oscl_TagTree< T, Alloc >::const_iterator Struct Reference	268
7.78 Oscl_TagTree< T, Alloc >::iterator Struct Reference	271
7.79 Oscl_TagTree< T, Alloc >::Node Struct Reference	274
7.80 Oscl_TAlloc< T, Alloc > Class Template Reference	276
7.81 Oscl_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference	279
7.82 Oscl_Vector< T, Alloc > Class Template Reference	280
7.83 Oscl_Vector_Base Class Reference	285
7.84 OSCL_wFastString Class Reference	289
7.85 OSCL_wHeapString< Alloc > Class Template Reference	292
7.86 OSCL_wHeapStringA Class Reference	294
7.87 OSCL_wStackString< MaxBufSize > Class Template Reference	297
7.88 OSCL_wString Class Reference	299
7.89 OsclAcceptMethod Class Reference	303
7.90 OsclAcceptRequest Class Reference	304
7.91 OsclActiveObject Class Reference	305
7.92 OsclAllocDestructDealloc Class Reference	309
7.93 OsclAOStatus Class Reference	310
7.94 OsclAsyncFile Class Reference	311
7.95 OsclAsyncFileBuffer Class Reference	314
7.96 OsclAuditCB Class Reference	316
7.97 OsclBindMethod Class Reference	317
7.98 OsclBindRequest Class Reference	318
7.99 OsclBinIStream Class Reference	319
7.100 OsclBinIStreamBigEndian Class Reference	321
7.101 OsclBinIStreamLittleEndian Class Reference	324
7.102 OsclBinOStream Class Reference	326
7.103 OsclBinOStreamBigEndian Class Reference	327
7.104 OsclBinOStreamLittleEndian Class Reference	329
7.105 OsclBinStream Class Reference	331
7.106 OsclBuf Class Reference	335
7.107 OsclCompareLess< T > Class Template Reference	337
7.108 OsclComponentRegistry Class Reference	338
7.109 OsclComponentRegistryData Class Reference	340
7.110 OsclComponentRegistryElement Class Reference	341
7.111 OsclConnectMethod Class Reference	343
7.112 OsclConnectRequest Class Reference	344

7.113OsclDestructDealloc Class Reference	345
7.114OsclDNS Class Reference	346
7.115OsclDNSI Class Reference	348
7.116OsclDNSIBase Class Reference	350
7.117OsclDNSMethod Class Reference	353
7.118OsclDNSObserver Class Reference	356
7.119OsclDNSRequest Class Reference	357
7.120OsclDNSRequestAO Class Reference	358
7.121OsclDoubleLink Class Reference	361
7.122OsclDoubleList< T > Class Template Reference	362
7.123OsclDoubleListBase Class Reference	363
7.124OsclDoubleRunner< T > Class Template Reference	365
7.125OsclError Class Reference	367
7.126OsclErrorAllocator Class Reference	369
7.127OsclErrorTrap Class Reference	371
7.128OsclErrorTrapImp Class Reference	372
7.129OsclException< LeaveCode > Class Template Reference	374
7.130OsclExclusiveArrayPtr< T > Class Template Reference	375
7.131OsclExclusivePtr< T > Class Template Reference	378
7.132OsclExclusivePtrA< T, Alloc > Class Template Reference	381
7.133OsclExecScheduler Class Reference	384
7.134OsclExecSchedulerBase Class Reference	386
7.135OsclExecSchedulerCommonBase Class Reference	387
7.136OsclFileCache Class Reference	396
7.137OsclFileCacheBuffer Class Reference	398
7.138OsclFileHandle Class Reference	400
7.139OsclFileStats Class Reference	401
7.140OsclFileStatsItem Class Reference	402
7.141OsclGetHostByNameMethod Class Reference	403
7.142OsclGetHostByNameRequest Class Reference	404
7.143OsclInit Class Reference	405
7.144OsclInteger64Transport Struct Reference	406
7.145OsclIPSocketI Class Reference	407
7.146OsclJump Class Reference	410
7.147OsclListenMethod Class Reference	411
7.148OsclListenRequest Class Reference	412

7.149OsclLockBase Class Reference	413
7.150OsclMem Class Reference	414
7.151OsclMemAllocator Class Reference	415
7.152OsclMemAllocDestructDealloc< T > Class Template Reference	416
7.153OsclMemAudit Class Reference	418
7.154OSCLMemAutoPtr< T, _Allocator > Class Template Reference	424
7.155OsclMemBasicAllocator Class Reference	428
7.156OsclMemBasicAllocDestructDealloc< T > Class Template Reference	429
7.157OsclMemGlobalAuditObject Class Reference	430
7.158OsclMemoryFragment Struct Reference	431
7.159OsclMemPoolAllocator Class Reference	432
7.160OsclMemPoolFixedChunkAllocator Class Reference	433
7.161OsclMemPoolFixedChunkAllocatorObserver Class Reference	437
7.162OsclMemPoolResizableAllocator Class Reference	438
7.163OsclMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference	444
7.164OsclMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference	445
7.165OsclMemPoolResizableAllocatorMemoryObserver Class Reference	446
7.166OsclMemPoolResizableAllocatorObserver Class Reference	447
7.167OsclMemStatsNode Class Reference	448
7.168OsclMutex Class Reference	449
7.169OsclNameString< __len > Class Template Reference	451
7.170OsclNativeFile Class Reference	452
7.171OsclNativeFileParams Class Reference	455
7.172OsclNetworkAddress Class Reference	456
7.173OsclNullLock Class Reference	457
7.174OsclPriorityLink Class Reference	458
7.175OsclPriorityList< T > Class Template Reference	459
7.176OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference	460
7.177OsclPriorityQueueBase Class Reference	464
7.178OsclProcStatus Class Reference	465
7.179OsclPtr Class Reference	467
7.180OsclPtrC Class Reference	469
7.181OsclRand Class Reference	471
7.182OsclReadyAlloc Class Reference	472
7.183OsclReadyCompare Class Reference	473
7.184OsclReadyQ Class Reference	474

7.185OsclRecvFromMethod Class Reference	476
7.186OsclRecvFromRequest Class Reference	478
7.187OsclRecvMethod Class Reference	480
7.188OsclRecvRequest Class Reference	481
7.189OsclRefCounter Class Reference	482
7.190OsclRefCounterDA Class Reference	484
7.191OsclRefCounterMemFrag Class Reference	486
7.192OsclRefCounterMTDA< LockType > Class Template Reference	488
7.193OsclRefCounterMTSA< DeallocType, LockType > Class Template Reference	490
7.194OsclRefCounterSA< DeallocType > Class Template Reference	492
7.195OsclRegistryAccessClient Class Reference	494
7.196OsclRegistryAccessClientImpl Class Reference	496
7.197OsclRegistryAccessClientTlsImpl Class Reference	497
7.198OsclRegistryAccessElement Class Reference	498
7.199OsclRegistryClient Class Reference	499
7.200OsclRegistryClientImpl Class Reference	501
7.201OsclRegistryClientTlsImpl Class Reference	503
7.202OsclRegistryServTlsImpl Class Reference	504
7.203OsclScheduler Class Reference	506
7.204OsclSchedulerObserver Class Reference	507
7.205OsclScopedLock< LockClass > Class Template Reference	508
7.206OsclSelect Class Reference	509
7.207OsclSemaphore Class Reference	511
7.208OsclSendMethod Class Reference	513
7.209OsclSendRequest Class Reference	514
7.210OsclSendToMethod Class Reference	515
7.211OsclSendToRequest Class Reference	516
7.212OsclSharedPtr< TheClass > Class Template Reference	517
7.213OsclShutdownMethod Class Reference	520
7.214OsclShutdownRequest Class Reference	521
7.215OsclSingleton< T, ID, Registry > Class Template Reference	522
7.216OsclSingletonRegistry Class Reference	524
7.217OsclSocketI Class Reference	525
7.218OsclSocketIBase Class Reference	530
7.219OsclSocketMethod Class Reference	535
7.220OsclSocketObserver Class Reference	538

7.221OsclSocketRequest Class Reference	539
7.222OsclSocketRequestAO Class Reference	540
7.223OsclSocketServ Class Reference	544
7.224OsclSocketServI Class Reference	546
7.225OsclSocketServIBase Class Reference	548
7.226OsclSocketServRequestList Class Reference	550
7.227OsclSocketServRequestQElem Class Reference	552
7.228OsclTCPSocket Class Reference	553
7.229OsclTCPSocketI Class Reference	559
7.230OsclThread Class Reference	562
7.231OsclThreadLock Class Reference	566
7.232OsclTickCount Class Reference	567
7.233OsclTimer< Alloc > Class Template Reference	569
7.234OsclTimerCompare Class Reference	572
7.235OsclTimerObject Class Reference	573
7.236OsclTimerObserver Class Reference	577
7.237OsclTimerQ Class Reference	578
7.238OsclTLS< T, ID, Registry > Class Template Reference	579
7.239OsclTLSEx< T, ID, Registry > Class Template Reference	581
7.240OsclTLSRegistry Class Reference	583
7.241OsclTLSRegistryEx Class Reference	584
7.242OsclTrapItem Class Reference	585
7.243OsclTrapStack Class Reference	586
7.244OsclTrapStackItem Class Reference	587
7.245OsclUDPSocket Class Reference	588
7.246OsclUDPSocketI Class Reference	593
7.247OsclUuid Struct Reference	595
7.248PVActiveBase Class Reference	597
7.249PVActiveStats Class Reference	601
7.250PVLogger Class Reference	602
7.251PVLoggerAppender Class Reference	608
7.252PVLoggerFilter Class Reference	609
7.253PVLoggerLayout Class Reference	611
7.254PVLoggerRegistry Class Reference	613
7.255PVSchedulerStopper Class Reference	616
7.256PVSockBufRecv Class Reference	617

7.257PVSockBufSend Class Reference	618
7.258PVThreadContext Class Reference	619
7.259RecvFromParam Class Reference	621
7.260RecvParam Class Reference	623
7.261SendParam Class Reference	624
7.262SendToParam Class Reference	625
7.263ShutdownParam Class Reference	626
7.264SocketRequestParam Class Reference	627
7.265StrCSumPtrLen Struct Reference	629
7.266StrPtrLen Struct Reference	632
7.267TimeValue Class Reference	634
7.268TLSStorageOps Class Reference	640
7.269TReadyQueLink Class Reference	641
7.270WStrPtrLen Struct Reference	642
8 oscl File Documentation	644
8.1 oscl_aostatus.h File Reference	644
8.2 oscl_assert.h File Reference	645
8.3 oscl_base.h File Reference	646
8.4 oscl_base_alloc.h File Reference	647
8.5 oscl_base_macros.h File Reference	648
8.6 oscl_bin_stream.h File Reference	649
8.7 oscl_byte_order.h File Reference	650
8.8 oscl_defalloc.h File Reference	651
8.9 oscl_dll.h File Reference	652
8.10 oscl_dns.h File Reference	653
8.11 oscl_dns_gethostname.h File Reference	654
8.12 oscl_dns_imp.h File Reference	655
8.13 oscl_dns_imp_base.h File Reference	656
8.14 oscl_dns_imp_pv.h File Reference	657
8.15 oscl_dns_method.h File Reference	658
8.16 oscl_dns_param.h File Reference	659
8.17 oscl_dns_request.h File Reference	660
8.18 oscl_dns_tuneables.h File Reference	661
8.19 oscl_double_list.h File Reference	662
8.20 oscl_errno.h File Reference	663
8.21 oscl_error.h File Reference	664

8.22 oscl_error_allocator.h File Reference	665
8.23 oscl_error_codes.h File Reference	666
8.24 oscl_error_imp.h File Reference	667
8.25 oscl_error_imp_cppexceptions.h File Reference	668
8.26 oscl_error_imp_fatalerror.h File Reference	669
8.27 oscl_error_imp_jumps.h File Reference	670
8.28 oscl_error_trapcleanup.h File Reference	672
8.29 oscl_exception.h File Reference	673
8.30 oscl_exclusive_ptr.h File Reference	674
8.31 oscl_file_async_read.h File Reference	675
8.32 oscl_file_cache.h File Reference	676
8.33 oscl_file_dir_utils.h File Reference	677
8.34 oscl_file_find.h File Reference	679
8.35 oscl_file_handle.h File Reference	680
8.36 oscl_file_io.h File Reference	681
8.37 oscl_file_native.h File Reference	682
8.38 oscl_file_server.h File Reference	683
8.39 oscl_file_stats.h File Reference	684
8.40 oscl_file_types.h File Reference	685
8.41 oscl_heapbase.h File Reference	686
8.42 oscl_init.h File Reference	687
8.43 oscl_int64_utils.h File Reference	688
8.44 oscl_ip_socket.h File Reference	689
8.45 oscl_linked_list.h File Reference	690
8.46 oscl_lock_base.h File Reference	691
8.47 oscl_map.h File Reference	692
8.48 oscl_math.h File Reference	693
8.49 oscl_media_data.h File Reference	694
8.50 oscl_media_status.h File Reference	695
8.51 oscl_mem.h File Reference	696
8.52 oscl_mem_align.h File Reference	699
8.53 oscl_mem_audit.h File Reference	700
8.54 oscl_mem_audit_internals.h File Reference	702
8.55 oscl_mem_auto_ptr.h File Reference	703
8.56 oscl_mem_basic_functions.h File Reference	704
8.57 oscl_mem_inst.h File Reference	705

8.58 oscl_mem_mempool.h File Reference	706
8.59 oscl_mempool_allocator.h File Reference	707
8.60 oscl_mutex.h File Reference	708
8.61 oscl_namestring.h File Reference	709
8.62 oscl_opaque_type.h File Reference	710
8.63 oscl_priqueue.h File Reference	711
8.64 oscl_procstatus.h File Reference	712
8.65 oscl_queue.h File Reference	713
8.66 oscl_rand.h File Reference	714
8.67 oscl_refcounter.h File Reference	715
8.68 oscl_refcounter_memfrag.h File Reference	716
8.69 oscl_registry_access_client.h File Reference	717
8.70 oscl_registry_client.h File Reference	718
8.71 oscl_registry_client_impl.h File Reference	719
8.72 oscl_registry_serv_impl.h File Reference	720
8.73 oscl_registry_serv_impl_global.h File Reference	721
8.74 oscl_registry_serv_impl_tls.h File Reference	722
8.75 oscl_registry_types.h File Reference	723
8.76 oscl_scheduler.h File Reference	724
8.77 oscl_scheduler_ao.h File Reference	725
8.78 oscl_scheduler_aobase.h File Reference	726
8.79 oscl_scheduler_readyq.h File Reference	727
8.80 oscl_scheduler_threadcontext.h File Reference	728
8.81 oscl_scheduler_tuneables.h File Reference	729
8.82 oscl_scheduler_types.h File Reference	730
8.83 oscl_semaphore.h File Reference	731
8.84 oscl_shared_ptr.h File Reference	732
8.85 oscl_singleton.h File Reference	733
8.86 oscl_snprintf.h File Reference	735
8.87 oscl_socket.h File Reference	736
8.88 oscl_socket_accept.h File Reference	737
8.89 oscl_socket_bind.h File Reference	738
8.90 oscl_socket_connect.h File Reference	739
8.91 oscl_socket_imp.h File Reference	740
8.92 oscl_socket_imp_base.h File Reference	741
8.93 oscl_socket_imp_pv.h File Reference	742

8.94 oscl_socket_listen.h File Reference	743
8.95 oscl_socket_method.h File Reference	744
8.96 oscl_socket_recv.h File Reference	745
8.97 oscl_socket_recv_from.h File Reference	746
8.98 oscl_socket_request.h File Reference	747
8.99 oscl_socket_send.h File Reference	748
8.100oscl_socket_send_to.h File Reference	749
8.101oscl_socket_serv_imp.h File Reference	750
8.102oscl_socket_serv_imp_base.h File Reference	751
8.103oscl_socket_serv_imp_pv.h File Reference	752
8.104oscl_socket_serv_imp_reqlist.h File Reference	753
8.105oscl_socket_shutdown.h File Reference	754
8.106oscl_socket_stats.h File Reference	755
8.107oscl_socket_tuneables.h File Reference	757
8.108oscl_socket_types.h File Reference	759
8.109oscl_stdstring.h File Reference	761
8.110oscl_str_ptr_len.h File Reference	763
8.111oscl_string.h File Reference	764
8.112oscl_string_containers.h File Reference	765
8.113oscl_string_rep.h File Reference	766
8.114oscl_string_uri.h File Reference	767
8.115oscl_string_utf8.h File Reference	768
8.116oscl_string_utils.h File Reference	769
8.117oscl_string_xml.h File Reference	770
8.118oscl_tagtree.h File Reference	771
8.119oscl_tcp_socket.h File Reference	772
8.120oscl_thread.h File Reference	773
8.121oscl_tickcount.h File Reference	775
8.122oscl_time.h File Reference	776
8.123oscl_timer.h File Reference	778
8.124oscl_tls.h File Reference	779
8.125oscl_tree.h File Reference	780
8.126oscl_types.h File Reference	781
8.127oscl_udp_socket.h File Reference	782
8.128oscl_utf8conv.h File Reference	783
8.129oscl_uuid.h File Reference	784

8.130oscl_vector.h File Reference	785
8.131osclconfig.h File Reference	786
8.132osclconfig_ansi_memory.h File Reference	788
8.133osclconfig_check.h File Reference	789
8.134osclconfig_compiler_warnings.h File Reference	790
8.135osclconfig_error.h File Reference	791
8.136osclconfig_error_check.h File Reference	792
8.137osclconfig_global_new_delete.h File Reference	793
8.138osclconfig_global_placement_new.h File Reference	794
8.139osclconfig_io.h File Reference	795
8.140osclconfig_io_check.h File Reference	802
8.141osclconfig_ix86.h File Reference	803
8.142osclconfig_lib.h File Reference	804
8.143osclconfig_lib_check.h File Reference	805
8.144osclconfig_limits_typedefs.h File Reference	806
8.145osclconfig_memory.h File Reference	807
8.146osclconfig_memory_check.h File Reference	808
8.147osclconfig_no_os.h File Reference	809
8.148osclconfig_proc.h File Reference	810
8.149osclconfig_proc_check.h File Reference	811
8.150osclconfig_proc_unix_android.h File Reference	813
8.151osclconfig_proc_unix_common.h File Reference	815
8.152osclconfig_time.h File Reference	817
8.153osclconfig_time_check.h File Reference	818
8.154osclconfig_unix_android.h File Reference	819
8.155osclconfig_unix_common.h File Reference	823
8.156osclconfig_util.h File Reference	827
8.157osclconfig_util_check.h File Reference	828
8.158pvlogger.h File Reference	829
8.159pvlogger_accessories.h File Reference	837
8.160pvlogger_c.h File Reference	838
8.161pvlogger_registry.h File Reference	840
9 oscl Page Documentation	841
9.1 Todo List	841

Chapter 1

oscl Module Index

1.1 oscl Modules

Here is a list of all modules:

OSCL config	20
OSCL Base	24
OSCL Memory	45
OSCL Util	61
OSCL Error	83
OSCL IO	93
OSCL Proc	101
OSCL Init	105

Chapter 2

oscl Hierarchical Index

2.1 oscl Class Hierarchy

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

_OscIHeapBase	108
HeapBase	134
Oscl_File	176
OSCL_String	254
OSCL_FastString	172
OSCL_HeapString< Alloc >	193
OSCL_HeapStringA	195
OSCL_StackString< MaxBufSize >	251
OsclActiveObject	305
OsclAsyncFile	311
OsclDNSRequestAO	358
OsclGetHostNameRequest	404
OsclSocketRequestAO	540
OsclAcceptRequest	304
OsclBindRequest	318
OsclConnectRequest	344
OsclListenRequest	412
OsclRecvFromRequest	478
OsclRecvRequest	481
OsclSendRequest	514
OsclSendToRequest	516
OsclShutdownRequest	521
PVSchedulerStopper	616
OsclAsyncFileBuffer	314
OsclBuf	335
OsclDNS	346
OsclFileCache	396
OsclNativeFile	452
OsclPtr	467
OsclPtrC	469
OsclRegistryClient	499
OsclSocketServ	544
OsclTCPSocket	553

OsclTimerObject	573
CallbackTimer< Alloc >	122
OsclDNSMethod	353
OsclGetHostByNameMethod	403
OsclSocketMethod	535
OsclAcceptMethod	303
OsclBindMethod	317
OsclConnectMethod	343
OsclListenMethod	411
OsclRecvFromMethod	476
OsclRecvMethod	480
OsclSendMethod	513
OsclSendToMethod	515
OsclShutdownMethod	520
OsclSocketServI	546
OsclUDPSocket	588
OsclExecSchedulerBase	386
OsclExecScheduler	384
allocator	111
BufferMgr	116
BufferState	117
BufFragGroup< ChainClass, max_frags >	118
MediaData< ChainClass, max_frags, local_bufsize >	139
BufFragStatusClass	121
MediaStatusClass	142
CallbackTimerObserver	124
OsclTimer< Alloc >	569
CFastRep	125
CHheapRep	127
CStackRep	130
DNSRequestParam	131
GetHostByNameParam	133
internalLeave	136
LinkedListElement< LLClass >	137
MemAllocator< T >	143
MM_AllocBlockFence	144
MM_AllocBlockHdr	145
MM_AllocInfo	146
MM_AllocNode	148
MM_AllocQueryInfo	149
MM_Audit_Imp	150
MM_AuditOverheadStats	158
MM_FailInsertParam	159
MM_Stats_CB	160
MM_Stats_t	161
NTPTime	163
Oscl_Alloc	167
Oscl_DefAlloc	169
_OsclBasicAllocator	106
OsclAllocDestructDealloc	309
OsclMemAllocDestructDealloc< T >	416
OsclMemBasicAllocDestructDealloc< T >	429

OsclMemAllocator	415
OsclMemBasicAllocator	428
OsclMemPoolFixedChunkAllocator	433
OsclMemPoolResizableAllocator	438
OsclReadyAlloc	472
Oscl_Dealloc	168
Oscl_DefAlloc	169
Oscl_File::OsclCacheObserver	184
Oscl_File::OsclFixedCacheParam	185
Oscl_FileFind	186
Oscl_FileServer	190
oscl_fsstat	192
Oscl_Int64_Utils	200
Oscl_Less< T >	202
Oscl_Linked_List_Base	207
Oscl_Linked_List< LLClass, Alloc >	203
Oscl_Map< Key, T, Alloc, Compare >	211
Oscl_Map< Key, T, Alloc, Compare >::value_compare	218
Oscl_MTLinked_List< LLClass, Alloc, TheLock >	220
Oscl_Opaque_Type_Alloc	224
Oscl_Queue< T, Alloc >	230
Oscl_Vector< T, Alloc >	280
Oscl_Vector< TOsclReady, OsclReadyAlloc >	280
Oscl_Opaque_Type_Alloc_LL	225
Oscl_Linked_List< LLClass, Alloc >	203
Oscl_Opaque_Type_Compare	227
OsclPriorityQueue< Qelem, Alloc, Container, Compare >	460
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclReadyCompare >	460
OsclReadyQ	474
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclTimerCompare >	460
OsclTimerQ	578
Oscl_Pair< T1, T2 >	229
Oscl_Queue_Base	233
Oscl_Queue< T, Alloc >	230
Oscl_Rb_Tree_Base	240
Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >	236
Oscl_Rb_Tree_Const_Iterator< Value >	241
Oscl_Rb_Tree_Iterator< Value >	244
Oscl_Rb_Tree_Node_Base	248
Oscl_Rb_Tree_Node< Value >	247
Oscl_Select1st< V, U >	250
oscl_stat_buf	253
Oscl_Tag_Base	262
Oscl_Tag< Alloc >	260
Oscl_TagTree< T, Alloc >	264
Oscl_TagTree< T, Alloc >::const_iterator	268
Oscl_TagTree< T, Alloc >::iterator	271
Oscl_TagTree< T, Alloc >::Node	274
Oscl_TAlloc< T, Alloc >::rebind< U, V >	279

Oscl_Vector_Base	285
Oscl_Vector< T, Alloc >	280
Oscl_Vector< TOsclReady, OsclReadyAlloc >	280
OSCL_wString	299
OSCL_wFastString	289
OSCL_wHeapString< Alloc >	292
OSCL_wHeapStringA	294
OSCL_wStackString< MaxBufSize >	297
OsclAOStatus	310
OsclAuditCB	316
OsclBinStream	331
OsclBinIStream	319
OsclBinIStreamBigEndian	321
OsclBinIStreamLittleEndian	324
OsclBinOStream	326
OsclBinOStreamBigEndian	327
OsclBinOStreamLittleEndian	329
OsclCompareLess< T >	337
OsclComponentRegistry	338
OsclComponentRegistryData	340
OsclComponentRegistryElement	341
OsclDestructDealloc	345
Oscl_TAlloc< T, Alloc >	276
OsclAllocDestructDealloc	309
OsclDNSIBase	350
OsclDNSI	348
OsclDNSObserver	356
OsclDNSRequest	357
OsclDoubleLink	361
OsclPriorityLink	458
OsclDoubleListBase	363
OsclDoubleList< T >	362
OsclPriorityList< T >	459
OsclDoubleRunner< T >	365
OsclError	367
OsclErrorAllocator	369
OsclErrorTrap	371
OsclErrorTrapImp	372
OsclException< LeaveCode >	374
OsclExclusiveArrayPtr< T >	375
OsclExclusivePtr< T >	378
OsclExclusivePtrA< T, Alloc >	381
OsclExecSchedulerCommonBase	387
OsclExecScheduler	384
OsclFileCacheBuffer	398
OsclFileHandle	400
OsclFileStats	401
OsclFileStatsItem	402
OsclInit	405
OsclInteger64Transport	406
OsclIPSocketI	407

OsclTCPSocketI	559
OsclUDPSocketI	593
OsclJump	410
OsclLockBase	413
OsclMutex	449
OsclNullLock	457
OsclThreadLock	566
OsclMem	414
OsclMemAudit	418
OSCLMemAutoPtr< T, _Allocator >	424
OsclMemGlobalAuditObject	430
OsclMemoryFragment	431
BufferFragment	115
OsclMemPoolAllocator	432
OsclMemPoolFixedChunkAllocatorObserver	437
OsclMemPoolResizableAllocator::MemPoolBlockInfo	444
OsclMemPoolResizableAllocator::MemPoolBufferInfo	445
OsclMemPoolResizableAllocatorMemoryObserver	446
OsclMemPoolResizableAllocatorObserver	447
OsclMemStatsNode	448
OsclNameString< __len >	451
OsclNativeFileParams	455
OsclNetworkAddress	456
OsclPriorityQueueBase	464
OsclPriorityQueue< Qelem, Alloc, Container, Compare >	460
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclReadyCompare >	460
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclTimerCompare >	460
OsclProcStatus	465
OsclRand	471
OsclReadyCompare	473
OsclRefCounter	482
Oscl_DefAllocWithRefCounter< DefAlloc >	170
OsclRefCounterDA	484
OsclRefCounterMTDA< LockType >	488
OsclRefCounterMTSA< DeallocType, LockType >	490
OsclRefCounterSA< DeallocType >	492
OsclRefCounterMemFrag	486
OsclRegistryAccessClient	494
OsclRegistryAccessElement	498
OsclRegistryClientImpl	501
OsclRegistryAccessClientImpl	496
OsclRegistryServTlsImpl	504
OsclRegistryAccessClientTlsImpl	497
OsclRegistryClientTlsImpl	503
OsclScheduler	506
OsclSchedulerObserver	507
OsclScopedLock< LockClass >	508
OsclSelect	509
OsclSemaphore	511
OsclSharedPtr< TheClass >	517

OsclSingleton< T, ID, Registry >	522
OsclSingletonRegistry	524
OsclSocketIBase	530
OsclSocketI	525
OsclSocketObserver	538
OsclSocketRequest	539
OsclSocketServIBase	548
OsclSocketServI	546
OsclSocketServRequestList	550
OsclSocketServRequestQElem	552
OsclThread	562
OsclTickCount	567
OsclTimerCompare	572
OsclTimerObserver	577
OsclTLS< T, ID, Registry >	579
OsclTLSEx< T, ID, Registry >	581
OsclTLSRegistry	583
OsclTLSRegistryEx	584
OsclTrapItem	585
OsclTrapStack	586
OsclTrapStackItem	587
OsclUuid	595
PVActiveBase	597
OsclActiveObject	305
OsclTimerObject	573
PVActiveStats	601
PVLogger	602
PVLoggerAppender	608
PVLoggerFilter	609
AllPassFilter	112
PVLoggerLayout	611
PVLoggerRegistry	613
PVSockBufRecv	617
PVSockBufSend	618
PVThreadContext	619
SocketRequestParam	627
AcceptParam	110
BindParam	114
ConnectParam	129
ListenParam	138
RecvFromParam	621
RecvParam	623
SendParam	624
SendToParam	625
ShutdownParam	626
StrPtrLen	632
StrCSumPtrLen	629
TimeValue	634
TLSStorageOps	640
TReadyQueLink	641
WStrPtrLen	642

Chapter 3

oscl Data Structure Index

3.1 oscl Data Structures

Here are the data structures with brief descriptions:

_OsclBasicAllocator	106
_OsclHeapBase	108
AcceptParam	110
allocator	111
AllPassFilter	112
BindParam	114
BufferFragment	115
BufferMgr	116
BufferState	117
BufFragGroup< ChainClass, max_frags >	118
BufFragStatusClass	121
CallbackTimer< Alloc >	122
CallbackTimerObserver	124
CFastRep	125
CHheapRep	127
ConnectParam	129
CStackRep	130
DNSRequestParam	131
GetHostNameParam	133
HeapBase	134
internalLeave	136
LinkedListElement< LLClass >	137
ListenParam	138
MediaData< ChainClass, max_frags, local_bufsize >	139
MediaStatusClass	142
MemAllocator< T >	143
MM_AllocBlockFence	144
MM_AllocBlockHdr	145
MM_AllocInfo	146
MM_AllocNode	148
MM_AllocQueryInfo	149
MM_Audit_Imp	150
MM_AuditOverheadStats	158

MM_FailInsertParam	159
MM_Stats_CB	160
MM_Stats_t	161
NTPTime (Time value as the number of seconds since 0h (UTC) Jan. 1, 1900)	163
OscI_Alloc	167
OscI_Dealloc	168
OscI_DefAlloc	169
OscI_DefAllocWithRefCounter< DefAlloc >	170
OSCL_FastString	172
OscI_File	176
OscI_File::OsclCacheObserver	184
OscI_File::OsclFixedCacheParam	185
OscI_FileFind	186
OscI_FileServer	190
oscl_fstat	192
OSCL_HeapString< Alloc >	193
OSCL_HeapStringA	195
OscI_Int64_Utils (Wrapper for commonly used int64/uint64 operations)	200
OscI_Less< T >	202
OscI_Linked_List< LLClass, Alloc >	203
OscI_Linked_List_Base	207
OscI_Map< Key, T, Alloc, Compare >	211
OscI_Map< Key, T, Alloc, Compare >::value_compare	218
OscI_MTLinked_List< LLClass, Alloc, TheLock >	220
OscI_Opaque_Type_Alloc	224
OscI_Opaque_Type_Alloc_LL	225
OscI_Opaque_Type_Compare	227
OscI_Pair< T1, T2 >	229
OscI_Queue< T, Alloc >	230
OscI_Queue_Base	233
OscI_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >	236
OscI_Rb_Tree_Base	240
OscI_Rb_Tree_Const_Iterator< Value >	241
OscI_Rb_Tree_Iterator< Value >	244
OscI_Rb_Tree_Node< Value >	247
OscI_Rb_Tree_Node_Base	248
OscI_Select1st< V, U >	250
OSCL_StackString< MaxBufSize >	251
oscl_stat_buf	253
OSCL_String	254
OscI_Tag< Alloc >	260
OscI_Tag_Base	262
OscI_TagTree< T, Alloc >	264
OscI_TagTree< T, Alloc >::const_iterator	268
OscI_TagTree< T, Alloc >::iterator	271
OscI_TagTree< T, Alloc >::Node	274
OscI_TAlloc< T, Alloc >	276
OscI_TAlloc< T, Alloc >::rebind< U, V >	279
OscI_Vector< T, Alloc >	280
OscI_Vector_Base	285
OSCL_wFastString	289
OSCL_wHeapString< Alloc >	292
OSCL_wHeapStringA	294
OSCL_wStackString< MaxBufSize >	297

OSCL_wString	299
OsclAcceptMethod	303
OsclAcceptRequest	304
OsclActiveObject	305
OsclAllocDestructDealloc	309
OsclAOStatus	310
OsclAsyncFile	311
OsclAsyncFileBuffer	314
OsclAuditCB	316
OsclBindMethod	317
OsclBindRequest	318
OsclBinIStream	319
OsclBinIStreamBigEndian	321
OsclBinIStreamLittleEndian	324
OsclBinOStream (Class OsclBinOStream implements the basic stream functions for an output stream)	326
OsclBinOStreamBigEndian (Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering)	327
OsclBinOStreamLittleEndian (Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering)	329
OsclBinStream	331
OsclBuf	335
OsclCompareLess< T >	337
OsclComponentRegistry	338
OsclComponentRegistryData	340
OsclComponentRegistryElement	341
OsclConnectMethod	343
OsclConnectRequest	344
OsclDestructDealloc	345
OsclDNS	346
OsclDNSI	348
OsclDNSIBase	350
OsclDNSMethod	353
OsclDNSObserver	356
OsclDNSRequest	357
OsclDNSRequestAO	358
OsclDoubleLink	361
OsclDoubleList< T >	362
OsclDoubleListBase	363
OsclDoubleRunner< T >	365
OsclError	367
OsclErrorAllocator (This class provides static methods to invoke the user defined memory allocation routines)	369
OsclErrorTrap	371
OsclErrorTrapImp	372
OsclException< LeaveCode > (Oscl_exception.h contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from)	374
OsclExclusiveArrayPtr< T > (Template class that defines an array pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory)	375
OsclExclusivePtr< T > (Template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory)	378

OsclExclusivePtrA< T, Alloc > (Template class that defines any pointer like object intended to be assigned an address obtained (directly or or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory)	381
OsclExecScheduler	384
OsclExecSchedulerBase	386
OsclExecSchedulerCommonBase	387
OsclFileCache	396
OsclFileCacheBuffer	398
OsclFileHandle	400
OsclFileStats	401
OsclFileStatsItem	402
OsclGetHostNameMethod	403
OsclGetHostNameRequest	404
OsclInit	405
OsclInteger64Transport	406
OsclIPSocketI	407
OsclJump	410
OsclListenMethod	411
OsclListenRequest	412
OsclLockBase	413
OsclMem	414
OsclMemAllocator	415
OsclMemAllocDestructDealloc< T >	416
OsclMemAudit	418
OSCLMemAutoPtr< T, _Allocator > (The oscl_auto_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the oscl_auto_ptr expires, its destructor uses delete to free the memory)	424
OsclMemBasicAllocator	428
OsclMemBasicAllocDestructDealloc< T >	429
OsclMemGlobalAuditObject	430
OsclMemoryFragment	431
OsclMemPoolAllocator	432
OsclMemPoolFixedChunkAllocator	433
OsclMemPoolFixedChunkAllocatorObserver	437
OsclMemPoolResizableAllocator	438
OsclMemPoolResizableAllocator::MemPoolBlockInfo	444
OsclMemPoolResizableAllocator::MemPoolBufferInfo	445
OsclMemPoolResizableAllocatorMemoryObserver	446
OsclMemPoolResizableAllocatorObserver	447
OsclMemStatsNode	448
OsclMutex	449
OsclNameString< __len >	451
OsclNativeFile	452
OsclNativeFileParams	455
OsclNetworkAddress	456
OsclNullLock	457
OsclPriorityLink	458
OsclPriorityList< T >	459
OsclPriorityQueue< Qelem, Alloc, Container, Compare >	460
OsclPriorityQueueBase	464
OsclProcStatus	465
OsclPtr	467
OsclPtrC	469

OsclRand	471
OsclReadyAlloc	472
OsclReadyCompare	473
OsclReadyQ	474
OsclRecvFromMethod	476
OsclRecvFromRequest	478
OsclRecvMethod	480
OsclRecvRequest	481
OsclRefCounter	482
OsclRefCounterDA	484
OsclRefCounterMemFrag	486
OsclRefCounterMTDA< LockType >	488
OsclRefCounterMTSA< DeallocType, LockType >	490
OsclRefCounterSA< DeallocType >	492
OsclRegistryAccessClient	494
OsclRegistryAccessClientImpl	496
OsclRegistryAccessClientTlsImpl	497
OsclRegistryAccessElement	498
OsclRegistryClient	499
OsclRegistryClientImpl	501
OsclRegistryClientTlsImpl	503
OsclRegistryServTlsImpl	504
OsclScheduler	506
OsclSchedulerObserver	507
OsclScopedLock< LockClass > (Template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope)	508
OsclSelect	509
OsclSemaphore	511
OsclSendMethod	513
OsclSendRequest	514
OsclSendToMethod	515
OsclSendToRequest	516
OsclSharedPtr< TheClass > (A parameterized smart pointer class)	517
OsclShutdownMethod	520
OsclShutdownRequest	521
OsclSingleton< T, ID, Registry >	522
OsclSingletonRegistry	524
OsclSocketI	525
OsclSocketIBase	530
OsclSocketMethod	535
OsclSocketObserver	538
OsclSocketRequest	539
OsclSocketRequestAO	540
OsclSocketServ	544
OsclSocketServI	546
OsclSocketServIBase	548
OsclSocketServRequestList	550
OsclSocketServRequestQELEM	552
OsclTCPSocket	553
OsclTCPSocketI	559
OsclThread	562
OsclThreadLock	566
OsclTickCount	567

OsclTimer< Alloc >	569
OsclTimerCompare	572
OsclTimerObject	573
OsclTimerObserver	577
OsclTimerQ	578
OsclTLS< T, ID, Registry >	579
OsclTLSEx< T, ID, Registry >	581
OsclTLSRegistry	583
OsclTLSRegistryEx	584
OsclTrapItem	585
OsclTrapStack	586
OsclTrapStackItem	587
OsclUDPSocket	588
OsclUDPSocketI	593
OsclUuid	595
PVActiveBase	597
PVActiveStats	601
PVLogger	602
PVLoggerAppender	608
PVLoggerFilter	609
PVLoggerLayout	611
PVLoggerRegistry	613
PVSchedulerStopper	616
PVSockBufRecv	617
PVSockBufSend	618
PVThreadContext	619
RecvFromParam	621
RecvParam	623
SendParam	624
SendToParam	625
ShutdownParam	626
SocketRequestParam	627
StrCSumPtrLen (Same as StrPtrLen , but includes checksum field and method to speed up querying)	629
StrPtrLen (This data structure encapsulates a set of functions used to perform)	632
TimeValue (Time value in a format native to the system)	634
TLSStorageOps	640
TReadyQueLink	641
WStrPtrLen (This data structure encapsulates a set of functions used to perform)	642

Chapter 4

oscl File Index

4.1 oscl File List

Here is a list of all files with brief descriptions:

<code>oscl_aostatus.h</code> (Some basic types used with active objects)	644
<code>oscl_assert.h</code> (The file <code>oscl_assert.h</code> provides an OSCL_ASSERT macro to document assumptions and test them during development)	645
<code>oscl_base.h</code> (The file <code>oscl_base.h</code> is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros)	646
<code>oscl_base_alloc.h</code> (A basic allocator that does not rely on other modules)	647
<code>oscl_base_macros.h</code> (This file defines common macros and constants for basic compilation support)	648
<code>oscl_bin_stream.h</code> (Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order)	649
<code>oscl_byte_order.h</code> (This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders))	650
<code>oscl_defalloc.h</code> (The file defines simple default memory allocator classes. These allocators are used by the <code>Oscl_Vector</code> and <code>Oscl_Map</code> class, etc)	651
<code>oscl_dll.h</code> (Defines a DLL entry point)	652
<code>oscl_dns.h</code> (The file <code>oscl_socket.h</code> defines the OSCL DNS APIs)	653
<code>oscl_dns_gethostbyname.h</code>	654
<code>oscl_dns_imp.h</code>	655
<code>oscl_dns_imp_base.h</code>	656
<code>oscl_dns_imp_pv.h</code>	657
<code>oscl_dns_method.h</code>	658
<code>oscl_dns_param.h</code>	659
<code>oscl_dns_request.h</code>	660
<code>oscl_dns_tuneables.h</code>	661
<code>oscl_double_list.h</code> (Internal use types for scheduler)	662
<code>oscl_errno.h</code> (Defines functions to access additional information on errors where supported through an errno or similar service)	663
<code>oscl_error.h</code> (OSCL Error trap and cleanup include file)	664
<code>oscl_error_allocator.h</code> (Defines a memory allocation class used by the oscl error layer)	665
<code>oscl_error_codes.h</code> (Defines basic error and leave codes)	666
<code>oscl_error_imp.h</code> (Internal error implementation support)	667
<code>oscl_error_imp_cppexceptions.h</code> (Implementation File for Leave using C++ exceptions)	668
<code>oscl_error_imp_fatalerror.h</code> (Implementation File for Leave using system fatal error)	669
<code>oscl_error_imp_jumps.h</code> (Implemenation of using Setjmp / Longjmp)	670

<code>oscl_error_trapcleanup.h</code> (OSCL Error trap and cleanup implementation include file)	672
<code>oscl_exception.h</code> (Contains all the exception handling macros and classes)	673
<code>oscl_exclusive_ptr.h</code> (This file defines the <code>OsclExclusivePtr</code> template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error)	674
<code>oscl_file_async_read.h</code>	675
<code>oscl_file_cache.h</code> (The file <code>oscl_file_cache.h</code> defines the class <code>OsclFileCache</code>)	676
<code>oscl_file_dir_utils.h</code> (The file <code>oscl_file_dir_utils.h</code> defines some unix-style directory ops)	677
<code>oscl_file_find.h</code> (The file <code>oscl_file_find.h</code> defines the class <code>Oscl_FileFind</code>)	679
<code>oscl_file_handle.h</code> (The file <code>oscl_file_handle.h</code> defines the class <code>OsclFileHandle</code>)	680
<code>oscl_file_io.h</code> (The file <code>oscl_file_io.h</code> defines the class <code>Oscl_File</code> . This is the public API to the basic file I/O operations)	681
<code>oscl_file_native.h</code> (The file <code>oscl_file_native.h</code> defines the class <code>OsclNativeFile</code> . This is the porting layer for basic file I/O operations)	682
<code>oscl_file_server.h</code> (The file <code>oscl_file_server.h</code> defines the class <code>Oscl_FileServer</code> . This is the porting layer for file server implementations)	683
<code>oscl_file_stats.h</code> (File stats class)	684
<code>oscl_file_types.h</code> (The file <code>oscl_file_types.h</code> defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here)	685
<code>oscl_heapbase.h</code> (OSCL Heap Base include file)	686
<code>oscl_init.h</code> (Global oscl initialization)	687
<code>oscl_int64_utils.h</code>	688
<code>oscl_ip_socket.h</code>	689
<code>oscl_linked_list.h</code> (The file <code>oscl_linked_list.h</code> defines the template class <code>Oscl_Linked_List</code> which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter)	690
<code>oscl_lock_base.h</code> (This file defines an abstract lock class, <code>OsclLockBase</code> , that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, <code>OsclNullLock</code> , is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the <code>OsclScopedLock</code> class which is template class takes care of freeing the lock when the class goes out of scope)	691
<code>oscl_map.h</code> (The file <code>oscl_map.h</code> defines the template class <code>Oscl_Map</code> which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter)	692
<code>oscl_math.h</code> (Provides math functions)	693
<code>oscl_media_data.h</code> (Defines a container class for media data made up of a collection of memory fragments)	694
<code>oscl_media_status.h</code> (Defines a status values for the <code>MediaData</code> containers)	695
<code>oscl_mem.h</code> (This file contains basic memory definitions for common use across platforms)	696
<code>oscl_mem_align.h</code>	699
<code>oscl_mem_audit.h</code> (This file contains the definition and partial implementation of MM_Audit class)	700
<code>oscl_mem_audit_internals.h</code> (This file contains the internal definitions for the mem audit library)	702
<code>oscl_mem_auto_ptr.h</code> (This file defines the <code>oscl_mem_auto_ptr</code> template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error)	703
<code>oscl_mem_basic_functions.h</code> (This file contains prototypes for the basic memory functions)	704
<code>oscl_mem_inst.h</code> (The file defines default memory instrumentation level)	705
<code>oscl_mem_mempool.h</code> (This file contains the definition of memory pool allocators)	706
<code>oscl_mempool_allocator.h</code> (This file contains the definition of memory pool allocator for leave/trap)	707
<code>oscl_mutex.h</code> (This file provides implementation of mutex)	708
<code>oscl_namestring.h</code> (Name string class include file)	709

<code>oscl_opaque_type.h</code> (The file <code>oscl_opaque_type.h</code> defines pure virtual classes for working with opaque types)	710
<code>oscl_pqueue.h</code> (Implements a priority queue data structure similar to STL)	711
<code>oscl_proctstatus.h</code>	712
<code>oscl_queue.h</code> (The file <code>oscl_queue.h</code> defines the template class <code>Oscl_Queue</code> . It is similar to the <code>STL::queue</code> class, with some differences: - less complete - based on array rather than a deque - some interfaces modeled on <code>oscl_vector</code> , for ease of transition Memory allocation is abstracted through the use of an allocator template parameter)	713
<code>oscl_rand.h</code> (Provides pseudo-random number generation)	714
<code>oscl_refcounter.h</code> (A general purpose reference counter to object lifetimes)	715
<code>oscl_refcounter_memfrag.h</code> (This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount)	716
<code>oscl_registry_access_client.h</code> (Client-side implementation Registry Access implementation)	717
<code>oscl_registry_client.h</code> (Client-side implementation of <code>OsclRegistry</code>)	718
<code>oscl_registry_client_impl.h</code> (Client-side implementation of <code>OsclRegistryInterface</code>)	719
<code>oscl_registry_serv_impl.h</code> (Server-side implementation of <code>OsclRegistry</code> interfaces)	720
<code>oscl_registry_serv_impl_global.h</code>	721
<code>oscl_registry_serv_impl_tls.h</code>	722
<code>oscl_registry_types.h</code> (Common types used in <code>Oscl registry</code> interfaces)	723
<code>oscl_scheduler.h</code>	724
<code>oscl_scheduler_ao.h</code> (<code>Oscl Scheduler</code> user execution object classes)	725
<code>oscl_scheduler_aobase.h</code> (<code>Oscl Scheduler</code> internal active object classes)	726
<code>oscl_scheduler_readyq.h</code> (Ready q types for <code>oscl scheduler</code>)	727
<code>oscl_scheduler_threadcontext.h</code> (Thread context functions needed by <code>oscl scheduler</code>)	728
<code>oscl_scheduler_tuneables.h</code> (Tunable settings for <code>Oscl Scheduler</code>)	729
<code>oscl_scheduler_types.h</code> (Scheduler common types include file)	730
<code>oscl_semaphore.h</code> (This file provides implementation of mutex)	731
<code>oscl_shared_ptr.h</code> (This file defines a template class <code>OsclSharedPtr</code> which is a "smart pointer" to the parameterized type)	732
<code>oscl_singleton.h</code> (This file defines the <code>OsclSingleton</code> class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time)	733
<code>oscl_snprintf.h</code> (Provides a portable implementation of <code>snprintf</code>)	735
<code>oscl_socket.h</code> (The file <code>oscl_socket.h</code> defines the OSCL Socket APIs)	736
<code>oscl_socket_accept.h</code>	737
<code>oscl_socket_bind.h</code>	738
<code>oscl_socket_connect.h</code>	739
<code>oscl_socket_imp.h</code>	740
<code>oscl_socket_imp_base.h</code>	741
<code>oscl_socket_imp_pv.h</code>	742
<code>oscl_socket_listen.h</code>	743
<code>oscl_socket_method.h</code>	744
<code>oscl_socket_recv.h</code>	745
<code>oscl_socket_recv_from.h</code>	746
<code>oscl_socket_request.h</code>	747
<code>oscl_socket_send.h</code>	748
<code>oscl_socket_send_to.h</code>	749
<code>oscl_socket_serv_imp.h</code>	750
<code>oscl_socket_serv_imp_base.h</code>	751
<code>oscl_socket_serv_imp_pv.h</code>	752
<code>oscl_socket_serv_imp_reqlist.h</code>	753
<code>oscl_socket_shutdown.h</code>	754

oscl_socket_stats.h	755
oscl_socket_tunables.h	757
oscl_socket_types.h	759
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)	761
oscl_str_ptr_len.h (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally)	763
oscl_string.h (Provides a standardized set of string containers that can be used in place of character arrays)	764
oscl_string_containers.h (Provides a standardized set of string containers that can be used in place of character arrays)	765
oscl_string_rep.h (Contains some internal implementation for string containers)	766
oscl_string_uri.h (Utilities to unescape URIs)	767
oscl_string_utf8.h (Utilities to validate and truncate UTF-8 encoded strings)	768
oscl_string_utils.h (Utilities to parse and convert strings)	769
oscl_string_xml.h (Utilities to escape special characters in XML strings)	770
oscl_tagtree.h (The file oscl_tagtree.h ..)	771
oscl_tcp_socket.h	772
oscl_thread.h	773
oscl_tickcount.h (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally)	775
oscl_time.h (The file oscl_time.h defines two classes NTPTime and TimeValue for getting, manipulating, and formatting time values. The TimeValue class is based on the native system time format while NTPTime is used for the standard Network Time Protocol format)	776
oscl_timer.h	778
oscl_tls.h	779
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)	780
oscl_types.h (This file contains basic type definitions for common use across platforms)	781
oscl_udp_socket.h	782
oscl_utf8conv.h (Utilities to convert unicode to utf8 and vice versa)	783
oscl_uuid.h (This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32)	784
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)	785
osclconfig.h (This file contains configuration information for the linux platform)	786
osclconfig_ansi_memory.h (This file contains common typedefs based on the ANSI C limits.h header)	788
osclconfig_check.h	789
osclconfig_compiler_warnings.h (This file contains the ability to turn off/on compiler warnings)	790
osclconfig_error.h (This file contains the common typedefs and header files needed to compile osclerror)	791
osclconfig_error_check.h	792
osclconfig_global_new_delete.h	793
osclconfig_global_placement_new.h	794
osclconfig_io.h (This file contains common typedefs based on the ANSI C limits.h header)	795
osclconfig_io_check.h	802
osclconfig_ix86.h (This file contains configuration information for the ix86 processor family)	803

oslconfig_lib.h (This file contains configuration information for the ANSI build)	804
oslconfig_lib_check.h	805
oslconfig_limits_typedefs.h (This file contains common typedefs based on the ANSI C limits.h header)	806
oslconfig_memory.h	807
oslconfig_memory_check.h	808
oslconfig_no_os.h	809
oslconfig_proc.h (This file contains configuration information for the linux platform)	810
oslconfig_proc_check.h	811
oslconfig_proc_unix_android.h	813
oslconfig_proc_unix_common.h	815
oslconfig_time.h	817
oslconfig_time_check.h	818
oslconfig_unix_android.h	819
oslconfig_unix_common.h	823
oslconfig_util.h	827
oslconfig_util_check.h	828
pvlogger.h (This file contains basic logger interfaces for common use across platforms)	829
pvlogger_accessories.h	837
pvlogger_c.h (This file contains basic logger interfaces for common use across platforms. C-callable version)	838
pvlogger_registry.h	840

Chapter 5

oscl Page Index

5.1 oscl Related Pages

Here is a list of all related documentation pages:

Todo List	841
---------------------	-----

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_SYMBIAN_ERRORTRAP 0
- #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0
- #define OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS 0
- #define OSCL_HAS_PV_C_OS_TIME_FUNCS 0
- #define OSCL_HAS_UNIX_TIME_FUNCS 0
- #define OSCL_HAS_SYMBIAN_TIMERS 0
- #define OSCL_HAS_SYMBIAN_MATH 0
- #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0
- #define OSCL_HAS_PTHREAD_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0
- #define OSCL_HAS_SAVAJE_IO_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0
- #define OSCL_HAS_SYMBIAN_DNS_SERVER 0
- #define OSCL_HAS_BERKELEY_SOCKETS 0

Typedefs

- `typedef int8 __int8_check__`
- `typedef uint8 __uint8_check__`
- `typedef int16 __int16_check__`
- `typedef uint16 __uint16_check__`
- `typedef int32 __int32_check__`
- `typedef uint32 __uint32_check__`

6.1.1 Define Documentation

6.1.1.1 `#define OSCL_ASSERT_ALWAYS 0`

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

6.1.1.2 `#define OSCL_BYTE_ORDER_BIG_ENDIAN 0`

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

6.1.1.3 `#define OSCL_BYTE_ORDER_LITTLE_ENDIAN 1`

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

6.1.1.4 `#define OSCL_HAS_BERKELEY_SOCKETS 0`

6.1.1.5 `#define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0`

6.1.1.6 `#define OSCL_HAS_MSWIN_SUPPORT 0`

6.1.1.7 `#define OSCL_HAS_PRAGMA_PACK 0`

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

- 6.1.1.8 #define OSCL_HAS_PTHREAD_SUPPORT 0
- 6.1.1.9 #define OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS 0
- 6.1.1.10 #define OSCL_HAS_PV_C_OS_SUPPORT 0
- 6.1.1.11 #define OSCL_HAS_PV_C_OS_TIME_FUNCS 0
- 6.1.1.12 #define OSCL_HAS_SAVAJE_IO_SUPPORT 0
- 6.1.1.13 #define OSCL_HAS_SAVAJE_SUPPORT 0
- 6.1.1.14 #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0
- 6.1.1.15 #define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0
- 6.1.1.16 #define OSCL_HAS_SYMBIAN_DNS_SERVER 0
- 6.1.1.17 #define OSCL_HAS_SYMBIAN_ERRORTRAP 0
- 6.1.1.18 #define OSCL_HAS_SYMBIAN_MATH 0
- 6.1.1.19 #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0
- 6.1.1.20 #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- 6.1.1.21 #define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0
- 6.1.1.22 #define OSCL_HAS_SYMBIAN_SUPPORT 0
- 6.1.1.23 #define OSCL_HAS_SYMBIAN_TIMERS 0
- 6.1.1.24 #define OSCL_HAS_UNIX_SUPPORT 0
- 6.1.1.25 #define OSCL_HAS_UNIX_TIME_FUNCS 0
- 6.1.1.26 #define OSCL_INTEGERS_WORD_ALIGNED 1

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

6.1.2 Typedef Documentation

- 6.1.2.1 `typedef int16 __int16__check__`
- 6.1.2.2 `typedef int32 __int32__check__`
- 6.1.2.3 `typedef int8 __int8__check__`
- 6.1.2.4 `typedef uint16 __uint16__check__`
- 6.1.2.5 `typedef uint32 __uint32__check__`
- 6.1.2.6 `typedef uint8 __uint8__check__`

6.2 OSCL Base

Files

- file [oscl_assert.h](#)

The file [oscl_assert.h](#) provides an OSCL_ASSERT macro to document assumptions and test them during development.

- file [oscl_base.h](#)

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

- file [oscl_base_alloc.h](#)

A basic allocator that does not rely on other modules.

- file [oscl_base_macros.h](#)

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

- file [oscl_byte_order.h](#)

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

- file [oscl_defalloc.h](#)

The file defines simple default memory allocator classes. These allocators are used by the [Oscl_Vector](#) and [Oscl_Map](#) class, etc.

- file [oscl_dll.h](#)

Defines a DLL entry point.

- file [oscl_exclusive_ptr.h](#)

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

- file [oscl_linked_list.h](#)

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

- file [oscl_lock_base.h](#)

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

- file [oscl_map.h](#)

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

- file [oscl_mem_inst.h](#)

The file defines default memory instrumentation level.

- file [oscl_opaque_type.h](#)

The file [oscl_opaque_type.h](#) defines pure virtual classes for working with opaque types.

- file [oscl_queue.h](#)

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

- file [oscl_refcounter.h](#)

A general purpose reference counter to object lifetimes.

- file [oscl_refcounter_memfrag.h](#)

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

- file [oscl_shared_ptr.h](#)

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

- file [oscl_stdstring.h](#)

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

- file [oscl_tagtree.h](#)

The file [oscl_tagtree.h](#) ...

- file [oscl_time.h](#)

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

- file [oscl_tree.h](#)

The file [oscl_tree.h](#) defines the template class [Oscl_Rb_Tree](#) which has a very similar API as the `STL Tree` class. It is an implementation of a Red-Black Tree for use by the [Oscl_Map](#) class. Memory allocation is abstracted through the use of an allocator template parameter.

- file [oscl_types.h](#)

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

- file [oscl_vector.h](#)

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

Data Structures

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

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

- class [Oscl_Alloc](#)
- class [Oscl_Dealloc](#)
- class [Oscl_DefAlloc](#)
- class [Oscl_DefAllocWithRefCounter](#)
- struct [Oscl_Less](#)
- class [Oscl_Linked_List](#)
- class [Oscl_Linked_List_Base](#)
- class [Oscl_Map](#)
- class [Oscl_MTLinked_List](#)
- class [Oscl_Opaque_Type_Alloc](#)
- class [Oscl_Opaque_Type_Alloc_LL](#)
- class [Oscl_Opaque_Type_Compare](#)
- struct [Oscl_Pair](#)
- class [Oscl_Queue](#)
- class [Oscl_Queue_Base](#)
- class [Oscl_Rb_Tree](#)
- class [Oscl_Rb_Tree_Base](#)
- struct [Oscl_Rb_Tree_Const_Iterator](#)
- struct [Oscl_Rb_Tree_Iterator](#)
- struct [Oscl_Rb_Tree_Node](#)
- struct [Oscl_Rb_Tree_Node_Base](#)
- struct [Oscl_Select1st](#)
- struct [Oscl_Tag](#)
- struct [Oscl_Tag_Base](#)
- class [Oscl_TagTree](#)
- class [Oscl_TAlloc](#)
- class [Oscl_Vector](#)
- class [Oscl_Vector_Base](#)
- class [OsclAllocDestructDealloc](#)
- class [OsclDestructDealloc](#)
- class [OsclExclusiveArrayPtr](#)

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

- class [OsclExclusivePtr](#)

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

- class [OsclExclusivePtrA](#)

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

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

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

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

- class [OsclSharedPtr](#)

A parameterized smart pointer class.

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

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

- class [TLSStorageOps](#)

Defines

- #define [OSCL_ASSERT](#)(*_expr*) ((*_expr*)?((void)0):OSCL Assert(# *_expr*, __FILE__, __LINE__))
- #define [OSCL_HAS_SINGLETON_SUPPORT](#) 1
- #define [NULL_TERM_CHAR](#) '\0'

The NULL_TERM_CHAR is used to terminate c-style strings.

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

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

- #define [OSCL_INLINE](#) inline
- #define [OSCL_COND_EXPORT_REF](#)
- #define [OSCL_COND_IMPORT_REF](#)
- #define [OSCL_CONST_CAST](#)(*type*, *exp*) ((*type*)(*exp*))

Type casting macros.

- #define [OSCL_STATIC_CAST](#)(*type*, *exp*) ((*type*)(*exp*))
- #define [OSCL_REINTERPRET_CAST](#)(*type*, *exp*) ((*type*)(*exp*))
- #define [OSCL_DYNAMIC_CAST](#)(*type*, *exp*) ((*type*)(*exp*))
- #define [OSCL_UNUSED_ARG](#)(*vbl*) (void)(*vbl*)
- #define [OSCL_UNUSED_RETURN](#)(*value*) return *value*
- #define [OSCL_MIN](#)(*a*, *b*) ((*a*) < (*b*) ? (*a*) : (*b*))
- #define [OSCL_MAX](#)(*a*, *b*) ((*a*) > (*b*) ? (*a*) : (*b*))
- #define [OSCL_ABS](#)(*a*) ((*a*) > (0) ? (*a*) : -(*a*))
- #define [OSCL_TEMPLATED_DESTRUCTOR_CALL](#)(*type*, *simple_type*) *type* :: ~*simple_type* ()
- #define [OSCL_UNSIGNED_CONST](#)(*x*) *x*
- #define [OSCL_PACKED_VAR](#) "error"
- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [ALLOCATE](#)(*n*) allocate_fl(*n*, __FILE__, __LINE__)
- #define [ALLOC_AND_CONSTRUCT](#)(*n*) alloc_and_construct_fl(*n*, __FILE__, __LINE__)
- #define [OSCL_DLL_ENTRY_POINT](#)() void oscl_dll_entry_point() {}
- #define [OSCL_DLL_ENTRY_POINT_DEFAULT](#)()

- #define PVMEM_INST_LEVEL 1
- #define OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT
- #define OSCL_TLS_BASE_SLOTS OSCL_TLS_ID_BASE_LAST +1
- #define OSCL_TLS_EXTERNAL_SLOTS 0
- #define OSCL_TLS_MAX_SLOTS (OSCL_TLS_BASE_SLOTS + OSCL_TLS_EXTERNAL_SLOTS)

TypeDefs

- typedef char CtimeStrBuf [CTIME_BUFFER_SIZE]
- typedef char PV8601timeStrBuf [PV8601TIME_BUFFER_SIZE]
- typedef OsclAny TOsclTlsKey
- typedef int c_bool

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

- typedef void OsclAny

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

- typedef char mbchar

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

- typedef unsigned int uint

The uint type is a convenient abbreviation for unsigned int.

- typedef uint8 octet

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

- typedef float OsclFloat

The Float type defined as OsclFloat.

- typedef OSCL_NATIVE_INT64_TYPE int64

- typedef OSCL_NATIVE_UINT64_TYPE uint64

- typedef OSCL_NATIVE_WCHAR_TYPE oscl_wchar

- typedef oscl_wchar OSCL_TCHAR

define OSCL_TCHAR

Enumerations

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

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

Functions

- OSCL_COND_IMPORT_REF void [_OSCL_Abort\(\)](#)
This function terminates the current process abnormally.
- OSCL_IMPORT_REF void [OSCL_Assert](#) (const char *expr, const char *filename, int line_number)
OSCL_ASSERT macro evaluates an expression and when the result is false, prints a diagnostic message and aborts the program.
- void [PVOsclBase_Init\(\)](#)
- void [PVOsclBase_Cleanup\(\)](#)
- void [little_endian_to_host](#) (char *data, uint32 size)
Convert little endian to host format.
- void [host_to_little_endian](#) (char *data, unsigned int size)
Convert host to little endian format.
- void [big_endian_to_host](#) (char *data, unsigned int size)
Convert big endian to host format.
- void [host_to_big_endian](#) (char *data, unsigned int size)
Convert host to big endian format.
- OSCL_IMPORT_REF uint32 [oscl_strlen](#) (const char *str)
- OSCL_IMPORT_REF uint32 [oscl_strlen](#) (const [oscl_wchar](#) *str)
- OSCL_IMPORT_REF char * [oscl_strncpy](#) (char *dest, const char *src, uint32 count)
- OSCL_IMPORT_REF [oscl_wchar](#) * [oscl_strncpy](#) ([oscl_wchar](#) *dest, const [oscl_wchar](#) *src, uint32 count)
- OSCL_IMPORT_REF int32 [oscl_strcmp](#) (const char *str1, const char *str2)
- OSCL_IMPORT_REF int32 [oscl_strcmp](#) (const [oscl_wchar](#) *str1, const [oscl_wchar](#) *str2)
- OSCL_IMPORT_REF int32 [oscl_strncmp](#) (const char *str1, const char *str2, uint32 count)
- OSCL_IMPORT_REF int32 [oscl_strncmp](#) (const [oscl_wchar](#) *str1, const [oscl_wchar](#) *str2, uint32 count)
- OSCL_IMPORT_REF char * [oscl_strncat](#) (char *dest, const char *src, uint32 count)
- OSCL_IMPORT_REF [oscl_wchar](#) * [oscl_strncat](#) ([oscl_wchar](#) *dest, const [oscl_wchar](#) *src, uint32 count)
- OSCL_IMPORT_REF const char * [oscl_strchr](#) (const char *str, int32 c)
- OSCL_IMPORT_REF char * [oscl_strchr](#) (char *str, int32 c)
- OSCL_IMPORT_REF const [oscl_wchar](#) * [oscl_strchr](#) (const [oscl_wchar](#) *str, int32 c)
- OSCL_IMPORT_REF [oscl_wchar](#) * [oscl_strchr](#) ([oscl_wchar](#) *str, int32 c)
- OSCL_IMPORT_REF const char * [oscl strrchr](#) (const char *str, int32 c)
- OSCL_IMPORT_REF char * [oscl strrchr](#) (char *str, int32 c)
- OSCL_IMPORT_REF const [oscl_wchar](#) * [oscl strrchr](#) (const [oscl_wchar](#) *str, int32 c)
- OSCL_IMPORT_REF [oscl_wchar](#) * [oscl strrchr](#) ([oscl_wchar](#) *str, int32 c)
- OSCL_IMPORT_REF char * [oscl_strset](#) (char *dest, char val, uint32 count)
- OSCL_IMPORT_REF [oscl_wchar](#) * [oscl_strset](#) ([oscl_wchar](#) *dest, [oscl_wchar](#) val, uint32 count)
- OSCL_IMPORT_REF int32 [oscl_CIstrcmp](#) (const char *str1, const char *str2)
- OSCL_IMPORT_REF int32 [oscl_CIstrcmp](#) (const [oscl_wchar](#) *str1, const [oscl_wchar](#) *str2)
- OSCL_IMPORT_REF int32 [oscl_CIstrncmp](#) (const char *str1, const char *str2, uint32 count)

- OSCL_IMPORT_REF int32 `oscl_Clstrncmp` (const `oscl_wchar` *str1, const `oscl_wchar` *str2, uint32 count)
- OSCL_IMPORT_REF char `oscl_tolower` (const char car)
- OSCL_IMPORT_REF `oscl_wchar oscl_tolower` (const `oscl_wchar` car)
- OSCL_IMPORT_REF bool `oscl_isLetter` (const char car)
- OSCL_IMPORT_REF const char * `oscl_strstr` (const char *str1, const char *str2)
- OSCL_IMPORT_REF char * `oscl_strstr` (char *str1, const char *str2)
- OSCL_IMPORT_REF const `oscl_wchar` * `oscl_strstr` (const `oscl_wchar` *str1, const `oscl_wchar` *str2)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_strstr` (`oscl_wchar` *str1, const `oscl_wchar` *str2)
- OSCL_IMPORT_REF char * `oscl_streat` (char *dest, const char *src)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_streat` (`oscl_wchar` *dest, const `oscl_wchar` *src)
- OSCL_IMPORT_REF void `PV8601ToRFC822` (`PV8601timeStrBuf` pv8601_buffer, `CtimeStrBuf` ctime_buffer)
- OSCL_IMPORT_REF void `RFC822ToPV8601` (`CtimeStrBuf` ctime_buffer, `PV8601timeStrBuf`)
- OSCL_COND_IMPORT_REF `TimeValue operator-` (const `TimeValue` &a, const `TimeValue` &b)
- bool `operator==` (const `OsclSharedPtr` &b) const

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

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

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

- void `Bind` (TheClass *ptr, `OsclRefCounter` *in_refcnt)

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

Variables

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

6.2.1 Detailed Description

Additional osclbase comment

Additional osclbase comment

Additional osclbase comment

6.2.2 Define Documentation

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

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

6.2.2.3 #define NULL (0)

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

6.2.2.4 #define NULL_TERM_CHAR '\0'

The NULL_TERM_CHAR is used to terminate c-style strings.

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

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

6.2.2.7 #define OSCL_COND_EXPORT_REF

6.2.2.8 #define OSCL_COND_IMPORT_REF

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

Type casting macros.

Parameters:

type Destination type of cast

exp Expression to cast

6.2.2.10 #define OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT

6.2.2.11 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

6.2.2.12 #define OSCL_DLL_ENTRY_POINT() void oscl_dll_entry_point() {}

DLL entry/exit point.

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

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

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

Usage :

```
LocalDliEntry() { custom operations... }  
LocalDliExit() { custom operations... }  
OSCL_DLL_ENTRY_POINT()
```

6.2.2.13 #define OSCL_DLL_ENTRY_POINT_DEFAULT()

Default DLL entry/exit point function.

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

Usage :

```
OSCL_DLL_ENTRY_POINT_DEFAULT()
```

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

6.2.2.15 #define OSCL_HAS_SINGLETON_SUPPORT 1

6.2.2.16 #define OSCL_INLINE inline

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

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

6.2.2.19 #define OSCL_PACKED_VAR "error"

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

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

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

6.2.2.23 #define OSCL_TLS_BASE_SLOTS OSCL_TLS_ID_BASE_LAST +1

6.2.2.24 #define OSCL_TLS_EXTERNAL_SLOTS 0

6.2.2.25 #define OSCL_TLS_MAX_SLOTS (OSCL_TLS_BASE_SLOTS + OSCL_TLS_EXTERNAL_SLOTS)

6.2.2.26 #define OSCL_UNSIGNED_CONST(x) x

6.2.2.27 #define OSCL_UNUSED_ARG(vbl) (void)(vbl)

The following two macros are used to avoid compiler warnings.

OSCL_UNUSED_ARG(vbl) is used to "reference" an otherwise unused parameter or variable, often one which is used only in an OSCL_ASSERT and thus unreferenced in release mode **OSCL_UNUSED_RETURN(val)** provides a "return" of a value, in places which will not actually be executed, such as after an OSCL_LEAVE or Thread::exit or abort. The value needs to be of an appropriate type for the current

function, though zero will usually suffice. Note that OSCL_UNUSED_RETURN will not be necessary for 'void' functions, as there is no requirement for a value-return operation.

6.2.2.28 #define OSCL_UNUSED_RETURN(value) return value

6.2.2.29 #define 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 CtimeStrBuff[CTIME_BUFFER_SIZE]

6.2.3.3 typedef OSCL_NATIVE_INT64_TYPE int64

6.2.3.4 typedef char mbchar

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

6.2.3.5 typedef uint8 octet

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

6.2.3.6 typedef oscl_wchar OSCL_TCHAR

define OSCL_TCHAR

6.2.3.7 typedef OSCL_NATIVE_WCHAR_TYPE oscl_wchar

6.2.3.8 typedef void OsclAny

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

6.2.3.9 typedef float OsclFloat

The Float type defined as OsclFloat.

6.2.3.10 typedef char PV8601timeStrBuf[PV8601TIME_BUFFER_SIZE]

6.2.3.11 typedef OsclAny TOsclTlsKey

6.2.3.12 typedef unsigned int uint

The uint type is a convenient abbreviation for unsigned int.

6.2.3.13 `typedef OSCL_NATIVE_UINT64_TYPE uint64`

6.2.4 Enumeration Type Documentation

6.2.4.1 `enum TimeUnit`s

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

Enumeration values:

`SECONDS`

`MILLISECONDS`

`MICROSECONDS`

6.2.5 Function Documentation

6.2.5.1 `OSCL_COND_IMPORT_REF void _OSCL_Abort ()`

This function terminates the current process abnormally.

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

Convert big endian to host format.

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

Parameters:

`data` A pointer to the input/output buffer

`size` The number of bytes in the buffer.

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

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

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

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

6.2.5.5 `void host_to_big_endian (char * data, unsigned int size)`

Convert host to big endian format.

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

Parameters:

`data` A pointer to the input/output buffer

`size` The number of bytes in the buffer.

6.2.5.6 void host_to_little_endian (char * *data*, unsigned int *size*)

Convert host to little endian format.

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

Parameters:

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

6.2.5.7 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.

6.2.5.8 OSCL_COND_IMPORT_REF TimeValue operator- (const TimeValue & *a*, const TimeValue & *b*)

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

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

6.2.5.10 OSCL_IMPORT_REF void OSCL_Assert (const char * *expr*, const char * *filename*, int *line_number*)

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

Parameters:

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

6.2.5.11 OSCL_IMPORT_REF int32 oscl_CIstrcmp (const oscl_wchar * *str1*, const oscl_wchar * *str2*)

Case in-sensitive string comparision.

Parameters:

- str1* string to compare

str2 string to compare

Returns:

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

6.2.5.12 OSCL_IMPORT_REF int32 oscl_CIstrcmp (const char * str1, const char * str2)

Case in-sensitive string comparision.

Parameters:

str1 string to compare

str2 string to compare

Returns:

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

6.2.5.13 OSCL_IMPORT_REF int32 oscl_CIstrncmp (const oscl_wchar * str1, const oscl_wchar * str2, uint32 count)

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

Parameters:

str1 string to compare

str2 string to compare

count Number of characters to compare

Returns:

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

6.2.5.14 OSCL_IMPORT_REF int32 oscl_CIstrncmp (const char * str1, const char * str2, uint32 count)

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

Parameters:

str1 string to compare

str2 string to compare

count Number of characters to compare

Returns:

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

6.2.5.15 OSCL_IMPORT_REF bool oscl_isLetter (const char *car*)

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

Parameters:*car***Returns:**

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

6.2.5.16 OSCL_IMPORT_REF oscl_wchar* oscl_streat (oscl_wchar * *dest*, const oscl_wchar * *src*)

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

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

dest

6.2.5.17 OSCL_IMPORT_REF char* oscl_streat (char * *dest*, const char * *src*)

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

Parameters:*dest* null terminated destination string*src* source string**Returns:**

dest

6.2.5.18 OSCL_IMPORT_REF oscl_wchar* oscl_strchr (oscl_wchar * *str*, int32 *c*)**6.2.5.19 OSCL_IMPORT_REF const oscl_wchar* oscl_strchr (const oscl_wchar * *str*, int32 *c*)**

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

Parameters:*str* null terminated source string*c* character to search for**Returns:**

6.2.5.20 OSCL_IMPORT_REF char* oscl_strchr (char * str, int32 c)**6.2.5.21 OSCL_IMPORT_REF const char* oscl_strchr (const char * str, int32 c)**

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

Parameters:

str null terminated source string

c character to search for

Returns:**6.2.5.22 OSCL_IMPORT_REF int32 oscl_strcmp (const oscl_wchar * str1, const oscl_wchar * str2)**

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

Parameters:

str1 String to compare

str2 String to compare

Returns:

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

6.2.5.23 OSCL_IMPORT_REF int32 oscl_strcmp (const char * str1, const char * str2)

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

Parameters:

str1 String to compare

str2 String to compare

Returns:

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

6.2.5.24 OSCL_IMPORT_REF uint32 oscl_strlen (const oscl_wchar * str)

Gets the length of a wide char string

Parameters:

str NULL terminated string.

Returns:

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

6.2.5.25 OSCL_IMPORT_REF uint32 oscl_strlen (const char * str)

Gets the length of a string

Parameters:

str NULL terminated string.

Returns:

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

6.2.5.26 OSCL_IMPORT_REF oscl_wchar* oscl_strncat (oscl_wchar * dest, const oscl_wchar * src, uint32 count)

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

Parameters:

dest null terminated destination string

src source string

count number of characters to append.

Returns:

dest

6.2.5.27 OSCL_IMPORT_REF char* oscl_strncat (char * dest, const char * src, uint32 count)

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

Parameters:

dest null terminated destination string

src source string

count number of characters to append.

Returns:

dest

6.2.5.28 OSCL_IMPORT_REF int32 oscl_strcmp (const oscl_wchar * str1, const oscl_wchar * str2, uint32 count)

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

Parameters:

str1 String to compare

str2 String to compare

count Number of characters to compare

Returns:

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

6.2.5.29 OSCL_IMPORT_REF int32 oscl_strncmp (const char * *str1*, const char * *str2*, uint32 *count*)

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

Parameters:

str1 String to compare

str2 String to compare

count Number of characters to compare

Returns:

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

6.2.5.30 OSCL_IMPORT_REF oscl_wchar* oscl_strncpy (oscl_wchar * *dest*, const oscl_wchar * *src*, uint32 *count*)

Copies the chars of one string to another.

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

Parameters:

dest Destination string

src NULL terminated source string

count Number of chars to copy

Returns:

Returns dest.

6.2.5.31 OSCL_IMPORT_REF char* oscl_strncpy (char * *dest*, const char * *src*, uint32 *count*)

Copies the chars of one string to another.

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

Parameters:

dest Destination string

src NULL terminated source string

count Number of chars to copy

Returns:

Returns dest.

6.2.5.32 OSCL_IMPORT_REF oscl_wchar* oscl_strrchr (oscl_wchar *str, int32 c)

6.2.5.33 OSCL_IMPORT_REF const oscl_wchar* oscl_strrchr (const oscl_wchar *str, int32 c)

6.2.5.34 OSCL_IMPORT_REF char* oscl_strrchr (char *str, int32 c)

6.2.5.35 OSCL_IMPORT_REF const char* oscl_strrchr (const char *str, int32 c)

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

Parameters:

str null terminated source string

c character to search for

Returns:

6.2.5.36 OSCL_IMPORT_REF oscl_wchar* oscl_strset (oscl_wchar *dest, oscl_wchar val, uint32 count)

Sets the characters of a string to a specified character

Parameters:

dest buffer to modify

val character to set

count number of chars to set

Returns:

the value of dest

6.2.5.37 OSCL_IMPORT_REF char* oscl_strset (char *dest, char val, uint32 count)

Sets the characters of a string to a specified character

Parameters:

dest buffer to modify

val character to set

count number of chars to set

Returns:

the value of dest

6.2.5.38 OSCL_IMPORT_REF oscl_wchar* oscl_strstr (oscl_wchar * str1, const oscl_wchar * str2)

6.2.5.39 OSCL_IMPORT_REF const oscl_wchar* oscl_strstr (const oscl_wchar * str1, const oscl_wchar * str2)

find the occurrence of sub-string in a string.

Parameters:

str1 string.

str2 sub-string

Returns:

pointer to the begining of sub-string.

6.2.5.40 OSCL_IMPORT_REF char* oscl_strstr (char * str1, const char * str2)

6.2.5.41 OSCL_IMPORT_REF const char* oscl_strstr (const char * str1, const char * str2)

find the occurrence of sub-string in a string.

Parameters:

str1 string.

str2 sub-string

Returns:

pointer to the begining of sub-string.

6.2.5.42 OSCL_IMPORT_REF oscl_wchar oscl_tolower (const oscl_wchar car)

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

Parameters:

car upper case character.

Returns:

lower case character.

6.2.5.43 OSCL_IMPORT_REF char oscl_tolower (const char car)

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

Parameters:

car upper case character.

Returns:

lower case character.

**6.2.5.44 OSCL_IMPORT_REF void PV8601ToRFC822 ([PV8601timeStrBuf](#) *pv8601_buffer*,
[CtimeStrBuf](#) *ctime_buffer*)**

6.2.5.45 void PVOsclBase_Cleanup ()

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

6.2.5.46 void PVOsclBase_Init ()

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

Exceptions:

leaves if out-of-memory

6.2.5.47 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 long **MSEC_PER_SEC** = 1000

6.2.6.3 const uint32 **OSCL_TLS_ID_BASE_LAST** = 11

6.2.6.4 const uint32 **OSCL_TLS_ID_ERRORHOOK** = 1

6.2.6.5 const uint32 **OSCL_TLS_ID_MAGICNUM** = 0

6.2.6.6 const uint32 **OSCL_TLS_ID_OSCLREGISTRY** = 10

6.2.6.7 const uint32 **OSCL_TLS_ID_PAYLOADPARSER** = 7

6.2.6.8 const uint32 **OSCL_TLS_ID_PVERRORTRAP** = 5

6.2.6.9 const uint32 **OSCL_TLS_ID_PVLOGGER** = 2

6.2.6.10 const uint32 **OSCL_TLS_ID_PVMFRECOGNIZER** = 8

6.2.6.11 const uint32 **OSCL_TLS_ID_PVSCHEDULER** = 4

6.2.6.12 const uint32 **OSCL_TLS_ID_SDPMEDIAPARSER** = 6

6.2.6.13 const uint32 **OSCL_TLS_ID_SQLITE3** = 11

6.2.6.14 const uint32 **OSCL_TLS_ID_TEST** = 3

6.2.6.15 const uint32 **OSCL_TLS_ID_WMDRM** = 9

6.2.6.16 const int **PV8601TIME_BUFFER_SIZE** = 21

6.2.6.17 const uint32 **unix_ntp_offset** = 2208988800U

6.2.6.18 const long **USEC_PER_SEC** = 1000000

6.3 OSCL Memory

Files

- file [oscl_mem.h](#)

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

- file [oscl_mem_audit.h](#)

This file contains the definition and partial implementation of MM_Audit class.

- file [oscl_mem_audit_internals.h](#)

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

- file [oscl_mem_auto_ptr.h](#)

This file defines the oscl_mem_auto_ptr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

- file [oscl_mem_basic_functions.h](#)

This file contains prototypes for the basic memory functions.

- file [oscl_mem_mempool.h](#)

This file contains the definition of memory pool allocators.

Data Structures

- class [allocator](#)
- class [allocator](#)
- class [HeapBase](#)
- struct [MM_AllocBlockFence](#)
- struct [MM_AllocBlockHdr](#)
- struct [MM_AllocInfo](#)
- struct [MM_AllocNode](#)
- struct [MM_AllocQueryInfo](#)
- class [MM_Audit_Imp](#)
- struct [MM_AuditOverheadStats](#)
- struct [MM_FailInsertParam](#)
- struct [MM_Stats_CB](#)
- struct [MM_Stats_t](#)
- class [OsclAuditCB](#)
- class [OsclMem](#)
- class [OsclMemAllocator](#)
- class [OsclMemAllocator](#)
- class [OsclMemAllocDestructDealloc](#)
- class [OsclMemAllocDestructDealloc](#)
- class [OsclMemAudit](#)
- class [OSCLMemAutoPtr](#)

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

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

Defines

- #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE
- #define OSCL_HAS_GLOBAL_NEW_DELETE 1
- #define OSCL_CLEANUP_BASE_CLASS(T) _OSCL_CLEANUP_BASE_CLASS(T)
- #define OSCL_ALLOC_NEW(T_allocator, T, params) new(T_allocator.allocate(1)) T params
- #define OSCL_TRAP_ALLOC_NEW(T_ptr, T_allocator, T, params) _OSCL_TRAP_NEW(T_allocator.allocate(1),T_allocator.deallocate,T_ptr,T,params)
- #define OSCL_ALLOC_DELETE(ptr, T_allocator, T)
- #define OSCL_MALLOC(count) _oscl_default_audit_malloc(count)
- #define oscl_malloc(a) OSCL_MALLOC(a)
- #define OSCL_DEFAULT_MALLOC(x) OSCL_MALLOC(x)
- #define OSCL_AUDIT_MALLOC(auditCB, count) _oscl_audit_malloc(count, auditCB)
- #define OSCL_CALLOC(num, size) _oscl_default_audit_calloc(num,size)
- #define oscl_calloc(a, b) OSCL_CALLOC(a,b)
- #define OSCL_AUDIT_CALLOC(auditCB, num, size) _oscl_audit_calloc(num,size, auditCB)
- #define OSCL_REALLOC(ptr, new_size) _oscl_default_audit_realloc(ptr,new_size)
- #define oscl_realloc(a, b) OSCL_REALLOC(a,b)
- #define OSCL_AUDIT_REALLOC(auditCB, ptr, new_size) _oscl_audit_realloc(ptr,new_size, auditCB)
- #define OSCL_FREE(ptr) _oscl_audit_free(ptr)
- #define oscl_free(x) OSCL_FREE(x)
- #define OSCL_DEFAULT_FREE(x) OSCL_FREE(x)
- #define OSCL_NEW(T, params) new T params
- #define OSCL_PLACEMENT_NEW(ptr, constructor) new(ptr) constructor
- #define OSCL_TRAP_NEW(T_ptr, T, params) _OSCL_TRAP_NEW(_oscl_default_audit_new(sizeof(T)),_oscl_audit_free,T_ptr,T,params)
- #define OSCL_AUDIT_NEW(auditCB, T, params) new(_oscl_audit_new(sizeof(T),auditCB)) T params
- #define OSCL_TRAP_AUDIT_NEW(T_ptr, auditCB, T, params) _OSCL_TRAP_NEW(_oscl_audit_new(sizeof(T),auditCB),_oscl_audit_free,T_ptr,T,params)
- #define OSCL_DELETE(ptr)
- #define OSCL_AUDIT_ARRAY_NEW(auditCB, T, count) new(_oscl_audit_new(sizeof(T)*(count),auditCB)) T
- #define OSCL_ARRAY_NEW(T, count) new T[count]
- #define OSCL_ARRAY_DELETE(ptr) delete [] ptr
- #define OSCL_TRAP_NEW(exp, freeFunc, T_ptr, T, params)

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

Typedefs

- typedef **OSCLMemAutoPtr< char, Oscl_TAlloc< char, OsclMemBasicAllocator > >** **MMAudit_CharAutoPtr**
- typedef **OSCLMemAutoPtr< uint8, Oscl_TAlloc< uint8, _OsclBasicAllocator > >** **MMAudit_Uint8AutoPtr**
- typedef **OSCLMemAutoPtr< MM_AllocNode, Oscl_TAlloc< MM_AllocNode, OsclMemBasicAllocator > >** **MM_AllocNodeAutoPtr**
- typedef **OSCLMemAutoPtr< OsclMemStatsNode, Oscl_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > >** **MM_StatsNodeTagTreeType**
- typedef **OSCLMemAutoPtr< OsclMemStatsNode, Oscl_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > >** **OsclMemStatsNodeAutoPtr**
- typedef **Oscl_TAlloc< MM_StatsNodeTagTreeType, OsclMemBasicAllocator >** **TagTreeAllocator**
- typedef **Oscl_TagTree< MM_StatsNodeTagTreeType, TagTreeAllocator >** **OsclTagTreeType**

Functions

- **OSCL_COND_IMPORT_REF void * _oscl_malloc** (int32 count)
- **OSCL_COND_IMPORT_REF void * _oscl_calloc** (int32 nelems, int32 size)
- **OSCL_COND_IMPORT_REF void * _oscl_realloc** (void *src, int32 count)
- **OSCL_COND_IMPORT_REF void _oscl_free** (void *src)
- **OSCL_COND_IMPORT_REF void * oscl_memcpy** (void *dest, const void *src, uint32 count)
- **OSCL_COND_IMPORT_REF void * oscl_memmove** (void *dest, const void *src, uint32 count)
- **OSCL_COND_IMPORT_REF void * oscl_memmove32** (void *dest, const void *src, uint32 count)
- **OSCL_COND_IMPORT_REF void * oscl_memset** (void *dest, uint8 val, uint32 count)

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

Variables

- const uint32 `ALLOC_NODE_FLAG` = 0x80000000

6.3.1 Define Documentation

6.3.1.1 #define _OSCL_CLEANUP_BASE_CLASS(T) this → T::~T()

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

Parameters:

T: base class name.

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

Value:

```
{
    int32 __err; \
    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),audit-CB)) T params

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

Parameters:

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

Returns:

pointer to the newly created object of type *T*

Exceptions:

may leave with code = bad alloc

**6.3.1.32 #define OSCL_AUDIT_REALLOC(auditCB, ptr, new_size)
_oscl_audit_realloc(ptr,new_size,auditCB)**

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

Parameters:

auditCB input memory management audit object
ptr original memory block
new_size New size of the block

Returns:

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

Exceptions:

none

6.3.1.33 #define oscl_calloc(a, b) OSCL_CALLOC(a,b)**6.3.1.34 #define OSCL_CALLOC(num, size) _oscl_default_audit_calloc(num,size)**

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

Parameters:

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

Returns:

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

Exceptions:

none

6.3.1.35 #define OSCL_CLEANUP_BASE_CLASS(T) _OSCL_CLEANUP_BASE_CLASS(T)

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

Parameters:

T: name of the base class.

6.3.1.36 #define OSCL_DEFAULT_FREE(x) OSCL_FREE(x)

Another back-compatibility definition.

6.3.1.37 #define OSCL_DEFAULT_MALLOC(x) OSCL_MALLOC(x)

Another back-compatibility definition.

6.3.1.38 #define OSCL_DELETE(ptr)**Value:**

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

Oscl "delete" operator.

Parameters:

ptr pointer to memory block previously allocated with OSCL_NEW

Returns:

void

6.3.1.39 #define OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT**6.3.1.40 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE**

Previously this was in oscl_mem_imp.h

6.3.1.41 #define oscl_free(x) OSCL_FREE(x)**6.3.1.42 #define OSCL_FREE(ptr) _oscl_audit_free(ptr)**

Deallocates or frees a memory block.

Parameters:

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

6.3.1.43 #define OSCL_HAS_GLOBAL_NEW_DELETE 1

6.3.1.44 #define oscl_malloc(a) OSCL_MALLOC(a)

6.3.1.45 #define OSCL_MALLOC(count) _oscl_default_audit_malloc(count)

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

Parameters:

count number of bytes to allocate

Returns:

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

Exceptions:

none

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

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

Parameters:

T data type for 'new' operation

params object initialization parameters

Returns:

pointer to the newly created object of type T

Exceptions:

may leave with code = bad alloc

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

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

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

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

Parameters:

ptr original memory block

new_size New size of the block

Returns:

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

Exceptions:

none

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

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

Parameters:

T_ptr variable to hold return value— pointer to new object of type T.

T_allocator allocator for objects of type T, must be an [Oscl_TAlloc<T, Allocator>](#), where Allocator is an [Oscl_DefAlloc](#)

T type of object to create

params object initialization parameters

Returns:

pointer to created object

Exceptions:

none , unless thrown by the given allocator

**6.3.1.51 #define OSCL_TRAP_AUDIT_NEW(T_ptr, auditCB, T, params) _OSCL_TRAP_-
 NEW(_oscl_audit_new(sizeof(T),auditCB),_oscl_audit_free,T_ptr,T,params)**

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

Parameters:

T_ptr variable to hold return value— pointer to new object of type T.

auditCB input memory management audit object

T data type for 'new' operation

params object initialization parameters

Returns:

pointer to the newly created object of type T

Exceptions:

may leave with code = bad alloc

**6.3.1.52 #define OSCL_TRAP_NEW(T_ptr, T, params) _OSCL_TRAP_NEW(_oscl_default_-
 audit_new(sizeof(T)),_oscl_audit_free,T_ptr,T,params)**

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

Parameters:

T_ptr variable to hold return value— pointer to new object of type T.

T data type for 'new' operation

params object initialization parameters

Returns:

pointer to the newly created object of type T

Exceptions:

may leave with code = bad alloc

6.3.2 Typedef Documentation

- 6.3.2.1 `typedef OSCLMemAutoPtr<MM_AllocNode, Oscl_TAlloc<MM_AllocNode, OsclMemBasicAllocator> > MM_AllocNodeAutoPtr`
- 6.3.2.2 `typedef OSCLMemAutoPtr<OsclMemStatsNode, Oscl_TAlloc<OsclMemStatsNode, OsclMemBasicAllocator> > MM_StatsNodeTagTreeType`
- 6.3.2.3 `typedef OSCLMemAutoPtr<char, Oscl_TAlloc<char, OsclMemBasicAllocator> > MMAuditCharAutoPtr`
- 6.3.2.4 `typedef OSCLMemAutoPtr<uint8, Oscl_TAlloc<uint8, _OsclBasicAllocator> > MMAuditUint8AutoPtr`
- 6.3.2.5 `typedef OSCLMemAutoPtr<OsclMemStatsNode, Oscl_TAlloc<OsclMemStatsNode, OsclMemBasicAllocator> > OsclMemStatsNodeAutoPtr`
- 6.3.2.6 `typedef Oscl_TagTree<MM_StatsNodeTagTreeType, TagTree_Allocator> OsclTagTreeType`
- 6.3.2.7 `typedef Oscl_TAlloc<MM_StatsNodeTagTreeType, OsclMemBasicAllocator> TagTree_Allocator`

6.3.3 Function Documentation

- 6.3.3.1 `OSCL_IMPORT_REF void* _oscl_audit_calloc (size_t, size_t, OsclAuditCB &, const char *f = NULL, const int l = 0)`
- 6.3.3.2 `OSCL_IMPORT_REF void _oscl_audit_free (void *)`
- 6.3.3.3 `OSCL_IMPORT_REF void* _oscl_audit_malloc (size_t, OsclAuditCB &, const char *f = NULL, const int l = 0)`

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

- 6.3.3.4 OSCL_IMPORT_REF void* _oscl_audit_new (size_t, OsclAuditCB &, const char **f*=NULL, const int *l*=0)**
- 6.3.3.5 OSCL_IMPORT_REF void* _oscl_audit_realloc (void *, size_t, OsclAuditCB &, const char **f*=NULL, const int *l*=0)**
- 6.3.3.6 OSCL_COND_IMPORT_REF void* _oscl_calloc (int32 *nelems*, int32 *size*)**
- 6.3.3.7 OSCL_IMPORT_REF void* _oscl_default_audit_calloc (size_t, size_t, const char **f*=NULL, const int *l*=0)**
- 6.3.3.8 OSCL_IMPORT_REF void* _oscl_default_audit_malloc (size_t, const char **f*=NULL, const int *l*=0)**
- 6.3.3.9 OSCL_IMPORT_REF void* _oscl_default_audit_new (size_t, const char **f*=NULL, const int *l*=0)**
- 6.3.3.10 OSCL_IMPORT_REF void* _oscl_default_audit_realloc (void *, size_t, const char **f*=NULL, const int *l*=0)**
- 6.3.3.11 OSCL_COND_IMPORT_REF void _oscl_free (void **src*)**
- 6.3.3.12 OSCL_COND_IMPORT_REF void* _oscl_malloc (int32 *count*)**
- 6.3.3.13 OSCL_COND_IMPORT_REF void* _oscl_realloc (void **src*, int32 *count*)**
- 6.3.3.14 void operator delete (void *) [inline]**
- 6.3.3.15]**
- void operator delete[] (void **aPtr*) [inline]
- 6.3.3.16 void* operator new (size_t) [inline]**
- 6.3.3.17 void* operator new (size_t *aSize*, const char **aFile*, int *aLine*) [inline]**
- 6.3.3.18]**
- void* operator new[] (size_t *aSize*) [inline]
- 6.3.3.19]**
- void* operator new[] (size_t *aSize*, const char **aFile*, int *aLine*) [inline]
- 6.3.3.20 OSCL_COND_IMPORT_REF uint oscl_mem_aligned_size (uint *size*)**

Get memory-aligned size of an object.

Parameters:

size size of object

Returns:

memory-aligned size

6.3.3.21 OSCL_COND_IMPORT_REF int oscl_memcmp (const void * buf1, const void * buf2, uint32 count)

Compare characters in two buffers

Parameters:

buf1 first buffer

buf2 second buffer

count number of bytes to compare

Returns:

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

6.3.3.22 OSCL_COND_IMPORT_REF void* oscl_memcpy (void * dest, const void * src, uint32 count)

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

Parameters:

dest new buffer

src buffer to copy

count number of bytes to copy

Returns:

the value of dest

6.3.3.23 OSCL_COND_IMPORT_REF void* oscl_memmove (void * dest, const void * src, uint32 count)

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

Parameters:

dest new buffer

src buffer to copy

count number of bytes to copy

Returns:

the value of dest

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

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

Parameters:

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

Returns:

the value of dest

6.3.3.25 OSCL_COND_IMPORT_REF void* oscl_memset (void * dest, uint8 val, uint32 count)

Sets the bytes of a buffer to a specified character

Parameters:

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

Returns:

the value of dest

6.3.3.26 OSCL_IMPORT_REF void OsclMemInit ([OsclAuditCB](#) & auditCB)

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

Parameters:

auditCB memory management audit object

6.3.4 Variable Documentation

6.3.4.1 const uint32 MM_AllocBlockHdr::ALLOC_NODE_FLAG = 0x80000000 [static, inherited]

6.4 OSCL Util

Files

- file [oscl_bin_stream.h](#)
Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.
- file [oscl_math.h](#)
Provides math functions.
- file [oscl_media_data.h](#)
Defines a container class for media data made up of a collection of memory fragments.
- file [oscl_media_status.h](#)
Defines a status values for the [MediaData](#) containers.
- file [oscl_pqueue.h](#)
Implements a priority queue data structure similar to STL.
- file [oscl_rand.h](#)
Provides pseudo-random number generation.
- file [oscl_registry_access_client.h](#)
Client-side implementation Registry Access implementation.
- file [oscl_registry_client.h](#)
Client-side implementation of OsclRegistry.
- file [oscl_registry_client_impl.h](#)
Client-side implementation of OsclRegistryInterface.
- file [oscl_registry_serv_impl.h](#)
Server-side implementation of OsclRegistry interfaces.
- file [oscl_registry_types.h](#)
Common types used in Oscl registry interfaces.
- file [oscl_snprintf.h](#)
Provides a portable implementation of sprintf.
- file [oscl_str_ptr_len.h](#)
Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.
- file [oscl_string.h](#)
Provides a standardized set of string containers that can be used in place of character arrays.
- file [oscl_string_containers.h](#)
Provides a standardized set of string containers that can be used in place of character arrays.

- file [oscl_string_rep.h](#)
Contains some internal implementation for string containers.
- file [oscl_string_uri.h](#)
Utilities to unescape URIs.
- file [oscl_string_utf8.h](#)
Utilities to validate and truncate UTF-8 encoded strings.
- file [oscl_string_utils.h](#)
Utilities to parse and convert strings.
- file [oscl_string_xml.h](#)
Utilities to escape special characters in XML strings.
- file [oscl_tickcount.h](#)
Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.
- file [oscl_utf8conv.h](#)
Utilities to convert unicode to utf8 and vice versa.

Data Structures

- class [BufferFragment](#)
- class [BufferMgr](#)
- class [BufferState](#)
- class [BufFragGroup](#)
- class [BufFragStatusClass](#)
- class [CFastRep](#)
- class [CHheapRep](#)
- class [CStackRep](#)
- class [MediaData](#)
- class [MediaStatusClass](#)
- class [MemAllocator](#)
- class [OSCL_FastString](#)
- class [OSCL_HeapString](#)
- class [OSCL_HeapStringA](#)
- class [OSCL_StackString](#)
- class [OSCL_String](#)
- class [OSCL_wFastString](#)
- class [OSCL_wHeapString](#)
- class [OSCL_wHeapStringA](#)
- class [OSCL_wStackString](#)
- class [OSCL_wString](#)
- class [OsclBinIStream](#)
- class [OsclBinIStreamBigEndian](#)
- class [OsclBinIStreamLittleEndian](#)

- class [OsclBinOStream](#)

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

- class [OsclBinOStreamBigEndian](#)

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

- class [OsclBinOStreamLittleEndian](#)

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

- class [OsclBinStream](#)

- class [OsclCompareLess](#)

- class [OsclComponentRegistry](#)

- class [OsclComponentRegistryData](#)

- class [OsclComponentRegistryElement](#)

- class [OsclPriorityQueue](#)

- class [OsclPriorityQueueBase](#)

- class [OsclRand](#)

- class [OsclRegistryAccessClient](#)

- class [OsclRegistryAccessClientImpl](#)

- class [OsclRegistryAccessClientTlsImpl](#)

- class [OsclRegistryAccessElement](#)

- class [OsclRegistryClient](#)

- class [OsclRegistryClientImpl](#)

- class [OsclRegistryClientTlsImpl](#)

- class [OsclRegistryServTlsImpl](#)

- class [OsclTickCount](#)

- struct [StrCSumPtrLen](#)

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

- struct [StrPtrLen](#)

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

- struct [WStrPtrLen](#)

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

Defines

- #define [oscl_isdigit](#)(c) ((c) >= '0' && (c) <= '9')
- #define [OSCLTICKCOUNT_MAX_TICKS](#) 0xffffffff
- #define [MAX_NUMBER_OF_BYTE_PER_UTF8](#) 3

Typedefs

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

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

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

Enumerations

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

Functions

- **OSCL_IMPORT_REF const char * skip_whitespace** (const char *ptr)
- **OSCL_IMPORT_REF char * skip_whitespace** (char *ptr)
- **OSCL_IMPORT_REF const char * skip_whitespace** (const char *start, const char *end)
- **OSCL_IMPORT_REF const char * skip_to_whitespace** (const char *start, const char *end)
- **OSCL_IMPORT_REF const char * skip_to_line_term** (const char *start_ptr, const char *end_ptr)
- **OSCL_IMPORT_REF const char * skip_whitespace_and_line_term** (const char *start, const char *end)
- **OSCL_IMPORT_REF int extract_string** (const char *in_ptr, char *outstring, int maxsize)
- **OSCL_IMPORT_REF int extract_string** (const char *start, const char *end, char *outstring, int maxsize)
- **OSCL_IMPORT_REF bool PV_atoi** (const char *buf, const char new_format, uint32 &value)
- **OSCL_IMPORT_REF bool PV_atoi** (const char *buf, const char new_format, int length, uint32 &value)
- **OSCL_IMPORT_REF bool PV_atoi** (const char *buf, const char new_format, int length, [uint64](#) &value)
- **OSCL_IMPORT_REF bool PV_atof** (const char *buf, [OsclFloat](#) &value)
- **OSCL_IMPORT_REF bool PV_atof** (const char *buf, int length, [OsclFloat](#) &value)
- **OSCL_IMPORT_REF int oscl_abs** (int aVal)
- **OSCL_COND_IMPORT_REF double oscl_log** (double value)
- **OSCL_COND_IMPORT_REF double oscl_log10** (double value)
- **OSCL_COND_IMPORT_REF double oscl_sqrt** (double value)
- **OSCL_COND_IMPORT_REF double oscl_pow** (double x, double y)
- **OSCL_COND_IMPORT_REF double oscl_exp** (double value)
- **OSCL_COND_IMPORT_REF double oscl_sin** (double value)
- **OSCL_COND_IMPORT_REF double oscl_cos** (double value)
- **OSCL_COND_IMPORT_REF double oscl_tan** (double value)
- **OSCL_COND_IMPORT_REF double oscl_asin** (double value)
- **OSCL_COND_IMPORT_REF double oscl_atan** (double value)
- **OSCL_COND_IMPORT_REF double oscl_floor** (double value)
- **OSCL_IMPORT_REF int32 oscl_snprintf** (char *str, uint32 count, const char *fmt,...)
- **OSCL_IMPORT_REF int32 oscl_snprintf** ([oscl_wchar](#) *str, uint32 count, const [oscl_wchar](#) *fmt,...)
- **OSCL_IMPORT_REF int32 oscl_vsnprintf** (char *str, uint32 count, const char *fmt, va_list args)

- OSCL_IMPORT_REF int32 `oscl_vsnprintf` (oscl_wchar *str, uint32 count, const oscl_wchar *fmt, va_list args)
- OSCL_IMPORT_REF bool `oscl_str_unescape_uri` (const char *str_buf_in, char *str_buf_out, uint32 max_out_buf_bytes, uint32 max_bytes, uint32 &out_buf_len)

unescape any of the special escape sequence in the uri string
- OSCL_IMPORT_REF bool `oscl_str_unescape_uri` (const OSCL_String &oscl_str_in, OSCL_String &oscl_str_out, uint32 &out_buf_len)

unescape any of the special escape sequence in the uri string
- OSCL_IMPORT_REF bool `oscl_str_is_valid_utf8` (const uint8 *str_buf, uint32 &num_valid_characters, uint32 max_bytes=0, uint32 max_char_2_valid=0, uint32 *num_byte_4_char=NULL)

Check if the input string contains any illegal UTF-8 character. The function scans the string and validate that each character is a valid utf-8. It stops at the first NULL character, invalid character or the max_byte value. The string is valid if and only if every character is a valid utf-8 character and the scanning stopped on a character boundary.
- OSCL_IMPORT_REF int32 `oscl_str_truncate_utf8` (uint8 *str_buf, uint32 max_char, uint32 max_bytes=0)

Truncates the UTF-8 string upto the required size.
- OSCL_IMPORT_REF bool `oscl_str_need_escape_xml` (const char *str_buf, uint32 &num_escape_bytes, uint32 max_bytes=0)

Check if the input string contains any special ASCII character like &, <, >, ', ". The function scans the string and check if each character is a special character. It stops at the first NULL character (if max_bytes = 0), or the max_byte value.
- OSCL_IMPORT_REF int32 `oscl_str_escape_xml` (const char *str_buf_in, char *str_buf_out, uint32 max_out_buf_bytes, uint32 max_bytes=0, uint32 *num_bytes_written=NULL)

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

Convert UTF8 byte sequence to Unicode string.
- OSCL_IMPORT_REF int32 `oscl_UnicodeToUTF8` (const oscl_wchar *input, int32 inLength, char *output, int32 outLength)

Convert Unicode string to UTF8 byte sequence.
- `BufferFragment * GetFragment` (const int32 idx)
- `BufferState * GetBufferState` (const int32 idx)
- uint32 `get_size` () const
- uint32 `get_size` () const
- uint32 `get_maxsize` () const
- uint32 `get_maxsize` () const
- const chartype * `get_cstr` () const
- const chartype * `get_cstr` () const
- chartype * `get_str` () const
- chartype * `get_str` () const
- `OSCL_HeapString` ()
- `OSCL_wHeapString` ()

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

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

Variables

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

6.4.1 Define Documentation

6.4.1.1 #define MAX_NUMBER_OF_BYTE_PER_UTF8 3

Define the maximum UTF8 representation in bytes.

Todo:

Handle 4-byte surrogate pair representation

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

6.4.1.3 #define OSCLTICKCOUNT_MAX_TICKS 0xffffffff

6.4.2 Typedef Documentation

6.4.2.1 typedef void(* BufferFreeFuncPtr)(void *)

6.4.2.2 typedef uint32 MediaTimestamp

6.4.2.3 typedef WStrPtrLen OSCL_TStrPtrLen

6.4.2.4 typedef OsclAny* OsclComponentFactory

OsclComponentFactory is an opaque pointer.

6.4.2.5 typedef StrCSumPtrLen StrCSumPtrLen

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

6.4.2.6 typedef struct StrPtrLen StrPtrLen

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

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

6.4.2.7 `typedef struct WStrPtrLen WStrPtrLen`

This data structure encapsulates a set of functions used to perform standard string operations. It should be used for null-terminated constant strings (non-modifiable) of wchar type.

6.4.3 Enumeration Type Documentation

6.4.3.1 `enum TOSCL_StringOp`

Conversion operations for [OSCL_String](#) classes

Enumeration values:

`EOSCL_StringOp_CompressASCII`
`EOSCL_StringOp_UTF16ToUTF8`

6.4.3.2 `enum TOSCL_wStringOp`

Conversion operations for [OSCL_wString](#) classes

Enumeration values:

`EOSCL_wStringOp_ExpandASCII`
`EOSCL_wStringOp_UTF8ToUTF16`

6.4.4 Function Documentation

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

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

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

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

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

Reimplemented from [OSCL_wString](#).

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

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

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

Reimplemented from [OSCL_wString](#).

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

Assignment operator

Reimplemented from [OSCL_String](#).

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

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

Assignment operators

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

Reimplemented from [OSCL_wString](#).

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

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

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

Reimplemented from [OSCL_wString](#).

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

Assignment operator

Reimplemented from [OSCL_String](#).

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

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

Assignment operators

6.4.4.33 OSCL_IMPORT_REF int oscl_abs (int aVal)

6.4.4.34 OSCL_COND_IMPORT_REF double oscl_asin (double value)

Calculates the arc sine of a number

Parameters:

value source value

6.4.4.35 OSCL_COND_IMPORT_REF double oscl_atan (double value)

Calculates the arc tangent of a number

Parameters:

value source value

6.4.4.36 OSCL_COND_IMPORT_REF double oscl_cos (double value)

Calculates the cosine of a number

Parameters:

value source value

6.4.4.37 OSCL_COND_IMPORT_REF double oscl_exp (double value)

Calculates the exponential of e for a number

Parameters:

value source value

6.4.4.38 OSCL_COND_IMPORT_REF double oscl_floor (double value)

Calculates the floor of a number

Parameters:

value source value

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

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

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

Parameters:

src: input string.

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

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

Parameters:

src: character array, not necessarily null-terminated.

length: number of characters to copy.

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

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

Parameters:

cp: null-terminated string.

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

The default constructor creates an empty string.

6.4.4.44 OSCL_COND_IMPORT_REF double oscl_log (double value)

Calculates the natural log of a number

Parameters:

value source value

6.4.4.45 OSCL_COND_IMPORT_REF double oscl_log10 (double value)

Calculates the logarithm to base 10 of a number

Parameters:

value source value

6.4.4.46 OSCL_COND_IMPORT_REF double oscl_pow (double x, double y)

Calculates the value of x to the power of y

Parameters:

x base value

y power

6.4.4.47 OSCL_COND_IMPORT_REF double oscl_sin (double *value*)

Calculates the sine of a number

Parameters:

value source value

6.4.4.48 OSCL_IMPORT_REF int32 oscl_snprintf (*oscl_wchar* * *str*, uint32 *count*, const *oscl_wchar* * *fmt*, ...)**6.4.4.49 OSCL_IMPORT_REF int32 oscl_snprintf (char * *str*, uint32 *count*, const char * *fmt*, ...)****6.4.4.50 OSCL_COND_IMPORT_REF double oscl_sqrt (double *value*)**

Calculates the square root of a number

Parameters:

value source value

6.4.4.51 template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize >::OSCL_StackString (const OSCL_String & *src*) [inherited]**6.4.4.52 template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize >::OSCL_StackString (const OSCL_StackString< MaxBufSize > & *src*) [inherited]**

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

Parameters:

src: input string.

6.4.4.53 template<uint32 MaxBufSize> OSCL_StackString< MaxBufSize >::OSCL_StackString (const *chartype* * *buf*, uint32 *length*) [inherited]

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

Parameters:

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

length: the number of characters to copy.

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

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

Parameters:

cp: a null-terminated string.

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

Creates an OSCL_StackString initialized with an empty string.

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

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

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

Parameters:

str_buf_in Ptr to an input string

str_buf_out Ptr to an output buffer which stores the modified string

max_out_buf_bytes The size of str_buf_out.

max_bytes The maximum number of bytes to read (a zero value means read to the first NULL character). It is the length of str_buf_in.

num_bytes_written Number of bytes written in the output buffer, str_buf_out

Returns:

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

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

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

Parameters:

str_buf Ptr to an input string, which may not terminate with null, to be checked

num_valid_chars This is an output parameter which is the number of valid utf-8 characters actually read.

max_bytes The maximum number of bytes to read (a zero value means read to the first NULL character).

max_char_2_valid This is an input parameter. Specify the number of utf-8 characters the caller wants to validate.

num_byte_4_char This is an output parameter. The number of bytes used by the max_char characters

Returns:

True if the string is valid and false otherwise.

6.4.4.58 OSCL_IMPORT_REF bool oscl_str_need_escape_xml (const char * str_buf, uint32 & num_escape_bytes, uint32 max_bytes = 0)

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

Parameters:

str_buf Ptr to an input string, which may not terminate with null, to be checked

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

max_bytes The maximum number of bytes to read (a zero value means read to the first NULL character).

Returns:

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

6.4.4.59 OSCL_IMPORT_REF int32 oscl_str_truncate_utf8 (uint8 * str_buf, uint32 max_char, uint32 max_bytes = 0)

Truncates the UTF-8 string upto the required size.

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

Parameters:

str_buf Ptr to an input string which may not terminate with null

max_char The max number of the UTF-8 CHARACTERS

max_bytes The maximum number of bytes to read (a zero value means read to the first NULL character).

Returns:

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

**6.4.4.60 OSCL_IMPORT_REF bool oscl_str_unescape_uri (const OSCL_String & oscl_str_in,
OSCL_String & oscl_str_out, uint32 & out_buf_len)**

unescape any of the special escape sequence in the uri string

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

Parameters:

oscl_str_in Ptr to an input [OSCL_String](#)

oscl_str_out Ptr to an output [OSCL_String](#) which stores the modified string

out_buf_len The length of the result string (not including the null character)

Returns:

It returns true if succeeds, otherwise false.

**6.4.4.61 OSCL_IMPORT_REF bool oscl_str_unescape_uri (const char * str_buf_in, char *
str_buf_out, uint32 max_out_buf_bytes, uint32 max_bytes, uint32 & out_buf_len)**

unescape any of the special escape sequence in the uri string

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

Parameters:

str_buf_in Ptr to an input string

str_buf_out Ptr to an output buffer which stores the modified string

max_out_buf_bytes The size of str_buf_out.

max_bytes The maximum number of bytes to read. It is the length of str_buf_in.

out_buf_len The length of the result string (not including the null character)

Returns:

It returns true if succeeds, otherwise false.

6.4.4.62 OSCL_COND_IMPORT_REF double oscl_tan (double value)

Calculates the tangential of a number

Parameters:

value source value

6.4.4.63 OSCL_IMPORT_REF int32 oscl_UnicodeToUTF8 (const oscl_wchar * *input*, int32 *inLength*, char * *output*, int32 *outLength*)

Convert Unicode string to UTF8 byte sequence.

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

Parameters:

input Ptr to an input Unicode string. '\0' termination is not necessary.

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

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

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

Returns:

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

6.4.4.64 OSCL_IMPORT_REF int32 oscl_UTF8ToUnicode (const char * *input*, int32 *inLength*, oscl_wchar * *output*, int32 *outLength*)

Convert UTF8 byte sequence to Unicode string.

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

Parameters:

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

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

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

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

Returns:

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

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

Parameters:

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

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

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

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

Parameters:

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

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

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

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

Parameters:

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

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

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

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

Parameters:

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

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

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

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

Parameters:

buf: NULL-terminated wide string.

op: conversion operation to apply

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

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

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

Parameters:

buf: string or character array.

length: number of characters to copy.

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

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

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

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

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

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

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

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

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

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

6.4.5 Variable Documentation

6.4.5.1 const int32 APPEND_MEDIA_AT_END = -1

6.4.5.2 const uint8 OSCL_ASCII_CASE_MAGIC_BIT = 0x20

6.5 OSCL Error

Files

- file [oscl_errno.h](#)
Defines functions to access additional information on errors where supported through an errno or similar service.
- file [oscl_error.h](#)
OSCL Error trap and cleanup include file.
- file [oscl_error_allocator.h](#)
Defines a memory allocation class used by the oscl error layer.
- file [oscl_error_codes.h](#)
Defines basic error and leave codes.
- file [oscl_error_imp.h](#)
Internal error implementation support.
- file [oscl_error_imp_cppexceptions.h](#)
Implementation File for Leave using C++ exceptions.
- file [oscl_error_imp_fatalerror.h](#)
Implementation File for Leave using system fatal error.
- file [oscl_error_imp_jumps.h](#)
Implementation of using Setjmp / Longjmp.
- file [oscl_error_trapcleanup.h](#)
OSCL Error trap and cleanup implementation include file.
- file [oscl_exception.h](#)
contains all the exception handling macros and classes
- file [oscl_heapbase.h](#)
OSCL Heap Base include file.
- file [oscl_mempool_allocator.h](#)
This file contains the definition of memory pool allocator for leave/trap.
- file [oscl_namestring.h](#)
Name string class include file.

Data Structures

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

- class [OsclErrorAllocator](#)

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

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

oscl_exception.h contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

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

Defines

- #define [OSCL_TRAPSTACK_PUSH](#)(a) OsclError::PushL(a)
- #define [OSCL_TRAPSTACK_POP](#)() OsclError::Pop()
- #define [OSCL_TRAPSTACK_POPDEALLOC](#)() OsclError::PopDealloc()
- #define [OsclErrNone](#) 0
- #define [OsclErrGeneral](#) 100
- #define [OsclErrNoMemory](#) 101
- #define [OsclErrCancelled](#) 102
- #define [OsclErrNotSupported](#) 103
- #define [OsclErrArgument](#) 104
- #define [OsclErrBadHandle](#) 105
- #define [OsclErrAlreadyExists](#) 106
- #define [OsclErrBusy](#) 107
- #define [OsclErrNotReady](#) 108
- #define [OsclErrCorrupt](#) 109
- #define [OsclErrTimeout](#) 110
- #define [OsclErrOverflow](#) 111
- #define [OsclErrUnderflow](#) 112
- #define [OsclErrInvalidState](#) 113
- #define [OsclErrNoResources](#) 114
- #define [OsclErrNotInstalled](#) 115
- #define [OsclErrAlreadyInstalled](#) 116
- #define [OsclErrSystemCallFailed](#) 117
- #define [OsclErrNoHandler](#) 118
- #define [OsclErrThreadContextIncorrect](#) 119
- #define [OSCL_ERR_NONE](#) OsclErrNone
- #define [OSCL_BAD_ALLOC_EXCEPTION_CODE](#) OsclErrNoMemory
- #define [OsclSuccess](#) 0
- #define [OsclPending](#) 1
- #define [OsclFailure](#) -1
- #define [PVERROR_IMP_JUMPS](#)

- #define **PVError_DoLeave()** internalLeave __ilv; __ilv.a=0;throw(__ilv)
- #define **_PV_TRAP(_r, _s)**
- #define **_PV_TRAP_NO_TLS(_trapimp, _r, _s)**
- #define **OSCL_JUMP_MAX_JUMP_MARKS** OSCL_MAX_TRAP_LEVELS
- #define **internalLeave (-1)**
- #define **OSCL_MAX_TRAP_LEVELS** 20
- #define **PVERRORTRAP_REGISTRY_ID** OSCL_TLS_ID_PVERRORTRAP
- #define **PVERRORTRAP_REGISTRY** OsclTLSRegistry
- #define **OSCL_LEAVE(_leave_status)** OsclError::Leave(_leave_status)

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

- #define **OSCL_TRY(_leave_status, _statements)** _PV_TRAP(_leave_status,_statements)

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

- #define **OSCL_TRY_NO_TLS(_trapimp, _leave_status, _statements)** _PV_TRAP_NO_TLS(_-trapimp,_leave_status,_statements)
- #define **OSCL_FIRST_CATCH_ANY(_leave_status, _statements)** if (_leave_status!=OsclErrNone) { _statements; }

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

- #define **OSCL_FIRST_CATCH(_leave_status, _catch_value, _statements)** if (_leave_status!=OsclErrNone && _leave_status == _catch_value){_statements;}

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

- #define **OSCL_CATCH(_leave_status, _catch_value, _statements)** else if (_leave_status!=OsclErrNone && _leave_status == _catch_value){_statements;}

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

- #define **OSCL_CATCH_ANY(_leave_status, _statements)** else if (_leave_status!=OsclErrNone){ _-statements; }

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

- #define **OSCL_LAST_CATCH(_leave_status)** else if (_leave_status!=OsclErrNone){OSCL_-LEAVE(_leave_status);}

Use this macro if OSCL_CATCH_ANY has not been used. It will mark the end of the catch block.

Typedefs

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

Functions

- OSCL_IMPORT_REF bool **OSCL_IsErrnoSupported ()**

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

- OSCL_IMPORT_REF int [OSCL_GetLastError\(\)](#)
This function returns the value of the system's global error number variable.
- OSCL_IMPORT_REF bool [OSCL_SetLastError\(int newVal\)](#)
This function sets the last error code for the system.
- OSCL_IMPORT_REF char * [OSCL_StrError\(int errnum\)](#)
This function maps an error number to an error-message string.

6.5.1 Define Documentation

6.5.1.1 #define _PV_TRAP(__r, __s)

Value:

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

6.5.1.2 #define _PV_TRAP_NO_TLS(__trapimp, __r, __s)

Value:

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

6.5.1.3 #define internalLeave (-1)

6.5.1.4 #define OSCL_BAD_ALLOC_EXCEPTION_CODE OsclErrNoMemory

6.5.1.5 #define OSCL_CATCH(_leave_status, _catch_value, _statements) else if (_leave_status!=OsclErrNone && _leave_status == _catch_value){_statements;}

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

OSCL_FIRST_CATCH can be used to catch one exception type. Then the OSCL_CATCH macro can be used to catch each subsequent type. The catch block must end with OSCL_LAST_CATCH or OSCL_CATCH_ANY

Parameters:

oscl_leave_status is the result of any OSCL_THROW

exceptiontype is the exception handled by this catch block

**6.5.1.6 #define OSCL_CATCH_ANY(_leave_status, _statements) else if
(_leave_status!=OsclErrNone){ _statements;}**

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

Parameters:

_leave_status

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

6.5.1.7 #define OSCL_ERR_NONE OsclErrNone

For backward compatibility with old definitions

**6.5.1.8 #define OSCL_FIRST_CATCH(_leave_status, _catch_value, _statements) if
(_leave_status!=OsclErrNone && _leave_status == _catch_value){_statements;}**

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

Parameters:

oscl_leave_status is the leave code that was returned by OSCL_THROW

exceptiontype is the exception handled by this catch block This macro MUST be used in conjunction with either OSCL_LAST_CATCH or OSCL_CATCH_ANY

**6.5.1.9 #define OSCL_FIRST_CATCH_ANY(_leave_status, _statements) if
(_leave_status!=OsclErrNone) { _statements; }**

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

Parameters:

_leave_status

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

6.5.1.10 #define OSCL_JUMP_MAX_JUMP_MARKS OSCL_MAX_TRAP_LEVELS
**6.5.1.11 #define OSCL_LAST_CATCH(_leave_status) else if (_leave_status!=OsclErr-
None){OSCL_LEAVE(_leave_status);}**

Use this macro if OSCL_CATCH_ANY has not been used. It will mark the end of the catch block.

Parameters:

_leave_status will be propagated up the call stack. This macro will do an OSCL_LEAVE if the leave has not been handled by the calls above. This macro ends the try block.

6.5.1.12 #define OSCL_LEAVE(_leave_status) OsclError::Leave(_leave_status)

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

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

Parameters:

oscl_leave_status tells the cause for the Leave

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

Cleanup Stack user macros

6.5.1.17 #define OSCL_TRY(_leave_status, _statements) _PV_TRAP(_leave_status,_statements)

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

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

Parameters:

oscl_leave_status oscl_leave_status will receive the result of any OSCL_LEAVE (which will get called from a OSCL_THROW) on systems that do not support exception handling. This is unused on systems that do support exception handling

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

6.5.1.18 #define OSCL_TRY_NO_TLS(__trapimp, _leave_status, _statements)
 __PV_TRAP_NO_TLS(__trapimp,_leave_status,_statements)

6.5.1.19 #define OsclErrAlreadyExists 106

6.5.1.20 #define OsclErrAlreadyInstalled 116

6.5.1.21 #define OsclErrArgument 104

6.5.1.22 #define OsclErrBadHandle 105

6.5.1.23 #define OsclErrBusy 107

6.5.1.24 #define OsclErrCancelled 102

6.5.1.25 #define OsclErrCorrupt 109

6.5.1.26 #define OsclErrGeneral 100

6.5.1.27 #define OsclErrInvalidState 113

6.5.1.28 #define OsclErrNoHandler 118

6.5.1.29 #define OsclErrNoMemory 101

6.5.1.30 #define OsclErrNone 0

6.5.1.31 #define OsclErrNoResources 114

6.5.1.32 #define OsclErrNotInstalled 115

6.5.1.33 #define OsclErrNotReady 108

6.5.1.34 #define OsclErrNotSupported 103

6.5.1.35 #define OsclErrOverflow 111

6.5.1.36 #define OsclErrSystemCallFailed 117

6.5.1.37 #define OsclErrThreadContextIncorrect 119

6.5.1.38 #define OsclErrTimeout 110

6.5.1.39 #define OsclErrUnderflow 112

6.5.1.40 #define OsclFailure -1

6.5.1.41 #define OsclPending 1

6.5.1.42 #define OsclSuccess 0

6.5.1.43 #define PVError_DoLeave() internalLeave __ilv; __ilv.a=0; throw(__ilv)

6.5.1.44 #define PVERROR_IMP_JUMPS

6.5.1.45 #define PVERRORTRAP_REGISTRY OsclTLSRegistry

6.5.1.46 #define PVERRORTRAP_REGISTRY_ID OSCL_TLS_ID_PVERRORTRAP

6.5.2 Typedef Documentation

6.5.2.1 typedef int32 OsclLeaveCode

Leave Codes

6.5.2.2 typedef int32 OsclReturnCode

Return Codes

6.5.2.3 typedef void(* OsclTrapOperation)(OsclAny*)

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

6.5.3 Function Documentation

6.5.3.1 OSCL_IMPORT_REF int OSCL_GetLastError ()

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

Returns:

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

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

6.5.3.2 OSCL_IMPORT_REF bool OSCL_IsErrnoSupported ()

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

Returns:

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

6.5.3.3 OSCL_IMPORT_REF bool OSCL_SetLastError (int *newVal*)

This function sets the last error code for the system.

Parameters:

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

6.5.3.4 OSCL_IMPORT_REF char* OSCL_StrError (int *errnum*)

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

Parameters:

errnum This value represents the error number to map

Returns:

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

6.6 OSCL IO

Files

- file `oscl_dns.h`

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

- file `oscl_file_cache.h`

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

- file `oscl_file_dir_utils.h`

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

- file `oscl_file_find.h`

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

- file `oscl_file_handle.h`

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

- file `oscl_file_io.h`

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

- file `oscl_file_native.h`

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

- file `oscl_file_server.h`

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

- file `oscl_file_stats.h`

File stats class.

- file `oscl_file_types.h`

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

- file `oscl_socket.h`

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

Data Structures

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

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

Defines

- #define TOsclFileOffsetInt32 int32
- #define OSCL_FILE_STATS_LOGGER_NODE "OsclFileStats"
- #define OSCL_IO_FILENAME_MAXLEN 512
- #define OSCL_IO_EXTENSION_MAXLEN 512
- #define OSCL_FILE_WCHAR_PATH_DELIMITER _STRLIT("/")
 • #define OSCL_FILE_CHAR_PATH_DELIMITER _STRLIT_CHAR("/")

Typedefs

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

Enumerations

- enum TPVDNSFx { EPVVDNSGetHostByName }
- enum TPVDNSEvent { EPVVDNSSuccess, EPVVDNSPending, EPVVDNSTimeout, EPVVDNSFailure, EPVVDNSCancel }
- enum OSCL_FILEMGMT_PERMS { OSCL_FILEMGMT_PERMS_READ = 0x1, OSCL_FILEMGMT_PERMS_WRITE = 0x2, OSCL_FILEMGMT_PERMS_EXECUTE = 0x4 }
- enum OSCL_FILEMGMT_MODES { OSCL_FILEMGMT_MODE_DIR = 0x1 }
- enum OSCL_FILEMGMT_ERR_TYPE { OSCL_FILEMGMT_E_OK = 0, OSCL_FILEMGMT_E_PATH_TOO_LONG, OSCL_FILEMGMT_E_PATH_NOT_FOUND, OSCL_FILEMGMT_E_ALREADY_EXISTS, OSCL_FILEMGMT_E_NOT_EMPTY, OSCL_FILEMGMT_E_PERMISSION_DENIED, OSCL_FILEMGMT_E_NO_MATCH, OSCL_FILEMGMT_E_UNKNOWN, OSCL_FILEMGMT_E_SYS_SPECIFIC, OSCL_FILEMGMT_E_NOT_IMPLEMENTED }
- enum TOsclFileOp { EOscFileOp_Open, EOscFileOp_Close, EOscFileOp_Read, EOscFileOp_Write, EOscFileOp_Seek, EOscFileOp_Tell, EOscFileOp_Size, EOscFileOp_Flush, EOscFileOp_EndOfFile, EOscFileOp_NativeOpen, EOscFileOp_NativeClose, EOscFileOp_NativeRead, EOscFileOp_NativeWrite, EOscFileOp_NativeSeek, EOscFileOp_NativeTell, EOscFileOp_NativeSize, EOscFileOp_NativeFlush, EOscFileOp_NativeEndOfFile, EOscFileOp_Last }

Functions

- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_getcwd (oscl_wchar *path, uint32 size)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_getcwd (char *path, uint32 size)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_stat (const oscl_wchar *path, OSCL_STAT_BUF *statbuf)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_stat (const char *path, OSCL_STAT_BUF *statbuf)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_mkdir (const oscl_wchar *path)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_mkdir (const char *path)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rmdir (const oscl_wchar *path)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rmdir (const char *path)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_chdir (const oscl_wchar *path)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_chdir (const char *path)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rename (const oscl_wchar *oldpath, const oscl_wchar *newpath)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rename (const char *oldpath, const char *newpath)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_statsfs (OSCL_FSSTAT *stats, const char *path)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_statsfs (OSCL_FSSTAT *stats, const oscl_wchar *path)

6.6.1 Define Documentation

6.6.1.1 #define OSCL_FILE_CHAR_PATH_DELIMITER _STRLIT_CHAR("/")

6.6.1.2 #define OSCL_FILE_STATS_LOGGER_NODE "OsclFileStats"

6.6.1.3 #define OSCL_FILE_WCHAR_PATH_DELIMITER _STRLIT("/")

6.6.1.4 #define OSCL_IO_EXTENSION_MAXLEN 512

6.6.1.5 #define OSCL_IO_FILENAME_MAXLEN 512

6.6.1.6 #define TOsclFileOffsetInt32 int32

6.6.2 Typedef Documentation

6.6.2.1 typedef struct oscl_fsstat OSCL_FSSTAT

6.6.2.2 typedef struct oscl_stat_buf OSCL_STAT_BUF

6.6.2.3 typedef FILE* TOsclFileHandle

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

6.6.3 Enumeration Type Documentation

6.6.3.1 enum OSCL_FILEMGMT_ERR_TYPE

Enumeration values:

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

6.6.3.2 enum OSCL_FILEMGMT_MODES

Enumeration values:

- `OSCL_FILEMGMT_MODE_DIR`

6.6.3.3 enum OSCL_FILEMGMT_PERMS

Enumeration values:

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

6.6.3.4 enum TOsclFileOp

Enumeration values:

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

EOsclFileOp_NativeWrite
EOsclFileOp_NativeSeek
EOsclFileOp_NativeTell
EOsclFileOp_NativeSize
EOsclFileOp_NativeFlush
EOsclFileOp_NativeEndOfFile
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 size of a file in addition to whether the file exists or not

Parameters:

character path the full path of the file to stat.

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

**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 size of a file in addition to whether the file exists or not

Parameters:

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

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

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

Defines

- #define QUE_ITER_BEGIN(_type, _qname)
- #define QUE_ITER_END(_qname)
- #define PVSCHEDNAMELEN 30
- #define OSCL_ZEROIZE(ptr, size) oscl_memset(ptr, 0, size)
- #define PVEXECNAMELEN 30
- #define PV_SCHED_ENABLE_AO_STATS 1
- #define PV_SCHED_ENABLE_LOOP_STATS 0
- #define PV_SCHED_ENABLE_PERF_LOGGING 1
- #define PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS 1
- #define PV_SCHED_LOG_Q 0
- #define PV_SCHED_CHECK_Q 0
- #define PV_SCHED_FAIR_SCHEDULING 1
- #define OSCL_PERF_SUMMARY_LOGGING 0

Typedefs

- typedef PVActiveBase * TOsclReady

Enumerations

- enum TPVThreadContext { EPVThreadContext_InThread, EPVThreadContext_OsclThread, EPVThreadContext_NonOsclThread, EPVThreadContext_Undetermined }

Functions

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

Variables

- const int32 OSCL_REQUEST_ERR_NONE = 0
- const int32 OSCL_REQUEST_PENDING = (-0x7fffffff)
- const int32 OSCL_REQUEST_ERR_CANCEL = (-1)
- const int32 OSCL_REQUEST_ERR_GENERAL = (-2)

6.7.1 Define Documentation

6.7.1.1 #define OSCL_PERF_SUMMARY_LOGGING 0

6.7.1.2 #define OSCL_ZEROIZE(ptr, size) oscl_memset(ptr, 0, size)

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

6.7.1.3 #define PV_SCHED_CHECK_Q 0

6.7.1.4 #define PV_SCHED_ENABLE_AO_STATS 1

6.7.1.5 #define PV_SCHED_ENABLE_LOOP_STATS 0

6.7.1.6 #define PV_SCHED_ENABLE_PERF_LOGGING 1

6.7.1.7 #define PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS 1

6.7.1.8 #define PV_SCHED_FAIR_SCHEDULING 1

6.7.1.9 #define PV_SCHED_LOG_Q 0

6.7.1.10 #define PVEEXECNAMELEN 30

6.7.1.11 #define PVSCHEDEXNAMELEN 30

PV Scheduler class

6.7.1.12 #define QUE_ITER_BEGIN(_type, _qname)

Value:

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

6.7.1.13 #define QUE_ITER_END(_qname)

Value:

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

6.7.2 Typedef Documentation

6.7.2.1 `typedef PVActiveBase* TOsclReady`

6.7.3 Enumeration Type Documentation

6.7.3.1 `enum TPVThreadContext`

Thread context type

Enumeration values:

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

6.7.4 Function Documentation

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

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

6.7.5 Variable Documentation

6.7.5.1 `const int32 OSCL_REQUEST_ERR_CANCEL = (-1)`

6.7.5.2 `const int32 OSCL_REQUEST_ERR_GENERAL = (-2)`

6.7.5.3 `const int32 OSCL_REQUEST_ERR_NONE = 0`

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

6.8 OSCL Init

Files

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

Data Structures

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

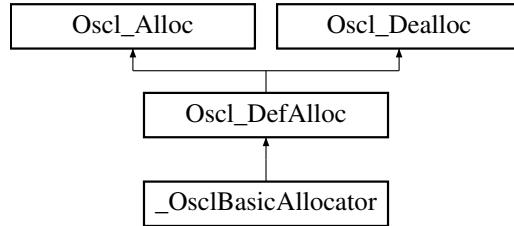
Chapter 7

oscl Data Structure Documentation

7.1 _OsclBasicAllocator Class Reference

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

Inheritance diagram for _OsclBasicAllocator::



Public Methods

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

7.1.1 Detailed Description

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

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

7.1.2 Constructor & Destructor Documentation

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

7.1.3 Member Function Documentation

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

Implements [Oscl_DefAlloc](#).

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

Implements [Oscl_DefAlloc](#).

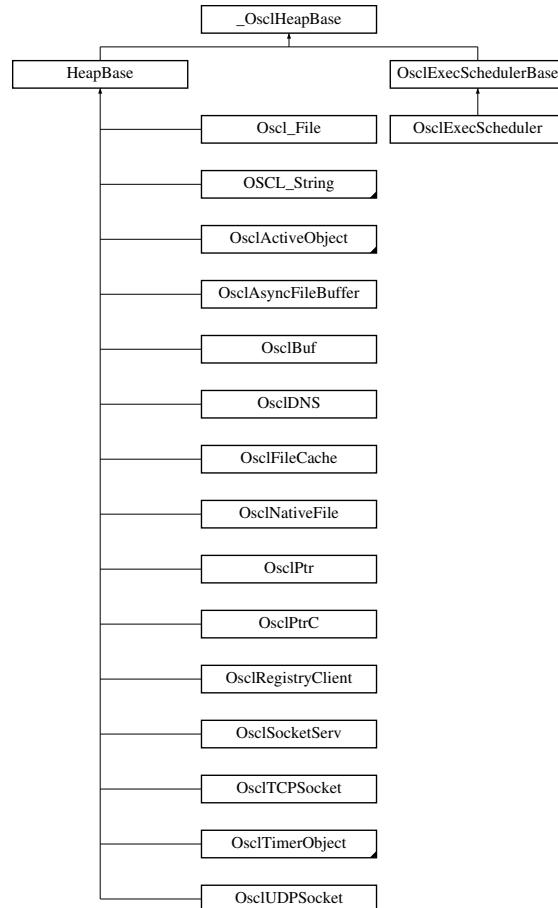
The documentation for this class was generated from the following file:

- [oscl_base_alloc.h](#)

7.2 _OsclHeapBase Class Reference

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

Inheritance diagram for _OsclHeapBase::



Public Methods

- virtual ~_OsclHeapBase ()

Protected Methods

- _OsclHeapBase ()
- _OsclHeapBase (const _OsclHeapBase &)

Friends

- class PVCleanupStack

7.2.1 Detailed Description

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

7.2.2 Constructor & Destructor Documentation

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

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

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

7.2.3 Friends And Related Function Documentation

7.2.3.1 `friend class PVCleanupStack [friend]`

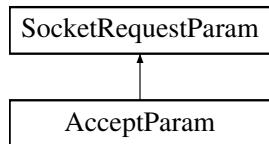
The documentation for this class was generated from the following file:

- [oscl_heapbase.h](#)

7.3 AcceptParam Class Reference

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

Inheritance diagram for AcceptParam::



Public Methods

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

Data Fields

- [OsclSocketI * iBlankSocket](#)

7.3.1 Constructor & Destructor Documentation

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

7.3.2 Field Documentation

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

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

- [oscl_socket_request.h](#)

7.4 allocator Class Reference

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

7.4.1 Detailed Description

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

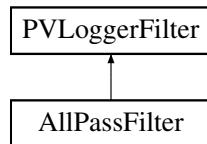
The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

7.5 AllPassFilter Class Reference

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

Inheritance diagram for AllPassFilter::



Public Types

- [typedef PVLoggerFilter::message_id_type message_id_type](#)
- [typedef PVLoggerFilter::log_level_type log_level_type](#)
- [typedef PVLoggerFilter::filter_status_type filter_status_type](#)

Public Methods

- [AllPassFilter \(\)](#)
- [virtual ~AllPassFilter \(\)](#)
- [filter_status_type FilterString \(char *tag, message_id_type msgID, log_level_type level\)](#)
- [filter_status_type FilterOpaqueMessge \(char *tag, message_id_type msgID, log_level_type level\)](#)

7.5.1 Detailed Description

Example filter that allows all messages to be logged.

7.5.2 Member Typedef Documentation

7.5.2.1 [typedef PVLoggerFilter::filter_status_type AllPassFilter::filter_status_type](#)

Reimplemented from [PVLoggerFilter](#).

7.5.2.2 [typedef PVLoggerFilter::log_level_type AllPassFilter::log_level_type](#)

Reimplemented from [PVLoggerFilter](#).

7.5.2.3 [typedef PVLoggerFilter::message_id_type AllPassFilter::message_id_type](#)

Reimplemented from [PVLoggerFilter](#).

7.5.3 Constructor & Destructor Documentation

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

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

7.5.4 Member Function Documentation

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

Implements [PVLoggerFilter](#).

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

Implements [PVLoggerFilter](#).

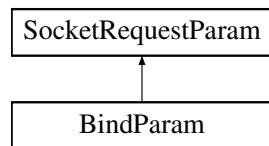
The documentation for this class was generated from the following file:

- [pvlogger_accessories.h](#)

7.6 BindParam Class Reference

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

Inheritance diagram for BindParam::



Public Methods

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

Data Fields

- [OsclNetworkAddress iAddr](#)

7.6.1 Constructor & Destructor Documentation

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

7.6.2 Field Documentation

7.6.2.1 [OsclNetworkAddress BindParam::iAddr](#)

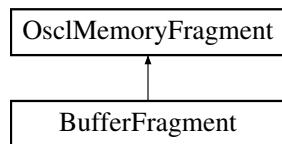
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.7 BufferFragment Class Reference

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

Inheritance diagram for BufferFragment::



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

- [oscl_media_data.h](#)

7.8 BufferMgr Class Reference

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

Public Methods

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

7.8.1 Constructor & Destructor Documentation

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

7.8.2 Member Function Documentation

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

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

- [oscl_media_data.h](#)

7.9 BufferState Class Reference

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

Public Methods

- `BufferState (BufferFreeFuncPtr the_free_function, void *bufptr=0)`
- `BufferState (BufferMgr *the_buf_mgr=0, void *bufptr=0)`
- `void increment_refcnt ()`
- `void decrement_refcnt ()`
- `void bind (void *in_ptr, BufferFreeFuncPtr in_free_function)`
- `void bind (void *in_ptr, BufferMgr *in_buf_mgr)`
- `void * get_ptr ()`
- `int32 getRefCount ()`
- `BufferFreeFuncPtr get_free_function ()`
- `BufferMgr * get_buf_mgr ()`
- `void reset ()`

7.9.1 Constructor & Destructor Documentation

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

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

7.9.2 Member Function Documentation

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

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

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

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

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

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

7.9.2.7 `int32 BufferState::getRefCount ()` [inline]

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

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

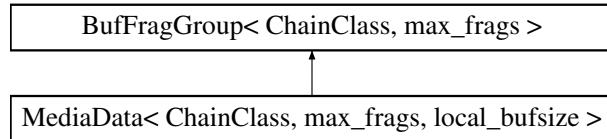
The documentation for this class was generated from the following file:

- `oscl_media_data.h`

7.10 BufFragGroup< ChainClass, max_frags > Class Template Reference

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

Inheritance diagram for BufFragGroup< ChainClass, max_frags >::



Public Methods

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

Protected Methods

- virtual void [Clear \(\)](#)
- [BufFragStatusClass::status_t AddFragment \(const BufferFragment &frag, BufferState *in_buffer_state, int32 location_offset=max_frags\)](#)

Protected Attributes

- [BufferFragment fragments \[max_frags\]](#)
- [BufferState * buffer_states \[max_frags\]](#)
- [ChainClass * next](#)
- uint32 [num.fragments](#)
- uint32 [length](#)

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

7.10.1 Constructor & Destructor Documentation

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

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

7.10.2 Member Function Documentation

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

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

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

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

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

7.10.2.5 template<class ChainClass, uint32 max_frags> int32 BufFragGroup< ChainClass, max_frags >::GetMaxFrags () const [inline]

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

7.10.2.7 template<class ChainClass, uint32 max_frags> int32 BufFragGroup< ChainClass, max_frags >::GetNumFrags () const [inline]

7.10.3 Field Documentation

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

7.10.3.2 template<class ChainClass, uint32 max_frags> BufferFragment BufFragGroup< ChainClass, max_frags >::fragments[max_frags] [protected]

7.10.3.3 template<class ChainClass, uint32 max_frags> uint32 BufFragGroup< ChainClass, max_frags >::length [protected]

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

7.10.3.5 template<class ChainClass, uint32 max_frags> uint32 BufFragGroup< ChainClass, max_frags >::num_fragments [protected]

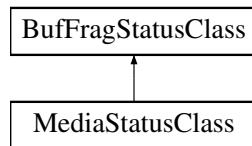
The documentation for this class was generated from the following file:

- [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`

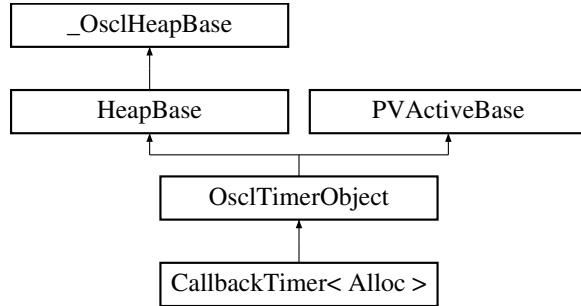
The documentation for this class was generated from the following file:

- [oscl_media_status.h](#)

7.12 CallbackTimer< Alloc > Class Template Reference

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

Inheritance diagram for CallbackTimer< Alloc >::



Public Methods

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

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

7.12.1 Constructor & Destructor Documentation

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

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

7.12.2 Member Function Documentation

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

Handles an active object's request completion event.

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

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

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

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

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

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

Implements [PVActiveBase](#).

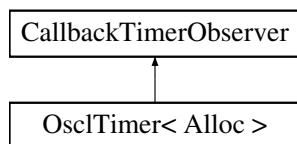
The documentation for this class was generated from the following file:

- [oscl_timer.h](#)

7.13 CallbackTimerObserver Class Reference

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

Inheritance diagram for CallbackTimerObserver::



Public Methods

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

7.13.1 Constructor & Destructor Documentation

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

7.13.2 Member Function Documentation

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

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

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

- [oscl_timer.h](#)

7.14 CFastRep Class Reference

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

Public Methods

- [CFastRep \(\)](#)
- OSCL_IMPORT_REF void [set_w](#) (char *cp, uint32 len, uint32 maxlen)
- OSCL_IMPORT_REF void [set_w \(oscl_wchar](#) *cp, uint32 len, uint32 maxlen)
- OSCL_IMPORT_REF void [set_r](#) (const char *cp, uint32 len)
- OSCL_IMPORT_REF void [set_r \(const oscl_wchar](#) *cp, uint32 len)
- OSCL_IMPORT_REF void [append](#) (const char *cp, uint32 len)
- OSCL_IMPORT_REF void [append \(const oscl_wchar](#) *cp, uint32 len)

Data Fields

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

7.14.1 Detailed Description

For internal use only– fast string representation

7.14.2 Constructor & Destructor Documentation

7.14.2.1 `CFastRep::CFastRep () [inline]`

7.14.3 Member Function Documentation

7.14.3.1 `OSCL_IMPORT_REF void CFastRep::append (const oscl_wchar * cp, uint32 len)`

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

7.14.3.3 `OSCL_IMPORT_REF void CFastRep::set_r (const oscl_wchar * cp, uint32 len)`

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

7.14.3.5 `OSCL_IMPORT_REF void CFastRep::set_w (oscl_wchar * cp, uint32 len, uint32 maxlen)`

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

7.14.4 Field Documentation

7.14.4.1 `OsclAny* CFastRep::buffer`

7.14.4.2 `uint32 CFastRep::maxsize`

7.14.4.3 `bool CFastRep::overwrite`

7.14.4.4 `uint32 CFastRep::size`

7.14.4.5 `bool CFastRep::writable`

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

- `oscl_string_rep.h`

7.15 CHeapRep Class Reference

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

Public Methods

- [CHeapRep \(\)](#)
- OSCL_IMPORT_REF bool [set](#) (uint32, const char *, [Oscl_DefAlloc](#) &)
- OSCL_IMPORT_REF bool [set](#) (uint32, const [oscl_wchar](#) *, [Oscl_DefAlloc](#) &)
- OSCL_IMPORT_REF bool [append](#) (uint32, const char *, uint32, const char *, [Oscl_DefAlloc](#) &)
- OSCL_IMPORT_REF bool [append](#) (uint32, const [oscl_wchar](#) *, uint32, const [oscl_wchar](#) *, [Oscl_DefAlloc](#) &)
- OSCL_IMPORT_REF void [add_ref](#) ()
- OSCL_IMPORT_REF void [remove_ref](#) ([Oscl_DefAlloc](#) &)

Static Public Methods

- OSCL_IMPORT_REF void [set_rep](#) (CHheapRep *&, [Oscl_DefAlloc](#) &, const char *, uint32)
- OSCL_IMPORT_REF void [set_rep](#) (CHheapRep *&, [Oscl_DefAlloc](#) &, const [oscl_wchar](#) *, uint32)
- OSCL_IMPORT_REF void [append_rep](#) (CHheapRep *&, [Oscl_DefAlloc](#) &, const char *, uint32)
- OSCL_IMPORT_REF void [append_rep](#) (CHheapRep *&, [Oscl_DefAlloc](#) &, const [oscl_wchar](#) *, uint32)
- OSCL_IMPORT_REF void [assign](#) (CHheapRep *&, CHheapRep *, [Oscl_DefAlloc](#) &)

Data Fields

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

7.15.1 Detailed Description

For internal use only– heap string representation

7.15.2 Constructor & Destructor Documentation

7.15.2.1 `OSCL_IMPORT_REF void CHeapRep::CHeapRep () [inline]`

7.15.3 Member Function Documentation

7.15.3.1 `OSCL_IMPORT_REF void CHeapRep::add_ref ()`

7.15.3.2 `OSCL_IMPORT_REF bool CHeapRep::append (uint32, const oscl_wchar *, uint32, const oscl_wchar *, Oscl_DefAlloc &)`

7.15.3.3 `OSCL_IMPORT_REF bool CHeapRep::append (uint32, const char *, uint32, const char *, Oscl_DefAlloc &)`

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

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

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

7.15.3.7 `OSCL_IMPORT_REF void CHeapRep::remove_ref (Oscl_DefAlloc &)`

7.15.3.8 `OSCL_IMPORT_REF bool CHeapRep::set (uint32, const oscl_wchar *, Oscl_DefAlloc &)`

7.15.3.9 `OSCL_IMPORT_REF bool CHeapRep::set (uint32, const char *, Oscl_DefAlloc &)`

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

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

7.15.4 Field Documentation

7.15.4.1 `OsclAny* CHeapRep::buffer`

7.15.4.2 `uint32 CHeapRep::maxsize`

7.15.4.3 `uint32 CHeapRep::refcount`

7.15.4.4 `uint32 CHeapRep::size`

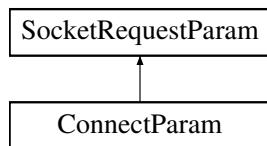
The documentation for this class was generated from the following file:

- `oscl_string_rep.h`

7.16 ConnectParam Class Reference

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

Inheritance diagram for ConnectParam::



Public Methods

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

Data Fields

- [OsclNetworkAddress iAddr](#)

7.16.1 Constructor & Destructor Documentation

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

7.16.2 Field Documentation

7.16.2.1 OsclNetworkAddress ConnectParam::iAddr

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

- [oscl_socket_request.h](#)

7.17 CStackRep Class Reference

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

Public Methods

- [CStackRep \(\)](#)
- [OSCL_IMPORT_REF void set \(const char *cp, uint32 len\)](#)
- [OSCL_IMPORT_REF void set \(const oscl_wchar *cp, uint32 len\)](#)
- [OSCL_IMPORT_REF void append \(const char *cp, uint32 len\)](#)
- [OSCL_IMPORT_REF void append \(const oscl_wchar *cp, uint32 len\)](#)

Data Fields

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

7.17.1 Detailed Description

For internal use only– stack string representation

7.17.2 Constructor & Destructor Documentation

7.17.2.1 CStackRep::CStackRep () [inline]

7.17.3 Member Function Documentation

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

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

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

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

7.17.4 Field Documentation

7.17.4.1 OsclAny* CStackRep::buffer

7.17.4.2 uint32 CStackRep::maxsize

7.17.4.3 uint32 CStackRep::size

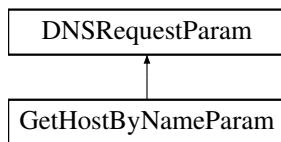
The documentation for this class was generated from the following file:

- [oscl_string_rep.h](#)

7.18 DNSRequestParam Class Reference

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

Inheritance diagram for DNSRequestParam::



Public Methods

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

Data Fields

- TPVDNSFx_n iFx_n
- OsclDNSRequest * iDNSRequest

Protected Methods

- DNSRequestParam (TPVDNSFx_n aFx_n)

Protected Attributes

- uint32 iRefCount

7.18.1 Constructor & Destructor Documentation

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

7.18.1.2 DNSRequestParam::DNSRequestParam (TPVDNSFx_n aFx_n) [protected]

7.18.2 Member Function Documentation

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

Implemented in [GetHostByNameParam](#).

7.18.2.2 void DNSRequestParam::InThread ()

7.18.2.3 void DNSRequestParam::RemoveRef ()

7.18.3 Field Documentation

7.18.3.1 OsclDNSRequest* DNSRequestParam::iDNSRequest

7.18.3.2 TPVDNSFxn DNSRequestParam::iFxn

7.18.3.3 uint32 DNSRequestParam::iRefCount [protected]

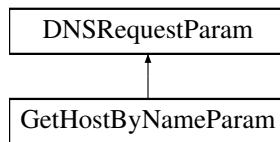
The documentation for this class was generated from the following file:

- [oscl_dns_param.h](#)

7.19 GetHostByNameParam Class Reference

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

Inheritance diagram for GetHostByNameParam::



Public Methods

- void [Destroy \(\)](#)
- [~GetHostByNameParam \(\)](#)

Static Public Methods

- [GetHostByNameParam * Create \(const char *name, OsclNetworkAddress *&addr\)](#)

Data Fields

- [char * iName](#)
- [OsclNetworkAddress * iAddr](#)

7.19.1 Constructor & Destructor Documentation

7.19.1.1 GetHostByNameParam::~GetHostByNameParam ()

7.19.2 Member Function Documentation

7.19.2.1 GetHostByNameParam* GetHostByNameParam::Create (const char * *name*, OsclNetworkAddress *& *addr*) [static]

7.19.2.2 void GetHostByNameParam::Destroy () [virtual]

Implements [DNSRequestParam](#).

7.19.3 Field Documentation

7.19.3.1 OsclNetworkAddress* GetHostByNameParam::iAddr

7.19.3.2 char* GetHostByNameParam::iName

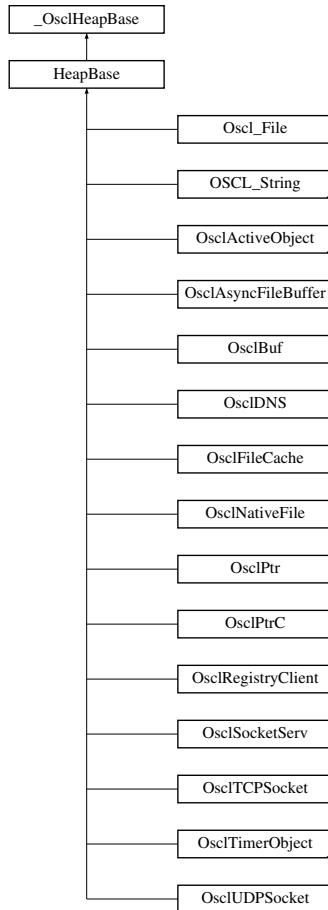
The documentation for this class was generated from the following file:

- [oscl_dns_param.h](#)

7.20 HeapBase Class Reference

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

Inheritance diagram for HeapBase::



Public Methods

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

7.20.1 Detailed Description

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

HeapBase has overloaded new and delete operators.

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

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

7.20.2 Constructor & Destructor Documentation

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

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

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

- [oscl_mem.h](#)

7.21 internalLeave Class Reference

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

Data Fields

- int a

7.21.1 Field Documentation

7.21.1.1 int internalLeave::a

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

- [oscl_error_imp_cppexceptions.h](#)

7.22 LinkedListElement< LLClass > Class Template Reference

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

Public Methods

- [LinkedListElement \(LLClass in_data\)](#)

Data Fields

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

7.22.1 Detailed Description

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

Linked List Element Class

7.22.2 Constructor & Destructor Documentation

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

7.22.3 Field Documentation

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

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

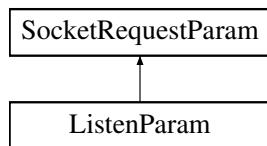
The documentation for this class was generated from the following file:

- [oscl_linked_list.h](#)

7.23 ListenParam Class Reference

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

Inheritance diagram for ListenParam::



Public Methods

- [ListenParam \(uint32 aSize\)](#)

Data Fields

- uint32 [iQSize](#)

7.23.1 Constructor & Destructor Documentation

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

7.23.2 Field Documentation

7.23.2.1 [uint32 ListenParam::iQSize](#)

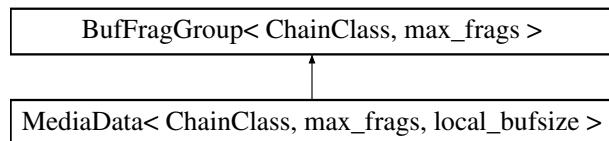
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.24 MediaData< ChainClass, max_frags, local_bufsize > Class Template Reference

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

Inheritance diagram for MediaData< ChainClass, max_frags, local_bufsize >::



Public Methods

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

Protected Methods

- [MediaStatusClass::status_t AddLocalFragment \(const BufferFragment &frag, int32 location_offset\)](#)

Protected Attributes

- [MediaTimestamp timestamp](#)
- uint8 [localbuf \[local_bufsize\]](#)
- uint32 [available_localbuf](#)
- int [num_reserved_fragments](#)

template<class ChainClass, uint32 max_frags, uint32 local_bufsize> class MediaData< ChainClass, max_frags, local_bufsize >

7.24.1 Constructor & Destructor Documentation

- 7.24.1.1 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> MediaData< ChainClass, max_frags, local_bufsize >::MediaData () [inline]
- 7.24.1.2 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> virtual MediaData< ChainClass, max_frags, local_bufsize >::~MediaData () [inline, virtual]

7.24.2 Member Function Documentation

- 7.24.2.1 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> MediaStatusClass::status_t MediaData< ChainClass, max_frags, local_bufsize >::AddLocalFragment (const BufferFragment &frag, int32 location_offset) [inline, protected]
- 7.24.2.2 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> virtual void MediaData< ChainClass, max_frags, local_bufsize >::Clear () [inline, virtual]

Reimplemented from [BufFragGroup< ChainClass, max_frags >](#).

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

7.24.3 Field Documentation

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

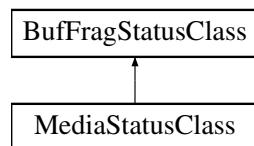
The documentation for this class was generated from the following file:

- [oscl_media_data.h](#)

7.25 MediaStatusClass Class Reference

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

Inheritance diagram for MediaStatusClass::



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

- [oscl_media_status.h](#)

7.26 MemAllocator< T > Class Template Reference

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

Public Types

- `typedef T * pointer`

Public Methods

- `virtual pointer allocate (void *hint=0, const int num_reserved_frags=1)=0`
- `virtual void deallocate (pointer p)=0`
- `virtual ~MemAllocator ()`

```
template<class T> class MemAllocator< T >
```

7.26.1 Member Typedef Documentation

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

7.26.2 Constructor & Destructor Documentation

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

7.26.3 Member Function Documentation

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

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

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

- [oscl_media_data.h](#)

7.27 MM_AllocBlockFence Struct Reference

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

Public Methods

- [MM_AllocBlockFence \(\)](#)
- [void fill_fence \(\)](#)
- [bool check_fence \(\)](#)

Data Fields

- [uint8 pad \[COMPUTE_MEM_ALIGN_SIZE\(sizeof\(MM_AllocBlockHdr\), MIN_FENCE_SIZE, MEM_ALIGN_SIZE\)\]](#)

7.27.1 Constructor & Destructor Documentation

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

7.27.2 Member Function Documentation

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

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

7.27.3 Field Documentation

[7.27.3.1 uint8 MM_AllocBlockFence::pad\[COMPUTE_MEM_ALIGN_SIZE\(sizeof\(MM_AllocBlockHdr\), MIN_FENCE_SIZE, MEM_ALIGN_SIZE\)\]](#)

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

- [oscl_mem_audit_internals.h](#)

7.28 MM_AllocBlockHdr Struct Reference

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

Public Methods

- bool [isAllocNodePtr \(\)](#)
- void [setAllocNodeFlag \(\)](#)
- [MM_AllocBlockHdr \(\)](#)
- [MM_AllocBlockHdr \(void *ptr, uint32 inSize\)](#)

Data Fields

- void * [pNode](#)
- uint32 [size](#)
- void * [pRootNode](#)
- uint32 [pad](#)

Static Public Attributes

- const uint32 [ALLOC_NODE_FLAG](#) = 0x80000000

7.28.1 Constructor & Destructor Documentation

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

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

7.28.2 Member Function Documentation

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

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

7.28.3 Field Documentation

[7.28.3.1 uint32 MM_AllocBlockHdr::pad](#)

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

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

[7.28.3.4 uint32 MM_AllocBlockHdr::size](#)

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

- [oscl_mem_audit_internals.h](#)

7.29 MM_AllocInfo Struct Reference

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

Public Methods

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

Data Fields

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

7.29.1 Constructor & Destructor Documentation

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

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

7.29.2 Member Function Documentation

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

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

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

7.29.3 Field Documentation

7.29.3.1 `uint32 MM_AllocInfo::allocNum`

7.29.3.2 `bool MM_AllocInfo::bSetFailure`

7.29.3.3 `uint32 MM_AllocInfo::lineNo`

7.29.3.4 `char* MM_AllocInfo::pFileName`

7.29.3.5 `void* MM_AllocInfo::pMemBlock`

7.29.3.6 `OsclMemStatsNode* MM_AllocInfo::pStatsNode`

7.29.3.7 `uint32 MM_AllocInfo::size`

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

- [oscl_mem_audit.h](#)

7.30 MM_AllocNode Struct Reference

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

Public Methods

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

Data Fields

- [MM_AllocInfo * pAllocInfo](#)
- [MM_AllocNode * pPrev](#)
- [MM_AllocNode * pNext](#)

7.30.1 Constructor & Destructor Documentation

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

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

7.30.2 Member Function Documentation

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

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

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

7.30.3 Field Documentation

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

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

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

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

- [oscl_mem_audit.h](#)

7.31 MM_AllocQueryInfo Struct Reference

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

Data Fields

- uint32 [allocNum](#)
- char [fileName](#) [MM_ALLOC_MAX_QUERY_FILENAME_LEN]
- uint32 [lineNo](#)
- uint32 [size](#)
- const void * [pMemBlock](#)
- char [tag](#) [MM_ALLOC_MAX_QUERY_TAG_LEN]

7.31.1 Field Documentation

7.31.1.1 uint32 MM_AllocQueryInfo::allocNum

7.31.1.2 char MM_AllocQueryInfo::fileName[MM_ALLOC_MAX_QUERY_FILENAME_LEN]

7.31.1.3 uint32 MM_AllocQueryInfo::lineNo

7.31.1.4 const void* MM_AllocQueryInfo::pMemBlock

7.31.1.5 uint32 MM_AllocQueryInfo::size

7.31.1.6 char MM_AllocQueryInfo::tag[MM_ALLOC_MAX_QUERY_TAG_LEN]

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

- [oscl_mem_audit.h](#)

7.32 MM_Audit_Imp Class Reference

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

Public Methods

- [MM_Audit_Imp \(\)](#)
- [~MM_Audit_Imp \(\)](#)
- [OSCL_IMPORT_REF void * MM_allocate \(const OsclMemStatsNode *statsNode, uint32 sizeIn, const char *pFileName, uint32 lineNumber, bool allocNodeTracking=false\)](#)
- [OSCL_IMPORT_REF bool MM_deallocate \(void *pMemBlockIn\)](#)
- [OSCL_IMPORT_REF MM_Stats_t * MM_GetStats \(const char *const tagIn\)](#)
- [OSCL_IMPORT_REF uint32 MM_GetStatsInDepth \(const char *tagIn, MM_Stats_CB *array_ptr, uint32 max_nodes\)](#)
- [OSCL_IMPORT_REF uint32 MM_GetTreeNodes \(const char *tagIn\)](#)
- [OSCL_IMPORT_REF bool MM_AddTag \(const char *tagIn\)](#)
- [OSCL_IMPORT_REF const OsclMemStatsNode * MM_GetTagName \(const char *tagIn\)](#)
- [OSCL_IMPORT_REF const OsclMemStatsNode * MM_GetExistingTag \(const char *tagIn\)](#)
- [OSCL_IMPORT_REF const OsclMemStatsNode * MM_GetRootNode \(\)](#)
- [OSCL_IMPORT_REF MM_AllocQueryInfo * MM_CreateAllocNodeInfo \(uint32 max_array_size\)](#)
- [OSCL_IMPORT_REF void MM_ReleaseAllocNodeInfo \(MM_AllocQueryInfo *info\)](#)
- [OSCL_IMPORT_REF uint32 MM_GetAllocNodeInfo \(MM_AllocQueryInfo *output_array, uint32 max_array_size, uint32 offset\)](#)
- [OSCL_IMPORT_REF bool MM_Validate \(const void *ptrIn\)](#)
- [uint32 MM_GetAllocNo \(void\)](#)
- [void MM_GetOverheadStats \(MM_AuditOverheadStats &stats\)](#)
- [uint32 MM_GetNumAllocNodes \(\)](#)
- [uint32 MM_GetMode \(void\)](#)
- [uint8 MM_GetPrefillPattern \(void\)](#)
- [uint32 MM_GetPostfillPattern \(void\)](#)
- [OSCL_IMPORT_REF void MM_SetMode \(uint32 inMode\)](#)
- [OSCL_IMPORT_REF void MM_SetPrefillPattern \(uint8 pattern\)](#)
- [OSCL_IMPORT_REF void MM_SetPostfillPattern \(uint8 pattern\)](#)
- [OSCL_IMPORT_REF void MM_SetTagLevel \(uint32 level\)](#)
- [OSCL_IMPORT_REF bool MM_SetFailurePoint \(const char *tagIn, uint32 alloc_number\)](#)
- [OSCL_IMPORT_REF void MM_UnsetFailurePoint \(const char *tagIn\)](#)
- [MM_AllocNode * addAllocNode \(void *pMem, uint32 sizeIn, OsclMemStatsNode *pStatsNode, const char *pFileName, uint32 lineNumber\)](#)
- [OsclMemStatsNode * removeAllocNode \(void *pMemBlockIn, uint32 &size\)](#)
- [void removeALLAllocNodes \(\)](#)
- [OsclMemStatsNode * createStatsNode \(const char *tagIn\)](#)
- [bool updateStatsNode \(OsclMemStatsNode *pCurrStatsNode, const MM_Stats_t &pDelta, bool bAdd\)](#)
- [bool updateStatsNodeInFailure \(const char *tagIn\)](#)
- [bool updateStatsNodeInFailure \(OsclMemStatsNode *pStatsNode\)](#)
- [bool pruneSubtree \(OsclMemStatsNode *pNode\)](#)
- [bool pruneSubtree \(const char *tagIn\)](#)
- [void retrieveParentTag \(char *tag\)](#)
- [int32 retrieveParentTagLength \(const char *tag, int32 bound\)](#)
- [void makeValidTag \(const char *tagIn, MMAuditCharAutoPtr &autoPtr\)](#)

- uint32 [getTagActualSize](#) (const char *tagIn)
- bool [isSetFailure](#) (const char *tagIn)
- bool [isSetFailure](#) (OsclMemStatsNode *statsNode)
- bool [validate_all_heap](#) ()

Static Public Methods

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

7.32.1 Constructor & Destructor Documentation

7.32.1.1 MM_Audit_Imp::MM_Audit_Imp ()

Constructor, create the root node in statistics table

7.32.1.2 MM_Audit_Imp::~MM_Audit_Imp ()

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

7.32.2 Member Function Documentation

7.32.2.1 MM_AllocNode* MM_Audit_Imp::addAllocNode (void * pMem, uint32 sizeIn, OsclMemStatsNode * pStatsNode, const char * pFileName, uint32 lineNumber)

Returns:

true if operation succeeds;

7.32.2.2 OsclMemStatsNode* MM_Audit_Imp::createStatsNode (const char * tagIn)

Returns:

true if operation succeeds;

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

Returns:

audit root pointer.

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

Returns:

original block size. leaves if bad pointer.

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

the size of the truncated tag; 0 means NO truncation

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

true if operation succeeds;

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

a valid tag; NULL will be converted into root tag

7.32.2.9 OSCL_IMPORT_REF bool MM_Audit_Imp::MM_AddTag (const char * tagIn) [inline]

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

Parameters:

tagIn input tag

Returns:

true if operation succeeds;

7.32.2.10 OSCL_IMPORT_REF void* MM_Audit_Imp::MM_allocate (const OsclMemStatsNode * statsNode, uint32 sizeIn, const char * pFileName, uint32 lineNumber, bool allocNodeTracking = false)

The following are APIs t __nothrow_ / const __nothrow_

Returns:

the memory pointer if operation succeeds.

7.32.2.11 OSCL_IMPORT_REF MM_AllocQueryInfo* MM_Audit_Imp::MM_CreateAllocNode-Info (uint32 max_array_size)

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

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

true if operation succeeds;

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

API to get the current allocation number

Returns:

the current allocation number

**7.32.2.14 OSCL_IMPORT_REF uint32 MM_Audit_Imp::MM_GetAllocNodeInfo
(MM_AllocQueryInfo * output_array, uint32 max_array_size, uint32 offset)**

API to query the list of alloc nodes. It copies the information into the provided output array.

Parameters:

output_array the array where the data will be written

max_array_size the max number of output array elements

offset the offset into the alloc node list from which the data should begin.

Returns:

the number of valid nodes in the output array

**7.32.2.15 OSCL_IMPORT_REF const OsclMemStatsNode* MM_Audit_Imp::MM_GetExisting-
Tag (const char * tagIn)**

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

Parameters:

tagIn input tag

Returns:

true if operation succeeds;

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

API to get the operating mode of the mm_audit class.

7.32.2.17 uint32 MM_Audit_Imp::MM_GetNumAllocNodes () [inline]

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

**7.32.2.18 void MM_Audit_Imp::MM_GetOverheadStats (MM_AuditOverheadStats & stats)
[inline]**

API to get the overhead statistics for the memory used by the mm_audit class.

7.32.2.19 uint32 MM_Audit_Imp::MM_GetPostfillPattern (void) [inline]

API to get the postfill pattern. The pattern is used to fill the memory before freeing it.

7.32.2.20 uint8 MM_Audit_Imp::MM_GetPrefillPattern (void) [inline]

API to get the prefill pattern. The pattern is used to fill the memory before returning it to the caller.

7.32.2.21 OSCL_IMPORT_REF const OsclMemStatsNode* MM_Audit_Imp::MM_GetRootNode () [inline]**7.32.2.22 OSCL_IMPORT_REF MM_Stats_t* MM_Audit_Imp::MM_GetStats (const char *const tagIn)**

API to get memory statistics through context string(tag)

Returns:

statistics pointer if operation succeeds

7.32.2.23 OSCL_IMPORT_REF uint32 MM_Audit_Imp::MM_GetStatsInDepth (const char * tagIn, MM_Stats_CB * array_ptr, uint32 max_nodes)

API to get memory statistics in detail through context string(tag) including its subtree

Returns:

statistics pointer array and actual number of nodes if operation succeeds

7.32.2.24 OSCL_IMPORT_REF const OsclMemStatsNode* MM_Audit_Imp::MM_GetTagName (const char * tagIn)

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

Parameters:

tagIn input tag

Returns:

pointer to [OsclMemStatsNode](#) which should be passed to MM_allocate

7.32.2.25 OSCL_IMPORT_REF uint32 MM_Audit_Imp::MM_GetTreeNodes (const char * tagIn)

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

Parameters:

tagIn input tag

Returns:

the number of tree nodes ; 0 means no tag node

**7.32.2.26 OSCL_IMPORT_REF void MM_Audit_Imp::MM_ReleaseAllocNodeInfo
([MM_AllocQueryInfo](#) * *info*)**

**7.32.2.27 OSCL_IMPORT_REF bool MM_Audit_Imp::MM_SetFailurePoint (const char * *tagIn*,
uint32 *alloc_number*)**

API to insert allocation failure deterministically according to allocation number associated with tag

Parameters:

tagIn input tag

alloc_number allocation number associated with tag

Returns:

true if operation succeeds;

7.32.2.28 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetMode (uint32 *inMode*)

API to set the operating mode of the mm_audit class.

7.32.2.29 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetPostfillPattern (uint8 *pattern*)

API to set the postfill pattern.

7.32.2.30 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetPrefillPattern (uint8 *pattern*)

API to set the prefill pattern.

7.32.2.31 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetTagLevel (uint32 *level*)

API to set the maximum tag level,i.e. tag level for a.b.c.d = 4

Parameters:

level input tag level to be set

**7.32.2.32 OSCL_IMPORT_REF void MM_Audit_Imp::MM_UnsetFailurePoint (const char *
tagIn)**

API to cancel the allocation failure point associated with tag

Parameters:

tagIn input tag

7.32.2.33 OSCL_IMPORT_REF bool MM_Audit_Imp::MM_Validate (const void * *ptrIn*)

API to check the input pointer is a valid pointer to a chunk of memory

Parameters:

ptrIn input pointer to be validated

Returns:

true if operation succeeds;

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

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

Returns:

true if operation succeeds;

7.32.2.36 `void MM_Audit_Imp::removeALLAllocNodes ()`

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

Returns:

true if operation succeeds;

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

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

Returns:

the length of a immediate parent tag for the input tag

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

Returns:

true if operation succeeds;

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

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

Returns:

true if operation succeeds;

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

Returns:

true if operation succeeds;

7.32.2.44 bool MM_Audit_Imp::validate_all_heap ()**Returns:**

true if operation succeeds;

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

- [oscl_mem_audit.h](#)

7.33 MM_AuditOverheadStats Struct Reference

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

Data Fields

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

7.33.1 Field Documentation

7.33.1.1 uint32 MM_AuditOverheadStats::per_allocation_overhead

7.33.1.2 uint32 MM_AuditOverheadStats::stats_overhead

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

- [oscl_mem_audit.h](#)

7.34 MM_FailInsertParam Struct Reference

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

Public Methods

- [MM_FailInsertParam \(\)](#)
- [void reset \(\)](#)
- [void * operator new \(oscl_memsize_t size\)](#)
- [void * operator new \(oscl_memsize_t size, MM_FailInsertParam *ptr\)](#)
- [void operator delete \(void *ptr\) throw \(\)](#)

Data Fields

- [uint32 nAllocNum](#)
- [uint16 xsubi \[3\]](#)

7.34.1 Constructor & Destructor Documentation

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

7.34.2 Member Function Documentation

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

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

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

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

7.34.3 Field Documentation

[7.34.3.1 uint32 MM_FailInsertParam::nAllocNum](#)

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

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

- [oscl_mem_audit.h](#)

7.35 MM_Stats_CB Struct Reference

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

Public Methods

- [MM_Stats_CB \(\)](#)
- [void * operator new \(oscl_memsize_t size\)](#)
- [void * operator new \(oscl_memsize_t size, MM_Stats_CB *ptr\)](#)
- [void operator delete \(void *ptr\) throw \(\)](#)

Data Fields

- [const char * tag](#)
- [const MM_Stats_t * pStats](#)
- [uint32 num_child_nodes](#)

7.35.1 Constructor & Destructor Documentation

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

7.35.2 Member Function Documentation

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

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

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

7.35.3 Field Documentation

[7.35.3.1 uint32 MM_Stats_CB::num_child_nodes](#)

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

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

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

- [oscl_mem_audit.h](#)

7.36 MM_Stats_t Struct Reference

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

Public Methods

- [MM_Stats_t \(\)](#)
- [MM_Stats_t \(uint32 sizeIn\)](#)
- [void reset \(\)](#)
- [void update \(const MM_Stats_t &delta, bool add\)](#)
- [void * operator new \(oscl_memsize_t size\)](#)
- [void * operator new \(oscl_memsize_t size, MM_Stats_t *ptr\)](#)
- [void operator delete \(void *ptr\) throw \(\)](#)

Data Fields

- [uint32 numBytes](#)
- [uint32 peakNumBytes](#)
- [uint32 numAllocs](#)
- [uint32 peakNumAllocs](#)
- [uint32 numAllocFails](#)
- [uint32 totalNumAllocs](#)
- [uint32 totalNumBytes](#)

7.36.1 Constructor & Destructor Documentation

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

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

7.36.2 Member Function Documentation

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

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

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

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

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

7.36.3 Field Documentation

7.36.3.1 `uint32 MM_Stats_t::numAllocFails`

7.36.3.2 `uint32 MM_Stats_t::numAllocs`

7.36.3.3 `uint32 MM_Stats_t::numBytes`

7.36.3.4 `uint32 MM_Stats_t::peakNumAllocs`

7.36.3.5 `uint32 MM_Stats_t::peakNumBytes`

7.36.3.6 `uint32 MM_Stats_t::totalNumAllocs`

7.36.3.7 `uint32 MM_Stats_t::totalNumBytes`

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

- [oscl_mem_audit.h](#)

7.37 NTPTime Class Reference

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

```
#include <oscl_time.h>
```

Public Methods

- **OSCL_COND_IMPORT_REF NTPTime ()**
The default constructor creates an NTPTime instance representing the current system time.
- **OSCL_COND_IMPORT_REF NTPTime (const NTPTime &src)**
Copy constructor to create a new NTPTime from an existing one.
- **OSCL_COND_IMPORT_REF NTPTime (const uint32 seconds)**
Construct an NTPTime from a uint32.
- **OSCL_COND_IMPORT_REF NTPTime (const int32 seconds)**
Construct an NTPTime from a int.
- **OSCL_COND_IMPORT_REF NTPTime (const TimeValue &t)**
Construct a NTPTime instance from a TimeValue instance.
- **OSCL_COND_IMPORT_REF NTPTime (const uint64 value)**
Construct a NTPTime instance from a uint64 value.
- **OSCL_COND_IMPORT_REF NTPTime & operator= (uint32 newval)**
The assignment operator for a 32 bit integer.
- **OSCL_COND_IMPORT_REF NTPTime & operator= (uint64 newval)**
The assignment operator for a 64 bit integer.
- **OSCL_COND_IMPORT_REF NTPTime & operator+= (uint64 val)**
The += operator is used to add a 64 bit 32.32 value to an existing NTPTime value.
- **OSCL_COND_IMPORT_REF NTPTime operator- (const NTPTime &npt) const**
The - operator allows subtraction of one NTPTime value from another. This is useful to measure an interval.
- **void set_from_system_time (const uint32 systemtime)**
This method converts a 32-bit system time to NTP time.
- **OSCL_COND_IMPORT_REF uint32 get_middle32 () const**
Grab the middle 32 bits of the 64 bit 32.32 representation.
- **OSCL_COND_IMPORT_REF uint32 get_upper32 () const**
This method returns the upper 32 bits of the 32.32 representation.
- **OSCL_COND_IMPORT_REF uint32 get_lower32 () const**
This method returns the lower 32 bits of the 32.32 representation.

- int32 [to_system_time \(\) const](#)

This method converts the ntp time value to system time.

- OSCL_COND_IMPORT_REF [uint64 get_value \(\) const](#)

This method returns the 32.32 ntp representation.

- OSCL_IMPORT_REF int [set_to_current_time \(\)](#)

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

7.37.1 Detailed Description

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

The NTPTime class: Conversion to/from Unix (epoch at 0h Jan. 1, 1970) amount to addition/subtraction of 2208988800. A single 64 bit value is used to represent the time. This value represents the number of seconds since 0h (UTC) Jan. 1, 1900. There is an implied binary point between the upper 32 bits and lower 32 bits (this is referred to as a 32.32 fractional representation). For example a binary value of 00000000 00000000 00000011 10000000 00000000 00000000 00000000 represents 3.5 seconds since Jan 1, 1900.

7.37.2 Constructor & Destructor Documentation

7.37.2.1 OSCL_COND_IMPORT_REF NTPTime::NTPTime ()

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

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

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

7.37.2.3 OSCL_COND_IMPORT_REF NTPTime::NTPTime (const uint32 seconds)

Construct an NTPTime from a uint32.

Parameters:

seconds The uint32 input represents the number of seconds since Jan. 1, 1900.

7.37.2.4 OSCL_COND_IMPORT_REF NTPTime::NTPTime (const int32 seconds)

Construct an NTPTime from a int.

Parameters:

seconds The int input represents the number of seconds since Jan. 1, 1900.

7.37.2.5 OSCL_COND_IMPORT_REF NTPTime::NTPTime (const TimeValue & t)

Construct a NTPTime instance from a TimeValue instance.

This constructor creates an NTPTime value representing the same absolute time as the TimeValue parameter.

Parameters:

t A reference to a TimeValue object.

7.37.2.6 OSCL_COND_IMPORT_REF NTPTime::NTPTime (const uint64 value)

Construct a NTPTime instance from a uint64 value.

Parameters:

value A 64 bit integer argument which is used as the ntp 32.32 fractional representation.

7.37.3 Member Function Documentation

7.37.3.1 OSCL_COND_IMPORT_REF uint32 NTPTime::get_lower32 ()

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

7.37.3.2 OSCL_COND_IMPORT_REF uint32 NTPTime::get_middle32 ()

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

7.37.3.3 OSCL_COND_IMPORT_REF uint32 NTPTime::get_upper32 ()

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

7.37.3.4 OSCL_COND_IMPORT_REF uint64 NTPTime::get_value ()

This method returns the 32.32 ntp representation.

7.37.3.5 OSCL_COND_IMPORT_REF NTPTime& NTPTime::operator+= (uint64 val)

The += operator is used to add a 64 bit 32.32 value to an existing NTPTime value.

Parameters:

val The 64 bit 32.32 value to add to this object's value.

7.37.3.6 OSCL_COND_IMPORT_REF NTPTime NTPTime::operator- (const NTPTime & npt) const

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

Parameters:

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

7.37.3.7 OSCL_COND_IMPORT_REF NTPTime& NTPTime::operator= (uint64 newval)

The assignment operator for a 64 bit integer.

Parameters:

newval A 64 bit value which represents the 32.32 fractional representation of the ntp time.

7.37.3.8 OSCL_COND_IMPORT_REF NTPTime& NTPTime::operator= (uint32 newval)

The assignment operator for a 32 bit integer.

Parameters:

newval A 32 bit integer representing the upper 32 bits of the 32.32 NTP time (e.g. the number of whole seconds since Jan 1, 1900 UTC).

7.37.3.9 void NTPTime::set_from_system_time (const uint32 systemtime)

This method converts a 32-bit system time to NTP time.

This method sets the value of the NTPTime instance to the absolute time represented by the 32 bit input argument.

Parameters:

systemtime This 32-bit value is interpreted as the number of seconds since the unix epoch Jan. 1 1970.

7.37.3.10 OSCL_IMPORT_REF int NTPTime::set_to_current_time ()

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

7.37.3.11 int32 NTPTime::to_system_time ()

This method converts the ntp time value to system time.

This method returns a 32 bit value representing the same absolute time as the NTP time value, but expressed as whole seconds since the unix epoch. The fractional part of the ntp value is discarded.

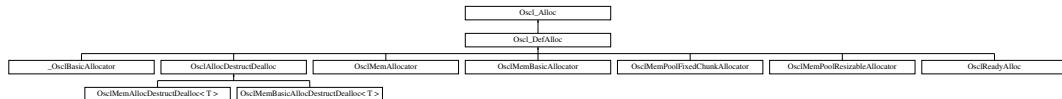
The documentation for this class was generated from the following file:

- [oscl_time.h](#)

7.38 Oscl_Alloc Class Reference

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

Inheritance diagram for Oscl_Alloc::



Public Methods

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

7.38.1 Member Function Documentation

7.38.1.1 virtual [OsclAny* Oscl_Alloc::allocate \(const uint32 size\)](#) [pure virtual]

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

7.38.1.2 virtual [OsclAny* Oscl_Alloc::allocate_fl \(const uint32 size, const char *file_name, const int line_num\)](#) [inline, virtual]

Reimplemented in [Oscl_DefAlloc](#), [OsclMemAllocator](#), [OsclMemAllocDestructDealloc< T >](#), and [OsclReadyAlloc](#).

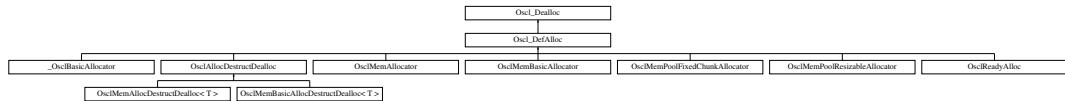
The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

7.39 Oscl_Dealloc Class Reference

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

Inheritance diagram for Oscl_Dealloc::



Public Methods

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

7.39.1 Member Function Documentation

7.39.1.1 virtual void Oscl_Dealloc::deallocate (OsclAny *p) [pure virtual]

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

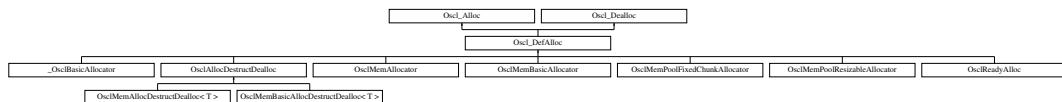
The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

7.40 Oscl_DefAlloc Class Reference

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

Inheritance diagram for Oscl_DefAlloc::



Public Methods

- virtual [OsclAny * allocate \(const uint32 size\)=0](#)
- virtual [OsclAny * allocate_fl \(const uint32 size, const char *file_name, const int line_num\)](#)
- virtual void [deallocate \(OsclAny *p\)=0](#)

7.40.1 Member Function Documentation

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

Implements [Oscl_Alloc](#).

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

7.40.1.2 virtual [OsclAny* Oscl_DefAlloc::allocate_fl \(const uint32 size, const char *file_name, const int line_num\)](#) [inline, virtual]

Reimplemented from [Oscl_Alloc](#).

Reimplemented in [OsclMemAllocator](#), [OsclMemAllocDestructDealloc< T >](#), and [OsclReadyAlloc](#).

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

Implements [Oscl_Dealloc](#).

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

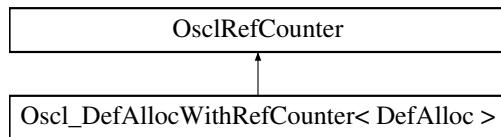
The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

7.41 Oscl_DefAllocWithRefCounter< DefAlloc > Class Template Reference

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

Inheritance diagram for Oscl_DefAllocWithRefCounter< DefAlloc >::



Public Methods

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

Static Public Methods

- Oscl_DefAllocWithRefCounter * [New](#) ()

7.41.1 Detailed Description

template<class DefAlloc> class Oscl_DefAllocWithRefCounter< DefAlloc >

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

7.41.2 Member Function Documentation

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

Add to the reference count

Implements [OsclRefCounter](#).

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

Delete object

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

Gets the current number of references

Implements [OsclRefCounter](#).

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

Create object

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

Delete from reference count

Implements [OsclRefCounter](#).

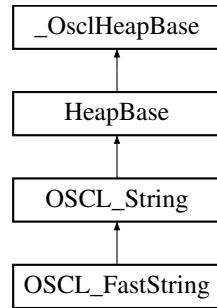
The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

7.42 OSCL_FastString Class Reference

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

Inheritance diagram for OSCL_FastString::



Public Types

- `typedef OSCL_String::chartype chartype`
- `typedef TOSCL_StringOp optype`
- `typedef OSCL_wString::chartype other_chartype`

Public Methods

- `OSCL_IMPORT_REF OSCL_FastString()`
- `OSCL_IMPORT_REF OSCL_FastString(const OSCL_FastString &src)`
- `OSCL_IMPORT_REF OSCL_FastString(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_FastString(chartype *buf, uint32 maxlen)`
- `OSCL_IMPORT_REF ~OSCL_FastString()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_FastString & operator=(const OSCL_FastString &src)`
- `OSCL_IMPORT_REF OSCL_FastString & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(chartype *cstr, uint32 maxlen)`
- `OSCL_IMPORT_REF void set(const other_chartype *buf, uint32 numofbyte, optype op)`
- `OSCL_IMPORT_REF void set_length()`

Friends

- class `OSCL_String`

7.42.1 Detailed Description

`OSCL_FastString` is a simple string class, compatible with regular character array strings.

This class does not allocate internal memory for the string but acts as a container for a user-defined buffer. This means no copying of the string is done and provides a faster way of manipulating strings. Depending on initialization, this container provides either read-only or read-write access to the string.

Implementation assumes the input string is null-terminated.

Parameters:

C: type of character.

7.42.2 Member Typedef Documentation

7.42.2.1 `typedef OSCL_String::chartype OSCL_FastString::chartype`

Reimplemented from [OSCL_String](#).

7.42.2.2 `typedef TOSCL_StringOp OSCL_FastString::optype`

7.42.2.3 `typedef OSCL_wString::chartype OSCL_FastString::other_chartype`

7.42.3 Constructor & Destructor Documentation

7.42.3.1 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString()`

Default constructor.

7.42.3.2 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(const OSCL_FastString &src)`

Creates a fast string that contains a copy of the input string. The string inherits the writable-ness of the source string.

Parameters:

src: input string.

7.42.3.3 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(const chartype *cstr)`

Create the string and initialize it to contain the input string. The string is not writable.

am: **null-terminated string.**

7.42.3.4 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(chartype *buf, uint32 maxlen)`

Create the string and initialize it to contain the input string. The string is writable.

Parameters:

cp: null-terminated string.

maxlen: maximum size of storage at *cp*, not incl null terminator. If input string is not null-terminated, the function leaves.

7.42.3.5 OSCL_IMPORT_REF OSCL_FastString::~OSCL_FastString ()**7.42.4 Member Function Documentation****7.42.4.1 OSCL_IMPORT_REF const chartype* OSCL_FastString::get_cstr () [virtual]**

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

Implements [OSCL_String](#).

7.42.4.2 OSCL_IMPORT_REF uint32 OSCL_FastString::get_maxsize () [virtual]

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

Implements [OSCL_String](#).

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

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

7.42.4.4 OSCL_IMPORT_REF chartype* OSCL_FastString::get_str () [virtual]

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

Implements [OSCL_String](#).

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

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

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

Assignment operators

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

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

Parameters:

buf: string or character array.

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

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

7.42.4.8 OSCL_IMPORT_REF void OSCL_FastString::set ([chartype](#) * *cstr*, uint32 *maxlen*)

This function can be used to reassign the string to a new writable string. If input string is not null-terminated, the function leaves.

7.42.4.9 OSCL_IMPORT_REF void OSCL_FastString::set_length ()

This function can be used to refresh the string size in case the contents of the string buffer have been modified since the container was created.

7.42.5 Friends And Related Function Documentation

7.42.5.1 friend class OSCL_String [friend]

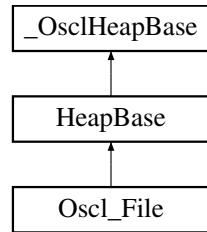
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.43 Oscl_File Class Reference

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

Inheritance diagram for Oscl_File::



Public Types

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

Public Methods

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

- OSCL_IMPORT_REF TOsclFileOffset Size ()
- OSCL_IMPORT_REF void SetLoggingEnable (bool aEnable)
- OSCL_IMPORT_REF void SetSummaryStatsLoggingEnable (bool aEnable)

Friends

- class OsclFileCache
- class OsclFileCacheBuffer
- class asynccfilereadwrite_test
- class largeasynccfilereadwrite_test
- class asynccfilereadcancel_test

7.43.1 Member Enumeration Documentation

7.43.1.1 enum Oscl_File::mode_type

Enumeration values:

MODE_READ Opens a file for reading. The file must exist.

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

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

MODE_BINARY Opens the file in 'binary' mode. This is the default.

MODE_TEXT Opens the file in 'text' mode. The default mode is 'binary'.

MODE_READ_PLUS Open a file for reading and writing. The file must exist. The default mode is 'binary'.

7.43.1.2 enum Oscl_File::seek_type

Enumeration values:

SEEKSET Beginning of file

SEEKCUR Current position of file pointer

SEEKEND End of file

7.43.1.3 enum Oscl_File::TSymbianAccessMode

Defines mode options for SetNativeAccessMode on Symbian.

Enumeration values:

ESymbianAccessMode_Rfile

ESymbianAccessMode_RfileBuf

7.43.2 Constructor & Destructor Documentation

7.43.2.1 OSCL_IMPORT_REF Oscl_File::Oscl_File ()

Constructor

7.43.2.2 OSCL_IMPORT_REF Oscl_File::Oscl_File (uint32 *aCacheSize*)

Deprecated Constructor, present for back-compatibility.

Parameters:

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

7.43.2.3 OSCL_IMPORT_REF Oscl_File::Oscl_File (uint32 *aCacheSize*, **OsclFileHandle** * *aFileHandle*)

Deprecated Constructor, present for back-compatibility.

Parameters:

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

aFileHandle: open file handle.

7.43.2.4 OSCL_IMPORT_REF Oscl_File::~Oscl_File ()

Destructor

7.43.3 Member Function Documentation

7.43.3.1 void Oscl_File::AddFixedCache (const **OsclFixedCacheParam** & *aParam*) [inline]

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

Parameters:

aParam: Cache location and size.

7.43.3.2 OSCL_IMPORT_REF int32 Oscl_File::Close ()

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

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

Returns:

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

7.43.3.3 OSCL_IMPORT_REF int32 Oscl_File::EndOfFile ()

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

Returns:

7.43.3.4 OSCL_IMPORT_REF int32 Oscl_File::Flush ()

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

Returns:

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

7.43.3.5 OSCL_IMPORT_REF int32 Oscl_File::GetError ()

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

Returns:

7.43.3.6 OsclFileHandle* Oscl_File::Handle () [inline]

Retrieve the file handle.

Returns:

file handle

7.43.3.7 OSCL_IMPORT_REF int32 Oscl_File::Open (const oscl_wchar *filename, uint32 mode, Oscl_FileServer &fileserv)

Opens a file.

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

Parameters:

filename name of file to open (Unicode)

mode combination of open mode flags

fileserv fileserv to use

Returns:

returns 0 if successful and a non-zero value otherwise

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

Opens a file.

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

Parameters:

filename name of file to open (Utf8)

mode combination of open mode flags

fileserv fileserv to use

Returns:

returns 0 if successful and a non-zero value otherwise

7.43.3.9 OSCL_IMPORT_REF uint32 Oscl_File::Read ([OsclAny * buffer](#), uint32 *size*, uint32 *numelements*)

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

Parameters:

buffer pointer to buffer of type void

size element size in bytes

numelements max number of elements to read

Returns:

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

7.43.3.10 void Oscl_File::RemoveFixedCache (const [TOsclFileOffset & aPos](#)) [inline]

RemoveFixedCache removes a fixed cache.

Parameters:

aPos: Cache location and size.

7.43.3.11 OSCL_IMPORT_REF int32 Oscl_File::Seek ([TOsclFileOffset offset](#), [seek_type origin](#))

The File Seek operation Sets the position for file pointer

Parameters:

offset offset from the specified origin.

origin starting point

Returns:

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

7.43.3.12 OSCL_IMPORT_REF void Oscl_File::SetAsyncReadBufferSize (uint32 *aSize*)

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

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

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

Parameters:

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

7.43.3.13 void Oscl_File::SetCacheObserver ([OsclCacheObserver](#) * *aObs*) [inline]

7.43.3.14 OSCL_IMPORT_REF int32 Oscl_File::SetFileHandle ([OsclFileHandle](#) * *aHandle*)

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

This call is not available when the Oscl_File object is already open.

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

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

Parameters:

aHandle: container for an open file handle.

Returns:

returns 0 if successful, non-zero if error.

7.43.3.15 OSCL_IMPORT_REF void Oscl_File::SetLoggingEnable (bool *aEnable*)

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

Parameters:

aEnable: true to enable, false to disable logging.

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

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

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

Parameters:

aMode: access mode.

7.43.3.17 OSCL_IMPORT_REF void Oscl_File::SetNativeBufferSize (int32 *aSize*)

SetNativeBufferSize configures the native file buffering feature, when available.

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

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

Parameters:

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

7.43.3.18 OSCL_IMPORT_REF void Oscl_File::SetPVCacheSize (uint32 *aSize*)

SetPVCacheSize configures the read/write cache.

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

Parameters:

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

7.43.3.19 OSCL_IMPORT_REF void Oscl_File::SetSummaryStatsLoggingEnable (bool *aEnable*)

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

Parameters:

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

7.43.3.20 OSCL_IMPORT_REF TOsclFileOffset Oscl_File::Size ()

Get the file size in bytes.

Returns:

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

7.43.3.21 OSCL_IMPORT_REF TOsclFileOffset Oscl_File::Tell ()

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

7.43.3.22 OSCL_IMPORT_REF uint32 Oscl_File::Write (const OsclAny * *buffer*, uint32 *size*, uint32 *numelements*)

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

Parameters:

buffer pointer to buffer of type void

size element size in bytes

numelements number of elements to write

Returns:

The number of elements written

7.43.4 Friends And Related Function Documentation

7.43.4.1 **friend class `asyncfilereadcancel_test` [friend]**

7.43.4.2 **friend class `asyncfilereadwrite_test` [friend]**

7.43.4.3 **friend class `largeasynccfilereadwrite_test` [friend]**

7.43.4.4 **friend class `OsclFileCache` [friend]**

7.43.4.5 **friend class `OsclFileCacheBuffer` [friend]**

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

- [oscl_file_io.h](#)

7.44 Oscl_File::OsclCacheObserver Class Reference

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

Public Methods

- virtual **OsclFileCacheBuffer * ChooseCurCache (OsclFileCache &aContext, TOsclFileOffset aPos)=0**

7.44.1 Detailed Description

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

7.44.2 Member Function Documentation

7.44.2.1 virtual **OsclFileCacheBuffer* Oscl_File::OsclCacheObserver::ChooseCurCache (OsclFileCache & aContext, TOsclFileOffset aPos) [pure virtual]**

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

- [oscl_file_io.h](#)

7.45 Oscl_File::OsclFixedCacheParam Class Reference

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

Public Methods

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

Data Fields

- [TOsclFileOffset iFilePosition](#)
- [uint32 iSize](#)

7.45.1 Detailed Description

Parameters for defining a fixed cache

7.45.2 Member Function Documentation

**7.45.2.1 bool Oscl_File::OsclFixedCacheParam::Contains (TOsclFileOffset *pos*) const
[inline]**

7.45.3 Field Documentation

7.45.3.1 TOsclFileOffset Oscl_File::OsclFixedCacheParam::iFilePosition

7.45.3.2 uint32 Oscl_File::OsclFixedCacheParam::iSize

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

- [oscl_file_io.h](#)

7.46 Oscl_FileFind Class Reference

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

Public Types

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

Public Methods

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

7.46.1 Detailed Description

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

7.46.2 Member Enumeration Documentation

7.46.2.1 enum Oscl_FileFind::element_type

Enumeration values:

`FILE_TYPE`

`DIR_TYPE`

`INVALID_TYPE`

7.46.2.2 enum Oscl_FileFind::error_type

Enumeration values:

`E_OK`

`E_INVALID_STATE`

`E_INVALID_ARG`

`E_PATH_TOO_LONG`

E_PATH_NOT_FOUND
E_NO_MATCH
E_BUFFER_TOO_SMALL
E_NOT_IMPLEMENTED
E_MEMORY_ERROR
E_OTHER

7.46.3 Constructor & Destructor Documentation

7.46.3.1 OSCL_IMPORT_REF Oscl_FileFind::Oscl_FileFind ()

constructor.

Returns:

none

7.46.3.2 OSCL_IMPORT_REF Oscl_FileFind::~Oscl_FileFind ()

destructor. will deallocate open handles if necessary

Returns:

none

7.46.4 Member Function Documentation

7.46.4.1 OSCL_IMPORT_REF void Oscl_FileFind::Close ()

closes the handle to directory.

Returns:

none

7.46.4.2 OSCL_IMPORT_REF const oscl_wchar* Oscl_FileFind::FindFirst (const oscl_wchar * directory, const oscl_wchar * pattern, oscl_wchar * buf, uint32 buflen)

Opens a directory for reading.

Parameters:

directory directory to search (utf16).

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

buf buffer for returned pathname (utf16).

buflen size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E_BUFFER_TOO_SMALL.

Returns:

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

7.46.4.3 OSCL_IMPORT_REF const char* Oscl_FileFind::FindFirst (const char * *directory*, const char * *pattern*, char * *buf*, uint32 *buflen*)

Finds first element matching the pattern.

Parameters:

directory directory to search (utf8).

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

buf buffer for returned pathname (utf8).

buflen size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E_BUFFER_TOO_SMALL.

Returns:

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

7.46.4.4 OSCL_IMPORT_REF oscl_wchar* Oscl_FileFind::FindNext (oscl_wchar * *buf*, uint32 *buflen*)

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

Parameters:

buf buffer to hold directory name(utf16)

buflen size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E_BUFFER_TOO_SMALL.

Returns:

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

7.46.4.5 OSCL_IMPORT_REF char* Oscl_FileFind::FindNext (char * *buf*, uint32 *buflen*)

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

Parameters:

buf buffer to hold directory name(utf8)

buflen size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E_BUFFER_TOO_SMALL.

Returns:

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

7.46.4.6 OSCL_IMPORT_REF element_type Oscl_FileFind::GetElementType ()

Returns the element type for the last element returned

Returns:

see enumeration above for more info.

7.46.4.7 OSCL_IMPORT_REF [error_type](#) Oscl_FileFind::GetLastError ()

Returns the error code for the last operation.

Returns:

see enumeration above for more info.

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

- [oscl_file_find.h](#)

7.47 Oscl_FileServer Class Reference

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

Public Methods

- OSCL_IMPORT_REF [Oscl_FileServer \(\)](#)
- OSCL_IMPORT_REF [~Oscl_FileServer \(\)](#)
- OSCL_IMPORT_REF int32 [Connect \(\)](#)
- OSCL_IMPORT_REF int32 [Close \(\)](#)
- OSCL_IMPORT_REF int32 [Oscl_DeleteFile \(const char *filename\)](#)
- OSCL_IMPORT_REF int32 [Oscl_DeleteFile \(const oscl_wchar *filename\)](#)

Friends

- class [Oscl_File](#)
- class [OsclNativeFile](#)

7.47.1 Constructor & Destructor Documentation

7.47.1.1 OSCL_IMPORT_REF Oscl_FileServer::Oscl_FileServer ()

Constructor

7.47.1.2 OSCL_IMPORT_REF Oscl_FileServer::~Oscl_FileServer ()

Destructor

7.47.2 Member Function Documentation

7.47.2.1 OSCL_IMPORT_REF int32 Oscl_FileServer::Close ()

Closes a file server.

Returns:

returns 0 on success and a non-zero value otherwise

7.47.2.2 OSCL_IMPORT_REF int32 Oscl_FileServer::Connect ()

Connects the server. This must be called before a file server can be used.

Returns:

returns 0 on success and a non-zero value otherwise

**7.47.2.3 OSCL_IMPORT_REF int32 Oscl_FileServer::Oscl_DeleteFile (const oscl_wchar *
filename)**

Deletes a file from the filesystem

Parameters:

filename name of the file to delete (Unicode)

Returns:

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

7.47.2.4 OSCL_IMPORT_REF int32 Oscl_FileServer::Oscl_DeleteFile (const char **filename*)

Deletes a file from the filesystem *

Parameters:

filename name of the file to delete (Utf8)

Returns:

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

7.47.3 Friends And Related Function Documentation**7.47.3.1 friend class Oscl_File [friend]****7.47.3.2 friend class OsclNativeFile [friend]**

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

- [oscl_file_server.h](#)

7.48 oscl_fsstat Struct Reference

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

Data Fields

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

7.48.1 Field Documentation

7.48.1.1 [uint64 oscl_fsstat::freebytes](#)

7.48.1.2 [uint64 oscl_fsstat::totalbytes](#)

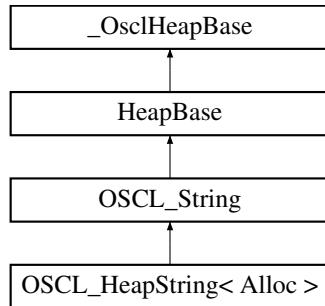
The documentation for this struct was generated from the following file:

- [oscl_file_dir_utils.h](#)

7.49 OSCL_HeapString< Alloc > Class Template Reference

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

Inheritance diagram for OSCL_HeapString< Alloc >::



Public Types

- `typedef OSCL_String::chartype chartype`
- `typedef TOSCL_StringOp 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]

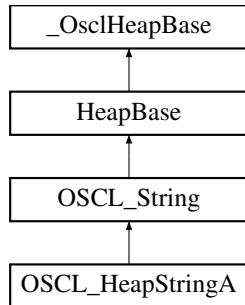
The documentation for this class was generated from the following file:

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

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

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

7.50.2 Member Typedef Documentation

7.50.2.1 `typedef OSCL_String::chartype OSCL_HeapStringA::chartype`

Reimplemented from [OSCL_String](#).

7.50.2.2 `typedef TOSCL_StringOp OSCL_HeapStringA::optype`

7.50.2.3 `typedef OSCL_wString::chartype OSCL_HeapStringA::other_chartype`

7.50.3 Constructor & Destructor Documentation

7.50.3.1 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA()`

The default constructor creates an empty string.

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

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

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

7.50.3.2 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (Oscl_DefAlloc *alloc, OsclRefCounter *ref = NULL)`

7.50.3.3 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const OSCL_HeapStringA & src)`

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

Parameters:

src: input string.

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

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

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

7.50.3.4 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const OSCL_HeapStringA & src, Oscl_DefAlloc * alloc, OsclRefCounter * ref = NULL)

7.50.3.5 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const OSCL_String & src, Oscl_DefAlloc * alloc = NULL, OsclRefCounter * ref = NULL)

7.50.3.6 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const chartype * cstr, Oscl_DefAlloc * alloc = NULL, OsclRefCounter * ref = NULL)

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

Parameters:

cp: null-terminated string.

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

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

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

7.50.3.7 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const chartype * buf, uint32 length, Oscl_DefAlloc * alloc = NULL, OsclRefCounter * ref = NULL)

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

Parameters:

src: character array, not necessarily null-terminated.

length: number of characters to copy.

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

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

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

7.50.3.8 OSCL_IMPORT_REF OSCL_HeapStringA::~OSCL_HeapStringA ()

7.50.4 Member Function Documentation

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

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

Implements [OSCL_String](#).

7.50.4.2 OSCL_IMPORT_REF uint32 OSCL_HeapStringA::get_maxsize () [virtual]

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

Implements [OSCL_String](#).

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

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

7.50.4.4 OSCL_IMPORT_REF chartype* OSCL_HeapStringA::get_str () [virtual]

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

Implements [OSCL_String](#).

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

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

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

Assignment operator

Reimplemented from [OSCL_String](#).

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

Assignment operators

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

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

Parameters:

buf: string or character array.

length: number of characters to copy.

op: conversion operation to apply

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

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

Parameters:

buf: NULL-terminated wide string.

op: conversion operation to apply

7.50.4.10 OSCL_IMPORT_REF void OSCL_HeapStringA::set (const **chartype** * *buf*, uint32 *length*)

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

Parameters:

buf: string or character array.

length: number of characters to copy.

7.50.5 Friends And Related Function Documentation

7.50.5.1 friend class OSCL_String [friend]

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

- [oscl_string_containers.h](#)

7.51 Oscl_Int64_Utils Class Reference

The Oscl_Int64_Utils class provides a wrapper for commonly used int64/uint64 operations.

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

Static Public Methods

- OSCL_IMPORT_REF void [set_int64](#) (int64 &input_value, const int32 upper, const int32 lower)
- OSCL_IMPORT_REF int32 [get_int64_upper32](#) (const int64 &input_value)
- OSCL_IMPORT_REF int32 [get_int64_lower32](#) (const int64 &input_value)
- OSCL_IMPORT_REF int32 [get_int64_middle32](#) (const int64 &input_value)
- OSCL_IMPORT_REF void [set_uint64](#) (uint64 &input_value, const uint32 upper, const uint32 lower)
- OSCL_IMPORT_REF uint32 [get_uint64_upper32](#) (const uint64 &input_value)
- OSCL_IMPORT_REF uint32 [get_uint64_lower32](#) (const uint64 &input_value)
- OSCL_IMPORT_REF uint32 [get_uint64_middle32](#) (const uint64 &input_value)

7.51.1 Detailed Description

The Oscl_Int64_Utils class provides a wrapper for commonly used int64/uint64 operations.

The Oscl_Int64_Utils class:

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

7.51.2 Member Function Documentation

- 7.51.2.1 **OSCL_IMPORT_REF int32 Oscl_Int64_Utils::get_int64_lower32 (const int64 & *input_value*) [static]**
- 7.51.2.2 **OSCL_IMPORT_REF int32 Oscl_Int64_Utils::get_int64_middle32 (const int64 & *input_value*) [static]**
- 7.51.2.3 **OSCL_IMPORT_REF int32 Oscl_Int64_Utils::get_int64_upper32 (const int64 & *input_value*) [static]**
- 7.51.2.4 **OSCL_IMPORT_REF uint32 Oscl_Int64_Utils::get_uint64_lower32 (const uint64 & *input_value*) [static]**
- 7.51.2.5 **OSCL_IMPORT_REF uint32 Oscl_Int64_Utils::get_uint64_middle32 (const uint64 & *input_value*) [static]**
- 7.51.2.6 **OSCL_IMPORT_REF uint32 Oscl_Int64_Utils::get_uint64_upper32 (const uint64 & *input_value*) [static]**
- 7.51.2.7 **OSCL_IMPORT_REF void Oscl_Int64_Utils::set_int64 (int64 & *input_value*, const int32 *upper*, const int32 *lower*) [static]**
- 7.51.2.8 **OSCL_IMPORT_REF void Oscl_Int64_Utils::set_uint64 (uint64 & *input_value*, const uint32 *upper*, const uint32 *lower*) [static]**

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

- [oscl_int64_utils.h](#)

7.52 Oscl_Less< T > Struct Template Reference

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

Public Methods

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

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

7.52.1 Member Function Documentation

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

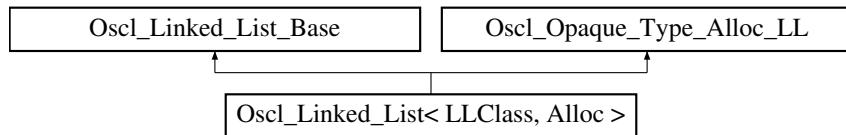
The documentation for this struct was generated from the following file:

- [oscl_map.h](#)

7.53 Oscl_Linked_List< LLClass, Alloc > Class Template Reference

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

Inheritance diagram for Oscl_Linked_List< LLClass, Alloc >::



Public Methods

- [Oscl_Linked_List \(\)](#)
- [~Oscl_Linked_List \(\)](#)
- int32 [dequeue_element \(LLClass &element\)](#)
- int32 [get_first \(LLClass &ele\)](#)
- int32 [get_next \(LLClass &ele\)](#)
- int32 [check_list \(\)](#)
- int32 [get_num_elements \(\)](#)
- int32 [add_element \(LLClass &new_element\)](#)
- int32 [add_to_front \(const LLClass &new_element\)](#)
- int32 [get_element \(int32 index, LLClass &element\)](#)
- int32 [remove_element \(const LLClass &data_to_remove\)](#)
- int32 [get_index \(const LLClass &data\)](#)
- int32 [remove_element \(const int32 index_to_remove\)](#)
- int32 [move_to_end \(const LLClass &data_to_move\)](#)
- int32 [move_to_front \(const LLClass &data_to_move\)](#)

7.53.1 Detailed Description

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

Oscl Linked List Class

7.53.2 Constructor & Destructor Documentation

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

Initialized the protected variables of list.

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

The destructor.

7.53.3 Member Function Documentation

7.53.3.1 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::add_element (LLClass & *new_element*) [inline]

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

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

7.53.3.2 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::add_to_front (const LLClass & *new_element*) [inline]

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

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

7.53.3.3 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::check_list () [inline]

Debug routine: Checks the list for elements.

Returns:

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

Reimplemented from [Oscl_Linked_List_Base](#).

7.53.3.4 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::dequeue_element (LLClass & *element*) [inline]

7.53.3.5 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::get_element (int32 *index*, LLClass & *element*) [inline]

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

Parameters:

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

Returns:

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

7.53.3.6 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::get_first (LLClass & *ele*) [inline]

Return the first element of list in passed parameter,

Parameters:

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

Returns:

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

7.53.3.7 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::get_index (const LLClass & *data*) [inline]

Returns the index for requested element.

Parameters:

data the element for which index to be return.

Returns:

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

7.53.3.8 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::get_next (LLClass & *ele*) [inline]

Return the next element of list in passed parameter,

Parameters:

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

Returns:

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

7.53.3.9 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::get_num_elements () [inline]

Get number of elements in the list.

Returns:

32-bit integer, number of elements in list.

7.53.3.10 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::move_to_end (const LLClass & *data_to_move*) [inline]

Moves the element to end of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

7.53.3.11 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::move_to_front (const LLClass & *data_to_move*) [inline]

Moves the element to front of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

7.53.3.12 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::remove_element (const int32 *index_to_remove*) [inline]

Removes the element for requested index.

Parameters:

index_to_remove

Returns:

on success return 1 otherwise return 0.

Reimplemented from [Oscl_Linked_List_Base](#).

7.53.3.13 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::remove_element (const LLClass & *data_to_remove*) [inline]

Removes the element from the list.

Parameters:

data_to_remove

Returns:

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

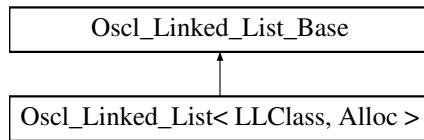
The documentation for this class was generated from the following file:

- [oscl_linked_list.h](#)

7.54 Oscl_Linked_List_Base Class Reference

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

Inheritance diagram for Oscl_Linked_List_Base::



Protected Methods

- virtual ~[Oscl_Linked_List_Base](#) ()
- OSCL_IMPORT_REF void [construct](#) ([Oscl_Opaque_Type_Alloc_LL](#) *op)
- OSCL_IMPORT_REF void [destroy](#) ()
- OSCL_IMPORT_REF int32 [get_first](#) ([OsclAny](#) *ele)
- OSCL_IMPORT_REF int32 [get_next](#) ([OsclAny](#) *ele)
- OSCL_IMPORT_REF int32 [check_list](#) ()
- OSCL_IMPORT_REF int32 [add_element](#) ([OsclAny](#) *new_element)
- OSCL_IMPORT_REF int32 [add_to_front](#) (const [OsclAny](#) *new_element)
- OSCL_IMPORT_REF int32 [get_element](#) (int32 index, [OsclAny](#) *element)
- OSCL_IMPORT_REF int32 [remove_element](#) (const [OsclAny](#) *data_to_remove)
- OSCL_IMPORT_REF int32 [get_index](#) (const [OsclAny](#) *data)
- OSCL_IMPORT_REF int32 [remove_element](#) (const int32 index_to_remove)
- OSCL_IMPORT_REF int32 [move_to_end](#) (const [OsclAny](#) *data_to_move)
- OSCL_IMPORT_REF int32 [move_to_front](#) (const [OsclAny](#) *data_to_move)

Protected Attributes

- [OsclAny](#) * head
- [OsclAny](#) * tail
- [OsclAny](#) * iterator
- int32 [num_elements](#)
- uint32 [sizeof_T](#)

7.54.1 Detailed Description

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

7.54.2 Constructor & Destructor Documentation

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

7.54.3 Member Function Documentation

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

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

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

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

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

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

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

Debug routine: Checks the list for elements.

Returns:

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

Reimplemented in [Oscl_Linked_List< LLClass, Alloc >](#).

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

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

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

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

Parameters:

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

Returns:

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

**7.54.3.7 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::get_first (OsclAny * *ele*)
[protected]**

Return the first element of list in passed parameter,

Parameters:

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

Returns:

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

**7.54.3.8 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::get_index (const OsclAny * *data*)
[protected]**

Returns the index for requested element.

Parameters:

data the element for which index to be return.

Returns:

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

**7.54.3.9 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::get_next (OsclAny * *ele*)
[protected]**

Return the next element of list in passed parameter,

Parameters:

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

Returns:

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

**7.54.3.10 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::move_to_end (const OsclAny *
data_to_move) [protected]**

Moves the element to end of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

**7.54.3.11 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::move_to_front (const OsclAny *
data_to_move) [protected]**

Moves the element to front of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

**7.54.3.12 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::remove_element (const int32
index_to_remove) [protected]**

Removes the element for requested index.

Parameters:

index_to_remove

Returns:

on success return 1 otherwise return 0.

Reimplemented in [Oscl_Linked_List< LLClass, Alloc >](#).

**7.54.3.13 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::remove_element (const OsclAny *
data_to_remove) [protected]**

Removes the element from the list.

Parameters:

data_to_remove

Returns:

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

7.54.4 Field Documentation

7.54.4.1 OsclAny* Oscl_Linked_List_Base::head [protected]**7.54.4.2 OsclAny* Oscl_Linked_List_Base::iterator [protected]****7.54.4.3 int32 Oscl_Linked_List_Base::num_elements [protected]****7.54.4.4 uint32 Oscl_Linked_List_Base::sizeof_T [protected]****7.54.4.5 OsclAny* Oscl_Linked_List_Base::tail [protected]**

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

- [oscl_linked_list.h](#)

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

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

Public Types

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

Public Methods

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

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

7.55.1 Detailed Description

template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> class Oscl_Map< Key, T, Alloc, Compare >

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

7.55.2 Member Typedef Documentation

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

7.55.3 Constructor & Destructor Documentation

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

Creates an empty map using comp as the key compare object

7.55.3.2 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Oscl_Map< Key, T, Alloc, Compare >::Oscl_Map (const **self & *x*) [inline]**

Oscl_Map copy constructor

7.55.4 Member Function Documentation

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

Returns a const iterator pointing to the beginning of the map

7.55.4.2 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator Oscl_Map< Key, T, Alloc, Compare >::begin () [inline]

Returns an iterator pointing to the beginning of the map

7.55.4.3 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> void Oscl_Map< Key, T, Alloc, Compare >::clear () [inline]

Erases all elements

7.55.4.4 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> size_type Oscl_Map< Key, T, Alloc, Compare >::count (const **key_type & *x*) const [inline]**

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

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

Returns true if map size is 0

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

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

7.55.4.7 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator Oscl_Map< Key, T, Alloc, Compare >::end () [inline]

Returns an iterator pointing to the end of the map.

7.55.4.8 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> pair_citerator_citerator Oscl_Map< Key, T, Alloc, Compare >::equal_range (const **key_type & *x*) const [inline]**

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

**7.55.4.9 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
 pair_iterator iterator Oscl_Map< Key, T, Alloc, Compare >::equal_range (const
 key_type & x) [inline]**

Finds a range containing all elements whose key is x

**7.55.4.10 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> void
 Oscl_Map< Key, T, Alloc, Compare >::erase (iterator first, iterator last) [inline]**

Erases all elements in the range [first,last)

**7.55.4.11 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> size_type
 Oscl_Map< Key, T, Alloc, Compare >::erase (const key_type & x) [inline]**

Erases the element with key x

**7.55.4.12 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> void
 Oscl_Map< Key, T, Alloc, Compare >::erase (iterator position) [inline]**

Erases the element pointed to by position

**7.55.4.13 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
 const_iterator Oscl_Map< Key, T, Alloc, Compare >::find (const key_type & x) const
 [inline]**

Finds an element whose key is x

**7.55.4.14 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator
 Oscl_Map< Key, T, Alloc, Compare >::find (const key_type & x) [inline]**

Finds an element whose key is x

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

Inserts the range [first,last) into the map

**7.55.4.16 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator
 Oscl_Map< Key, T, Alloc, Compare >::insert (iterator position, const value_type & x)
 [inline]**

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

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

Inserts x into the map

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

Returns the key compare object used by the map

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

Finds the first element whose key is not less than x

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

Finds the first element whose key is not less than x

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

Returns the maximum possible size of the map

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

Oscl_Map assignment operator

7.55.4.23]

template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> T& Oscl_Map< Key, T, Alloc, Compare >::operator[] (const `key_type` & k) [inline]

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

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

Returns the size of the map

**7.55.4.25 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
const_iterator Oscl_Map< Key, T, Alloc, Compare >::upper_bound (const key_type &
x) const [inline]**

Finds the first element whose key is not greater than x

**7.55.4.26 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator
Oscl_Map< Key, T, Alloc, Compare >::upper_bound (const key_type & x) [inline]**

Finds the first element whose key is not greater than x

**7.55.4.27 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
value_compare Oscl_Map< Key, T, Alloc, Compare >::value_comp () const
[inline]**

Returns the value compare object used by the map

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

- [oscl_map.h](#)

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

```
#include <oscl_map.h>
```

Public Methods

- bool [operator\(\)](#) (const [value_type](#) &x, const [value_type](#) &y) const

Protected Methods

- [value_compare](#) (Compare c)

Protected Attributes

- Compare [comp](#)

Friends

- class [Oscl_Map< Key, T, Alloc, Compare >](#)

```
template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> class Oscl_Map< Key,  
T, Alloc, Compare >::value_compare
```

7.56.1 Constructor & Destructor Documentation

```
7.56.1.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::value\_compare (Compare c) [inline, protected]
```

7.56.2 Member Function Documentation

```
7.56.2.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> bool  
Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::operator\(\) (const value\_type &  
x, const value\_type &y) const [inline]
```

7.56.3 Friends And Related Function Documentation

```
7.56.3.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> friend class  
Oscl\_Map< Key, T, Alloc, Compare > [friend]
```

7.56.4 Field Documentation

```
7.56.4.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Compare  
Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::comp [protected]
```

The documentation for this class was generated from the following file:

- [oscl_map.h](#)

7.57 Oscl_MTLinked_List< LLClass, Alloc, TheLock > Class Template Reference

```
#include <oscl_linked_list.h>
```

Public Methods

- [Oscl_MTLinked_List \(\)](#)
- [~Oscl_MTLinked_List \(\)](#)
- int32 [dequeue_element \(LLClass &element\)](#)
- int32 [add_element \(LLClass &new_element\)](#)
- int32 [add_to_front \(LLClass &new_element\)](#)
- uint32 [get_element \(int32 index, LLClass &element\)](#)
- int32 [remove_element \(const LLClass &data_to_remove\)](#)
- int32 [get_index \(const LLClass &data\)](#)
- int32 [remove_element \(const int32 index_to_remove\)](#)
- int32 [move_to_end \(const LLClass &data_to_move\)](#)
- int32 [move_to_front \(const LLClass &data_to_move\)](#)

Protected Attributes

- [Oscl_Linked_List< LLClass, Alloc > the_list](#)

7.57.1 Detailed Description

template<class LLClass, class Alloc, class TheLock> class Oscl_MTLinked_List< LLClass, Alloc, TheLock >

Oscl_MTLinked_List is a multi-threaded version of the LinkedList. It has mutex protection to allow access by different threads.

7.57.2 Constructor & Destructor Documentation

7.57.2.1 template<class LLClass, class Alloc, class TheLock> Oscl_MTLinked_List< LLClass, Alloc, TheLock >::Oscl_MTLinked_List () [inline]

Constructor for Oscl_MTLinked_List

7.57.2.2 template<class LLClass, class Alloc, class TheLock> Oscl_MTLinked_List< LLClass, Alloc, TheLock >::~Oscl_MTLinked_List () [inline]

Destructor for Oscl_MTLinked_List

7.57.3 Member Function Documentation

7.57.3.1 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::add_element (LLClass & *new_element*) [inline]

Adds new element to the Multi Threaded Linked list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

7.57.3.2 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::add_to_front (LLClass & *new_element*) [inline]

Adds new element at the start of the Multi Threaded Linked list. if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

7.57.3.3 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::dequeue_element (LLClass & *element*) [inline]

7.57.3.4 template<class LLClass, class Alloc, class TheLock> uint32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::get_element (int32 *index*, LLClass & *element*) [inline]

Search and returns the element in the Multi Threaded Linked List for passed index.

Parameters:

index, element The index is the count for the node.

Returns:

32-bit integer on success returns 1 otherwise returns 0.

7.57.3.5 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::get_index (const LLClass & *data*) [inline]

Returns the index for requested element.

Parameters:

data the element for which index to be return.

Returns:

32-bit integer if data is found in the list it returns index otherwise it returns -1.

7.57.3.6 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::move_to_end (const LLClass & *data_to_move*) [inline]

Moves the element to end of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

7.57.3.7 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::move_to_front (const LLClass & *data_to_move*) [inline]

Moves the element to front of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

7.57.3.8 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::remove_element (const int32 *index_to_remove*) [inline]

Removes the element for requested index.

Parameters:

index_to_remove

Returns:

on success return 1 otherwise return 0.

7.57.3.9 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::remove_element (const LLClass & *data_to_remove*) [inline]

Removes the element from the list.

Parameters:

data_to_remove

Returns:

32-bit integer on if element fount in the list returns 1 otherwise returns 0.

7.57.4 Field Documentation

7.57.4.1 template<class LLClass, class Alloc, class TheLock> Oscl_Linked_List<LLClass, Alloc> Oscl_MTLinked_List< LLClass, Alloc, TheLock >::the_list [protected]

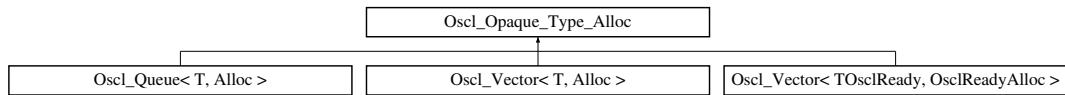
The documentation for this class was generated from the following file:

-
- [oscl_linked_list.h](#)

7.58 Oscl_Opaque_Type_Alloc Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl_Opaque_Type_Alloc::



Public Methods

- virtual void **construct** (**OsclAny** *p, const **OsclAny** *init_val)=0
- virtual void **destroy** (**OsclAny** *p)=0
- virtual **OsclAny** * **allocate** (const uint32 size)=0
- virtual void **deallocate** (**OsclAny** *p)=0

7.58.1 Detailed Description

This class combines opaque type operations with memory allocation operations.

7.58.2 Member Function Documentation

7.58.2.1 virtual **OsclAny* Oscl_Opaque_Type_Alloc::allocate (const uint32 size) [pure virtual]**

Allocate "size" bytes

7.58.2.2 virtual void Oscl_Opaque_Type_Alloc::construct (OsclAny** * p, const **OsclAny** * init_val) [pure virtual]**

Construct element at p using element at init_val as the initial value. Both pointers must be non-NULL.

7.58.2.3 virtual void Oscl_Opaque_Type_Alloc::deallocate (OsclAny** * p) [pure virtual]**

Deallocate memory previously allocated with "allocate"

7.58.2.4 virtual void Oscl_Opaque_Type_Alloc::destroy (OsclAny** * p) [pure virtual]**

Destroy element at p.

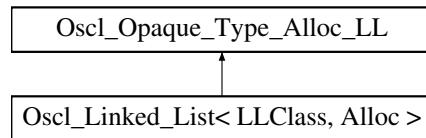
The documentation for this class was generated from the following file:

- [oscl_opaque_type.h](#)

7.59 Oscl_Opaque_Type_Alloc_LL Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl_Opaque_Type_Alloc_LL::



Public Methods

- virtual void `construct (OsclAny *p, const OsclAny *init_val)=0`
- virtual void `destroy (OsclAny *p)=0`
- virtual `OsclAny * allocate (const uint32 size)=0`
- virtual void `deallocate (OsclAny *p)=0`
- virtual `OsclAny * get_next (const OsclAny *elem) const=0`
- virtual void `set_next (OsclAny *elem, const OsclAny *nextelem)=0`
- virtual void `get_data (OsclAny *elem, OsclAny *data_val)=0`
- virtual bool `compare_data (const OsclAny *elem, const OsclAny *data_val) const=0`

7.59.1 Detailed Description

This class combines opaque type operations with memory allocation operations and linked list support

7.59.2 Member Function Documentation

7.59.2.1 virtual `OsclAny* Oscl_Opaque_Type_Alloc_LL::allocate (const uint32 size)` [pure virtual]

Allocate "size" bytes

7.59.2.2 virtual `bool Oscl_Opaque_Type_Alloc_LL::compare_data (const OsclAny * elem, const OsclAny * data_val) const` [pure virtual]

Compare data.

7.59.2.3 virtual `void Oscl_Opaque_Type_Alloc_LL::construct (OsclAny * p, const OsclAny * init_val)` [pure virtual]

Construct element at p using element at init_val as the initial value. Both pointers must be non-NULL.

7.59.2.4 virtual `void Oscl_Opaque_Type_Alloc_LL::deallocate (OsclAny * p)` [pure virtual]

Deallocate memory previously allocated with "allocate"

7.59.2.5 virtual void Oscl_Opaque_Type_Alloc_LL::destroy (OsclAny **p*) [pure virtual]

Destroy element at p.

7.59.2.6 virtual void Oscl_Opaque_Type_Alloc_LL::get_data (OsclAny **elem*, OsclAny **data_val*) [pure virtual]

Get data

7.59.2.7 virtual OsclAny* Oscl_Opaque_Type_Alloc_LL::get_next (const OsclAny **elem*) const [pure virtual]

Get next element in linked list.

7.59.2.8 virtual void Oscl_Opaque_Type_Alloc_LL::set_next (OsclAny **elem*, const OsclAny **nextelem*) [pure virtual]

Set next element in linked list.

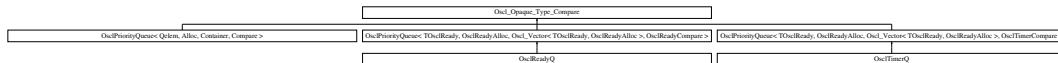
The documentation for this class was generated from the following file:

- [oscl_opaque_type.h](#)

7.60 Oscl_Opaque_Type_Compare Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl_Opaque_Type_Compare::



Public Methods

- virtual void `swap (OsclAny *a, const OsclAny *b)=0`
- virtual int `compare_LT (OsclAny *a, OsclAny *b) const=0`
- virtual int `compare_EQ (const OsclAny *a, const OsclAny *b) const=0`

7.60.1 Detailed Description

Opaque type operations with swap & comparisons.

7.60.2 Member Function Documentation

7.60.2.1 virtual int Oscl_Opaque_Type_Compare::compare_EQ (const OsclAny * a, const OsclAny * b) const [pure virtual]

Return a==b.

Implemented in `OsclPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >`.

7.60.2.2 virtual int Oscl_Opaque_Type_Compare::compare_LT (OsclAny * a, OsclAny * b) const [pure virtual]

Return a<b.

Implemented in `OsclPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >`.

7.60.2.3 virtual void Oscl_Opaque_Type_Compare::swap (OsclAny * a, const OsclAny * b) [pure virtual]

Swap element at "a" with element at "b". Both pointers must be non-NULL.

Implemented in `OsclPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >`.

The documentation for this class was generated from the following file:

- [oscl_opaque_type.h](#)

7.61 Oscl_Pair< T1, T2 > Struct Template Reference

```
#include <oscl_tree.h>
```

Public Methods

- [Oscl_Pair \(\)](#)
- [Oscl_Pair \(const T1 &a, const T2 &b\)](#)

Data Fields

- [T1 first](#)
- [T2 second](#)

```
template<class T1, class T2> struct Oscl_Pair< T1, T2 >
```

7.61.1 Constructor & Destructor Documentation

7.61.1.1 template<class T1, class T2> Oscl_Pair< T1, T2 >::Oscl_Pair () [inline]

7.61.1.2 template<class T1, class T2> Oscl_Pair< T1, T2 >::Oscl_Pair (const T1 &a, const T2 &b) [inline]

7.61.2 Field Documentation

7.61.2.1 template<class T1, class T2> T1 Oscl_Pair< T1, T2 >::first

7.61.2.2 template<class T1, class T2> T2 Oscl_Pair< T1, T2 >::second

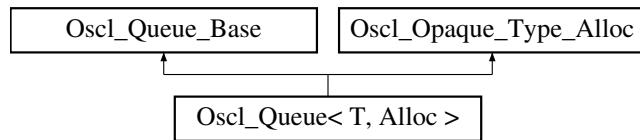
The documentation for this struct was generated from the following file:

- [oscl_tree.h](#)

7.62 Oscl_Queue< T, Alloc > Class Template Reference

```
#include <oscl_queue.h>
```

Inheritance diagram for Oscl_Queue< T, Alloc >::



Public Types

- typedef T [value_type](#)
- typedef T * [pointer](#)
- typedef T & [reference](#)
- typedef const T & [const_reference](#)
- typedef uint32 [size_type](#)

Public Methods

- [Oscl_Queue \(\)](#)
- [Oscl_Queue \(uint32 n\)](#)
- virtual [~Oscl_Queue \(\)](#)
- void [push \(const T &x\)](#)
- [reference front \(\)](#)
- [const_reference front \(\) const](#)
- void [pop \(\)](#)
- [reference back \(\)](#)
- [const_reference back \(\) const](#)
- void [clear \(\)](#)

7.62.1 Detailed Description

template<class T, class Alloc> class Oscl_Queue< T, Alloc >

Oscl_Queue Class. A subset of STL::Queue methods. Oscl_Queue supports constant time insertion (at the end) and removal of elements at the front of the queue. It does not support insertion or removal of elements at the other ends or middle of the queue, nor random access to elements. * No iteration capability is [currently] supplied. * No assignment or copy capability is [currently] supplied. The number of elements in a queue can vary dynamically, and memory management is performed automatically.

7.62.2 Member Typedef Documentation

- 7.62.2.1 **template<class T, class Alloc> typedef const T& Oscl_Queue< T, Alloc >::const_reference**
- 7.62.2.2 **template<class T, class Alloc> typedef T* Oscl_Queue< T, Alloc >::pointer**
- 7.62.2.3 **template<class T, class Alloc> typedef T& Oscl_Queue< T, Alloc >::reference**
- 7.62.2.4 **template<class T, class Alloc> typedef uint32 Oscl_Queue< T, Alloc >::size_type**
- 7.62.2.5 **template<class T, class Alloc> typedef T Oscl_Queue< T, Alloc >::value_type**

7.62.3 Constructor & Destructor Documentation

- 7.62.3.1 **template<class T, class Alloc> Oscl_Queue< T, Alloc >::Oscl_Queue () [inline]**

Creates an empty queue.

- 7.62.3.2 **template<class T, class Alloc> Oscl_Queue< T, Alloc >::Oscl_Queue (uint32 n) [inline]**

Creates an empty queue with capacity n.

Parameters:

n creates a queue with n elements. The main reason for specifying n is efficiency. If you know the capacity to which your queue must grow, then it is more efficient to allocate the queue all at once rather than rely on the automatic reallocation scheme.

- 7.62.3.3 **template<class T, class Alloc> virtual Oscl_Queue< T, Alloc >::~Oscl_Queue () [inline, virtual]**

The destructor.

7.62.4 Member Function Documentation

- 7.62.4.1 **template<class T, class Alloc> const_reference Oscl_Queue< T, Alloc >::back () const [inline]**

Returns the last element (const)

- 7.62.4.2 **template<class T, class Alloc> reference Oscl_Queue< T, Alloc >::back () [inline]**

Returns the last element: "back" (generally not too useful, but some debugging aids might want to find out what was just added)

- 7.62.4.3 **template<class T, class Alloc> void Oscl_Queue< T, Alloc >::clear () [inline]**

Removes all elements.

Reimplemented from [Oscl_Queue_Base](#).

7.62.4.4 template<class T, class Alloc> const_reference Oscl_Queue< T, Alloc >::front () const [inline]

Returns the first element (const)

7.62.4.5 template<class T, class Alloc> reference Oscl_Queue< T, Alloc >::front () [inline]

Returns the first element.

Reimplemented from [Oscl_Queue_Base](#).

7.62.4.6 template<class T, class Alloc> void Oscl_Queue< T, Alloc >::pop () [inline]

Removes the first element

Reimplemented from [Oscl_Queue_Base](#).

7.62.4.7 template<class T, class Alloc> void Oscl_Queue< T, Alloc >::push (const T & x) [inline]

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL_LEAVE will occur

Parameters:

x new element

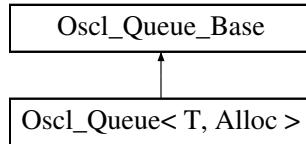
The documentation for this class was generated from the following file:

- [oscl_queue.h](#)

7.63 Oscl_Queue_Base Class Reference

```
#include <oscl_queue.h>
```

Inheritance diagram for Oscl_Queue_Base::



Public Methods

- uint32 `size () const`
- uint32 `capacity () const`
- bool `empty () const`
- OSCL_IMPORT_REF void `reserve (uint32 n)`

Protected Methods

- OSCL_IMPORT_REF void `construct (Oscl_Opaque_Type_Alloc *aType)`
- OSCL_IMPORT_REF void `construct (Oscl_Opaque_Type_Alloc *aType, uint32 n)`
- virtual `~Oscl_Queue_Base ()`
- OSCL_IMPORT_REF void `destroy ()`
- OSCL_IMPORT_REF void `push (const OsclAny *x)`
- OSCL_IMPORT_REF void `pop ()`
- OSCL_IMPORT_REF void `clear ()`

Protected Attributes

- uint32 `numelems`
- uint32 `bufsize`
- `OsclAny * elems`
- uint32 `sizeof_T`
- uint32 `ifront`
- uint32 `irear`

7.63.1 Detailed Description

`Oscl_Queue_Base` is a non-templatized base class for [Oscl_Queue](#). The purpose of this base class is to avoid large inline routines in the [Oscl_Queue](#) implementation. This class is not intended for direct instantiation except by [Oscl_Queue](#).

7.63.2 Constructor & Destructor Documentation

7.63.2.1 virtual Oscl_Queue_Base::~Oscl_Queue_Base () [inline, protected, virtual]

The destructor.

7.63.3 Member Function Documentation

7.63.3.1 **uint32 Oscl_Queue_Base::capacity () const [inline]**

Returns the allocated memory of the queue.

7.63.3.2 **OSCL_IMPORT_REF void Oscl_Queue_Base::clear () [protected]**

Removes all elements.

Reimplemented in [Oscl_Queue< T, Alloc >](#).

7.63.3.3 **OSCL_IMPORT_REF void Oscl_Queue_Base::construct (Oscl_Opaque_Type_Alloc * aType, uint32 n) [protected]**

7.63.3.4 **OSCL_IMPORT_REF void Oscl_Queue_Base::construct (Oscl_Opaque_Type_Alloc * aType) [protected]**

7.63.3.5 **OSCL_IMPORT_REF void Oscl_Queue_Base::destroy () [protected]**

Like an explicit destructor call.

7.63.3.6 **bool Oscl_Queue_Base::empty () const [inline]**

True if there are no elements in the queue

7.63.3.7 **OSCL_IMPORT_REF void Oscl_Queue_Base::pop () [protected]**

Removes the first element

Reimplemented in [Oscl_Queue< T, Alloc >](#).

7.63.3.8 **OSCL_IMPORT_REF void Oscl_Queue_Base::push (const OsclAny * x) [protected]**

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL_LEAVE will occur

Parameters:

x new element

7.63.3.9 **OSCL_IMPORT_REF void Oscl_Queue_Base::reserve (uint32 n)**

Reallocates memory if necessary to a capacity of *n* elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

Parameters:

n size of vector

7.63.3.10 uint32 Oscl_Queue_Base::size () const [inline]

Returns the size of the queue.

7.63.4 Field Documentation

7.63.4.1 uint32 Oscl_Queue_Base::bufsize [protected]**7.63.4.2 OsclAny* Oscl_Queue_Base::elems [protected]****7.63.4.3 uint32 Oscl_Queue_Base::ifront [protected]****7.63.4.4 uint32 Oscl_Queue_Base::irear [protected]****7.63.4.5 uint32 Oscl_Queue_Base::numelems [protected]****7.63.4.6 uint32 Oscl_Queue_Base::sizeof_T [protected]**

The documentation for this class was generated from the following file:

- [oscl_queue.h](#)

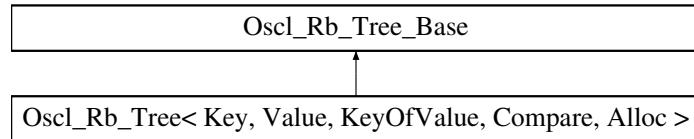


7.64 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

7.64 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::



Public Types

- typedef Key `key_type`
- typedef Value `value_type`
- typedef `value_type * pointer`
- typedef `const value_type * const_pointer`
- typedef `value_type & reference`
- typedef `const value_type & const_reference`
- typedef `Oscl_Rb_Tree_Node< Value >::link_type link_type`
- typedef `Oscl_Rb_Tree_Iterator< value_type > iterator`
- typedef `Oscl_Rb_Tree_Const_Iterator< value_type > const_iterator`
- typedef uint32 `size_type`
- typedef int32 `difference_type`

Public Methods

- `Oscl_Rb_Tree (const Compare &comp=Compare())`
- `Oscl_Rb_Tree (const Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > &x)`
- `~Oscl_Rb_Tree ()`
- `Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > & operator= (const Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > &x)`
- `iterator begin ()`
- `const_iterator begin () const`
- `iterator end ()`
- `const_iterator end () const`
- `bool empty () const`
- `size_type size () const`
- `size_type max_size () const`
- `Oscl_Pair< iterator, bool > insert_unique (const value_type &v)`
- `iterator insert_unique (iterator position, const value_type &v)`
- `void insert_unique (const iterator first, const iterator last)`
- `void insert_unique (const value_type *first, const value_type *last)`
- `void erase (iterator position)`
- `size_type erase (const key_type &x)`
- `void erase (iterator first, iterator last)`
- `void erase (const key_type *first, const key_type *last)`



7.64 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

- void [clear \(\)](#)
- [iterator find \(const Key &k\)](#)
- [const_iterator find \(const Key &k\) const](#)
- [size_type count \(const Key &k\) const](#)
- [iterator lower_bound \(const Key &k\)](#)
- [const_iterator lower_bound \(const Key &k\) const](#)
- [iterator upper_bound \(const Key &k\)](#)
- [const_iterator upper_bound \(const Key &k\) const](#)
- [Oscl_Pair< iterator, iterator > equal_range \(const Key &k\)](#)
- [Oscl_Pair< const_iterator, const_iterator > equal_range \(const Key &k\) const](#)



7.64 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> class Oscl_Rb_-Tree< Key, Value, KeyOfValue, Compare, Alloc >

7.64.1 Member Typedef Documentation

7.64.1.1 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [Oscl_Rb_Tree_Const_Iterator<value_type>](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const_iterator

7.64.1.2 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef const [value_type*](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const_pointer

7.64.1.3 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef const [value_type&](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const_reference

7.64.1.4 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef int32 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::difference_type

7.64.1.5 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [Oscl_Rb_Tree_Iterator<value_type>](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::iterator

7.64.1.6 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Key Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::key_type

7.64.1.7 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [Oscl_Rb_Tree_Node<Value>::link_type](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::link_type

7.64.1.8 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [value_type*](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::pointer

7.64.1.9 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [value_type&](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::reference

7.64.1.10 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef uint32 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::size_type

7.64.1.11 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Value Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::value_type

7.64.2 Constructor & Destructor Documentation

7.64.2.1 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::Oscl_Rb_Tree (const Compare & *comp* = Compare() [inline])

7.64.2.2 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::Oscl_Rb_Tree (const Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > & *x*) [inline]

7.64.2.3 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::~Oscl_Rb_Tree ()



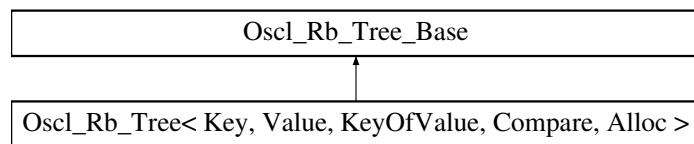
7.64 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

- [oscl_tree.h](#)

7.65 Oscl_Rb_Tree_Base Class Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl_Rb_Tree_Base::



Public Types

- `typedef Oscl_Rb_Tree_Node_Base::base_link_type base_link_type`

Public Methods

- `OSCL_IMPORT_REF void rotate_left (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF void rotate_right (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF void rebalance (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF base_link_type rebalance_for_erase (base_link_type z, base_link_type &root, base_link_type &leftmost, base_link_type &rightmost)`

7.65.1 Member Typedef Documentation

7.65.1.1 `typedef Oscl_Rb_Tree_Node_Base::base_link_type Oscl_Rb_Tree_Base::base_link_type`

7.65.2 Member Function Documentation

7.65.2.1 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rebalance (base_link_type x, base_link_type &root)`

7.65.2.2 `OSCL_IMPORT_REF base_link_type Oscl_Rb_Tree_Base::rebalance_for_erase (base_link_type z, base_link_type &root, base_link_type &leftmost, base_link_type &rightmost)`

7.65.2.3 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rotate_left (base_link_type x, base_link_type &root)`

7.65.2.4 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rotate_right (base_link_type x, base_link_type &root)`

The documentation for this class was generated from the following file:

- `oscl_tree.h`

7.66 Oscl_Rb_Tree_Const_Iterator< Value > Struct Template Reference

```
#include <oscl_tree.h>
```

Public Types

- `typedef Value value_type`
- `typedef const value_type & reference`
- `typedef const value_type * pointer`
- `typedef Oscl_Rb_Tree_Const_Iterator< Value > const_iterator`
- `typedef Oscl_Rb_Tree_Const_Iterator< Value > self`
- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef Oscl_Rb_Tree_Node< Value > * link_type`

Public Methods

- `Oscl_Rb_Tree_Const_Iterator()`
- `Oscl_Rb_Tree_Const_Iterator(link_type x)`
- `Oscl_Rb_Tree_Const_Iterator(const const_iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator==(const self &x)`
- `bool operator!=(const self &x)`
- `self & operator++()`
- `self operator++(int)`
- `self & operator--()`
- `self operator--(int)`

Data Fields

- `base_link_type node`

template<class Value> struct Oscl_Rb_Tree_Const_Iterator< Value >

7.66.1 Member Typedef Documentation

- 7.66.1.1 template<class Value> typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Const_Iterator< Value >::base_link_type
- 7.66.1.2 template<class Value> typedef Oscl_Rb_Tree_Const_Iterator<Value> Oscl_Rb_Tree_Const_Iterator< Value >::const_iterator
- 7.66.1.3 template<class Value> typedef Oscl_Rb_Tree_Node<Value>* Oscl_Rb_Tree_Const_Iterator< Value >::link_type
- 7.66.1.4 template<class Value> typedef const value_type* Oscl_Rb_Tree_Const_Iterator< Value >::pointer
- 7.66.1.5 template<class Value> typedef const value_type& Oscl_Rb_Tree_Const_Iterator< Value >::reference
- 7.66.1.6 template<class Value> typedef Oscl_Rb_Tree_Const_Iterator<Value> Oscl_Rb_Tree_Const_Iterator< Value >::self
- 7.66.1.7 template<class Value> typedef Value Oscl_Rb_Tree_Const_Iterator< Value >::value_type

7.66.2 Constructor & Destructor Documentation

- 7.66.2.1 template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator () [inline]
- 7.66.2.2 template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator ([link_type](#) x) [inline]
- 7.66.2.3 template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator (const [const_iterator](#) & it) [inline]

7.66.3 Member Function Documentation

- 7.66.3.1 template<class Value> [reference](#) Oscl_Rb_Tree_Const_Iterator< Value >::operator * () const [inline]
- 7.66.3.2 template<class Value> bool Oscl_Rb_Tree_Const_Iterator< Value >::operator!= (const [self](#) & x) [inline]
- 7.66.3.3 template<class Value> [self](#) Oscl_Rb_Tree_Const_Iterator< Value >::operator++ (int) [inline]
- 7.66.3.4 template<class Value> [self&](#) Oscl_Rb_Tree_Const_Iterator< Value >::operator++ () [inline]
- 7.66.3.5 template<class Value> [self](#) Oscl_Rb_Tree_Const_Iterator< Value >::operator- (int) [inline]
- 7.66.3.6 template<class Value> [self&](#) Oscl_Rb_Tree_Const_Iterator< Value >::operator- () [inline]

-
- [oscl_tree.h](#)

7.67 Oscl_Rb_Tree_Iterator< Value > Struct Template Reference

```
#include <oscl_tree.h>
```

Public Types

- `typedef Value value_type`
- `typedef value_type & reference`
- `typedef value_type * pointer`
- `typedef Oscl_Rb_Tree_Iterator< Value > iterator`
- `typedef Oscl_Rb_Tree_Iterator< Value > self`
- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef Oscl_Rb_Tree_Node< Value > * link_type`

Public Methods

- `Oscl_Rb_Tree_Iterator ()`
- `Oscl_Rb_Tree_Iterator (link_type x)`
- `Oscl_Rb_Tree_Iterator (const iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator== (const self &x)`
- `bool operator!= (const self &x)`
- `self & operator++ ()`
- `self operator++ (int)`
- `self & operator– ()`
- `self operator– (int)`

Data Fields

- `base_link_type node`

```
template<class Value> struct Oscl_Rb_Tree_Iterator< Value >
```

7.67.1 Member Typedef Documentation

- 7.67.1.1 template<class Value> typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Iterator< Value >::base_link_type
- 7.67.1.2 template<class Value> typedef Oscl_Rb_Tree_Iterator<Value> Oscl_Rb_Tree_Iterator< Value >::iterator
- 7.67.1.3 template<class Value> typedef Oscl_Rb_Tree_Node<Value>* Oscl_Rb_Tree_Iterator< Value >::link_type
- 7.67.1.4 template<class Value> typedef value_type* Oscl_Rb_Tree_Iterator< Value >::pointer
- 7.67.1.5 template<class Value> typedef value_type& Oscl_Rb_Tree_Iterator< Value >::reference
- 7.67.1.6 template<class Value> typedef Oscl_Rb_Tree_Iterator<Value> Oscl_Rb_Tree_Iterator< Value >::self
- 7.67.1.7 template<class Value> typedef Value Oscl_Rb_Tree_Iterator< Value >::value_type

7.67.2 Constructor & Destructor Documentation

- 7.67.2.1 template<class Value> Oscl_Rb_Tree_Iterator< Value >::Oscl_Rb_Tree_Iterator () [inline]
- 7.67.2.2 template<class Value> Oscl_Rb_Tree_Iterator< Value >::Oscl_Rb_Tree_Iterator (link_type x) [inline]
- 7.67.2.3 template<class Value> Oscl_Rb_Tree_Iterator< Value >::Oscl_Rb_Tree_Iterator (const iterator & it) [inline]

7.67.3 Member Function Documentation

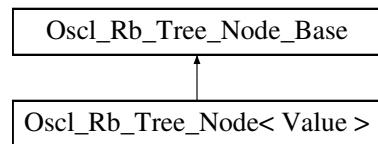
- 7.67.3.1 template<class Value> reference Oscl_Rb_Tree_Iterator< Value >::operator * () const [inline]
- 7.67.3.2 template<class Value> bool Oscl_Rb_Tree_Iterator< Value >::operator!= (const self & x) [inline]
- 7.67.3.3 template<class Value> self Oscl_Rb_Tree_Iterator< Value >::operator++ (int) [inline]
- 7.67.3.4 template<class Value> self& Oscl_Rb_Tree_Iterator< Value >::operator++ () [inline]
- 7.67.3.5 template<class Value> self Oscl_Rb_Tree_Iterator< Value >::operator- (int) [inline]
- 7.67.3.6 template<class Value> self& Oscl_Rb_Tree_Iterator< Value >::operator- () [inline]
- 7.67.3.7 template<class Value> pointer Oscl_Rb_Tree_Iterator< Value >::operator -> () const [inline]

- [oscl_tree.h](#)

7.68 Oscl_Rb_Tree_Node< Value > Struct Template Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl_Rb_Tree_Node< Value >::



Public Types

- [typedef Value value_type](#)
- [typedef Oscl_Rb_Tree_Node< Value > * link_type](#)

Data Fields

- [value_type value](#)

```
template<class Value> struct Oscl_Rb_Tree_Node< Value >
```

7.68.1 Member Typedef Documentation

7.68.1.1 template<class Value> typedef Oscl_Rb_Tree_Node<Value>* Oscl_Rb_Tree_Node< Value >::link_type

7.68.1.2 template<class Value> typedef Value Oscl_Rb_Tree_Node< Value >::value_type

7.68.2 Field Documentation

7.68.2.1 template<class Value> [value_type](#) Oscl_Rb_Tree_Node< Value >::value

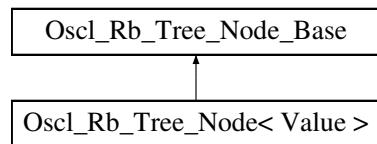
The documentation for this struct was generated from the following file:

- [oscl_tree.h](#)

7.69 Oscl_Rb_Tree_Node_Base Struct Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl_Rb_Tree_Node_Base::



Public Types

- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef enum Oscl_Rb_Tree_Node_Base::RedBl color_type`
- `enum RedBl { red, black }`

Static Public Methods

- `base_link_type minimum (base_link_type x)`
- `base_link_type maximum (base_link_type x)`

Data Fields

- `color_type color`
- `base_link_type parent`
- `base_link_type left`
- `base_link_type right`

7.69.1 Member Typedef Documentation

7.69.1.1 `typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Node_Base::base_link_type`

7.69.1.2 `typedef enum Oscl_Rb_Tree_Node_Base::RedBl Oscl_Rb_Tree_Node_Base::color_type`

7.69.2 Member Enumeration Documentation

7.69.2.1 `enum Oscl_Rb_Tree_Node_Base::RedBl`

Enumeration values:

`red`

`black`

7.69.3 Member Function Documentation

7.69.3.1 **base_link_type** Oscl_Rb_Tree_Node_Base::maximum (**base_link_type** *x*) [inline, static]

7.69.3.2 **base_link_type** Oscl_Rb_Tree_Node_Base::minimum (**base_link_type** *x*) [inline, static]

7.69.4 Field Documentation

7.69.4.1 **color_type** Oscl_Rb_Tree_Node_Base::color

7.69.4.2 **base_link_type** Oscl_Rb_Tree_Node_Base::left

7.69.4.3 **base_link_type** Oscl_Rb_Tree_Node_Base::parent

7.69.4.4 **base_link_type** Oscl_Rb_Tree_Node_Base::right

The documentation for this struct was generated from the following file:

- [oscl_tree.h](#)

7.70 Oscl_Select1st< V, U > Struct Template Reference

```
#include <oscl_map.h>
```

Public Methods

- const U & [operator\(\)](#) (const V &x) const

```
template<class V, class U> struct Oscl_Select1st< V, U >
```

7.70.1 Member Function Documentation

**7.70.1.1 template<class V, class U> const U& Oscl_Select1st< V, U >::operator() (const V & x)
const [inline]**

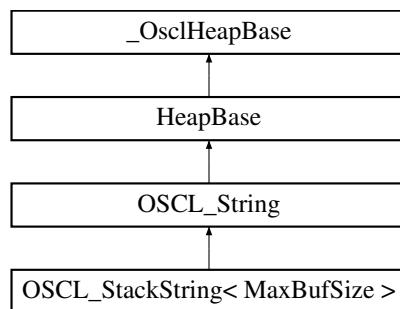
The documentation for this struct was generated from the following file:

- [oscl_map.h](#)

7.71 OSCL_StackString< MaxBufSize > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_StackString< MaxBufSize >::



Public Types

- `typedef OSCL_String::chartype chartype`
- `typedef TOSCL_StringOp optype`
- `typedef OSCL_wString::chartype other_chartype`

Public Methods

- `OSCL_StackString()`
- `OSCL_StackString(const OSCL_StackString &src)`
- `OSCL_StackString(const OSCL_String &src)`
- `OSCL_StackString(const chartype *cstr)`
- `OSCL_StackString(const chartype *buf, uint32 length)`
- `~OSCL_StackString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype * get_cstr() const`
- `chartype * get_str() const`
- `OSCL_StackString & operator=(const OSCL_StackString &src)`
- `OSCL_StackString & operator=(const OSCL_String &src)`
- `OSCL_StackString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`
- `void set(const other_chartype *buf, optype op)`
- `void set(const other_chartype *buf, uint32 length, optype op)`

Friends

- `class OSCL_String`

7.71.1 Detailed Description

template<uint32 MaxBufSize> class OSCL_StackString< MaxBufSize >

OSCL_StackString is a simple string class, compatible with regular character array strings.

The string array is fixed length, is allocated from the stack, and is modifiable. Operations that update the string will automatically truncate it to fit the fixed size storage. This is recommended for use for short strings (<255). Use [OSCL_HeapString](#) for very large strings to avoid stack overflow.

Parameters:

C: type of character.

MaxBufSize: maximum string length not including null terminator.

7.71.2 Member Typedef Documentation

7.71.2.1 template<uint32 MaxBufSize> typedef OSCL_String::chartype OSCL_StackString< MaxBufSize >::chartype

Reimplemented from [OSCL_String](#).

7.71.2.2 template<uint32 MaxBufSize> typedef TOSCL_StringOp OSCL_StackString< MaxBufSize >::optype

7.71.2.3 template<uint32 MaxBufSize> typedef OSCL_wString::chartype OSCL_StackString< MaxBufSize >::other_chartype

7.71.3 Friends And Related Function Documentation

7.71.3.1 template<uint32 MaxBufSize> friend class OSCL_String [friend]

The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.72 oscl_stat_buf Struct Reference

```
#include <oscl_file_dir_utils.h>
```

Data Fields

- uint32 [mode](#)
- uint32 [perms](#)

7.72.1 Field Documentation

7.72.1.1 uint32 oscl_stat_buf::mode

7.72.1.2 uint32 oscl_stat_buf::perms

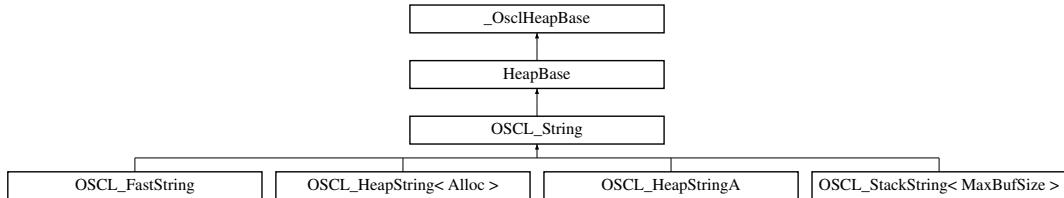
The documentation for this struct was generated from the following file:

- [oscl_file_dir_utils.h](#)

7.73 OSCL_String Class Reference

```
#include <oscl_string.h>
```

Inheritance diagram for OSCL_String::



Public Types

- `typedef char chartype`

Public Methods

- `virtual uint32 get_size () const=0`
- `virtual uint32 get_maxsize () const=0`
- `virtual const chartype * get_cstr () const=0`
- `virtual OSCL_IMPORT_REF bool is_writable () const`
- `virtual chartype * get_str () const=0`
- `OSCL_IMPORT_REF OSCL_String & operator= (const OSCL_String &src)`
- `OSCL_IMPORT_REF OSCL_String & operator= (const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_String & operator+= (const OSCL_String &src)`
- `OSCL_IMPORT_REF OSCL_String & operator+= (const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_String & operator+= (const chartype c)`
- `OSCL_IMPORT_REF bool operator== (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator!= (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator< (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator<= (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator> (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator>= (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator== (const chartype *cstr) const`
- `OSCL_IMPORT_REF chartype operator[] (uint32 index) const`
- `virtual OSCL_IMPORT_REF chartype read (uint32 index) const`
- `virtual OSCL_IMPORT_REF uint32 setrep_to_char (const oscl_wchar *src, uint32 len, TOSCL_StringOp op, Oscl_DefAlloc *aAlloc)`
- `virtual OSCL_IMPORT_REF int8 hash () const`
- `virtual OSCL_IMPORT_REF void write (uint32 index, chartype c)`
- `virtual OSCL_IMPORT_REF void write (uint32 offset, uint32 length, const chartype *buf)`

Protected Methods

- OSCL_IMPORT_REF OSCL_String()
- virtual OSCL_IMPORT_REF ~OSCL_String()
- virtual void set_rep (const chartype *cstr)=0
- virtual void append_rep (const chartype *cstr)=0
- virtual void set_rep (const OSCL_String &src)=0
- virtual void append_rep (const OSCL_String &src)=0
- virtual void set_len (uint32 len)=0

7.73.1 Detailed Description

A common base class for string classes with "char" character format

7.73.2 Member Typedef Documentation

7.73.2.1 `typedef char OSCL_String::chartype`

Reimplemented in `OSCL_HeapString< Alloc >`, `OSCL_HeapStringA`, `OSCL_StackString< MaxBufSize >`, `OSCL_FastString`, and `OSCL_HeapString< OsclMemAllocator >`.

7.73.3 Constructor & Destructor Documentation

7.73.3.1 `OSCL_IMPORT_REF OSCL_String::OSCL_String()` [protected]

7.73.3.2 `virtual OSCL_IMPORT_REF OSCL_String::~OSCL_String()` [protected, virtual]

7.73.4 Member Function Documentation

7.73.4.1 `virtual void OSCL_String::append_rep (const OSCL_String & src)` [protected, pure virtual]

Append the input string to the current string. The string may be truncated to fit the available storage.

7.73.4.2 `virtual void OSCL_String::append_rep (const chartype * cstr)` [protected, pure virtual]

Append the input null-terminated string to the current string. The string may be truncated to fit the available storage.

7.73.4.3 `virtual const chartype* OSCL_String::get_cstr ()` [pure virtual]

This function returns the C-style string for read access.

Implemented in `OSCL_HeapString< Alloc >`, `OSCL_HeapStringA`, `OSCL_StackString< MaxBufSize >`, `OSCL_FastString`, and `OSCL_HeapString< OsclMemAllocator >`.

7.73.4.4 virtual uint32 OSCL_String::get_maxsize () [pure virtual]

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsclMemAllocator >](#).

7.73.4.5 virtual uint32 OSCL_String::get_size () [pure virtual]

This function returns the string size not including the null-terminator.

Implemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsclMemAllocator >](#).

7.73.4.6 virtual chartype* OSCL_String::get_str () [pure virtual]

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsclMemAllocator >](#).

7.73.4.7 virtual OSCL_IMPORT_REF int8 OSCL_String::hash () [virtual]

This function performs a hash operation on the string. If the string is not writable, the function leaves.

7.73.4.8 virtual OSCL_IMPORT_REF bool OSCL_String::is_writable () [virtual]

This function returns true if the string is writable.

7.73.4.9 OSCL_IMPORT_REF bool OSCL_String::operator!= (const OSCL_String & src) const**7.73.4.10 OSCL_IMPORT_REF OSCL_String& OSCL_String::operator+= (const chartype c)**

Append operator. This operator appends the input character to this object. The string may be truncated to fit available storage.

7.73.4.11 OSCL_IMPORT_REF OSCL_String& OSCL_String::operator+= (const chartype * cstr)

Append operator. This operator appends the input string to this object. The string may be truncated to fit available storage.

am: null-terminated string

7.73.4.12 OSCL_IMPORT_REF OSCL_String& OSCL_String::operator+=(const OSCL_String & src)

Append operator. This operator appends the input string to this object. The string may be truncated to fit available storage.

7.73.4.13 OSCL_IMPORT_REF bool OSCL_String::operator<(const OSCL_String & src) const

7.73.4.14 OSCL_IMPORT_REF bool OSCL_String::operator<=(const OSCL_String & src) const

7.73.4.15 OSCL_IMPORT_REF OSCL_String& OSCL_String::operator=(const chartype * cstr)

Assignment operator

am: null-terminated string

Reimplemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsclMemAllocator >](#).

7.73.4.16 OSCL_IMPORT_REF OSCL_String& OSCL_String::operator=(const OSCL_String & src)

Assignment operator

Reimplemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), and [OSCL_HeapString< OsclMemAllocator >](#).

7.73.4.17 OSCL_IMPORT_REF bool OSCL_String::operator==(const chartype * cstr) const

Comparison operator

am: null-terminated string

7.73.4.18 OSCL_IMPORT_REF bool OSCL_String::operator==(const OSCL_String & src) const

Comparison operators

7.73.4.19 OSCL_IMPORT_REF bool OSCL_String::operator>(const OSCL_String & src) const

7.73.4.20 OSCL_IMPORT_REF bool OSCL_String::operator>=(const OSCL_String & src) const

7.73.4.21]

OSCL_IMPORT_REF chartype OSCL_String::operator[] (uint32 index) const

This is subscript notation to access a character at the given position. If the index is outside the current size range then the function leaves.

7.73.4.22 virtual OSCL_IMPORT_REF `chartype` OSCL_String::read (uint32 *index*) const [virtual]

This function returns the character at the given position. If the index is outside the current size range then the function leaves.

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 OSCL_IMPORT_REF uint32 OSCL_String::setrep_to_char (const `oscl_wchar` * *src*, uint32 *len*, TOSCL_StringOp *op*, Oscl_DefAlloc * *aAlloc*) [virtual]

This function allocates a temp storage for performing one of the following operations based on TOSCL_StringOp

- compress src string from oscl_wchar to utf8.
- convert src string from oscl_wchar to utf8. call parent `set_rep()` to copy resulting string.

Parameters:

- src*: reference input string
- len*: length of string to operate on
- op*: type operation mentioned above
- aAlloc*: optional, memory allocator if available

Returns:

length of compressed or converted string exclude terminated '\0'.

7.73.4.27 virtual OSCL_IMPORT_REF void OSCL_String::write (uint32 *offset*, uint32 *length*, const `chartype` * *buf*) [virtual]

This function replaces characters at the specified offset within the current string. If the string is not writable, the function leaves. The characters may be truncated to fit the current storage.

Parameters:

- offset*: the offset into the existing string buffer
- length*: number of characters to copy.
- ptr*: character buffer, not necessarily null-terminated.

**7.73.4.28 virtual OSCL_IMPORT_REF void OSCL_String::write (uint32 index, chartype c)
[virtual]**

This function stores a character at the specified position. If the string is not writable, the function leaves. If the index is outside the current size range then the function leaves.

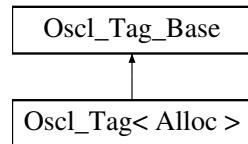
The documentation for this class was generated from the following file:

- [oscl_string.h](#)

7.74 Oscl_Tag< Alloc > Struct Template Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oscl_Tag< Alloc >::



Public Methods

- [Oscl_Tag](#) (const Oscl_Tag< Alloc > &t)
- [Oscl_Tag](#) (const [tag_base_type](#) &t)
- [~Oscl_Tag](#) ()
- bool [operator<](#) (const Oscl_Tag< Alloc > &x) const

Data Fields

- [Oscl_TAlloc< tag_base_unit, Alloc > tagAllocator](#)
- [tag_base_type tag](#)

```
template<class Alloc> struct Oscl_Tag< Alloc >
```

7.74.1 Constructor & Destructor Documentation

7.74.1.1 template<class Alloc> Oscl_Tag< Alloc >::Oscl_Tag (const Oscl_Tag< Alloc > & t)
`[inline]`

7.74.1.2 template<class Alloc> Oscl_Tag< Alloc >::Oscl_Tag (const tag_base_type & t)
`[inline]`

7.74.1.3 template<class Alloc> Oscl_Tag< Alloc >::~Oscl_Tag () `[inline]`

7.74.2 Member Function Documentation

7.74.2.1 template<class Alloc> bool Oscl_Tag< Alloc >::operator< (const Oscl_Tag< Alloc > & x) const `[inline]`

7.74.3 Field Documentation

7.74.3.1 template<class Alloc> tag_base_type Oscl_Tag< Alloc >::tag

7.74.3.2 template<class Alloc> Oscl_TAlloc<tag_base_unit, Alloc> Oscl_Tag< Alloc >::tagAllocator

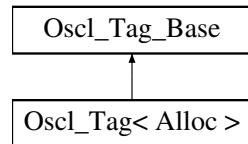
The documentation for this struct was generated from the following file:

- [oscl_tagtree.h](#)

7.75 Oscl_Tag_Base Struct Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oscl_Tag_Base::



Public Types

- `typedef char tag_base_unit`
- `typedef tag_base_unit * tag_base_type`
- `typedef uint32 size_type`

Public Methods

- `bool operator() (const tag_base_type &x, const tag_base_type &y) const`
- `size_type tag_len (const tag_base_type &t) const`
- `tag_base_type tag_copy (tag_base_type &dest, const tag_base_type &src) const`
- `int32 tag_cmp (const tag_base_type &x, const tag_base_type &y) const`
- `OSCL_IMPORT_REF tag_base_type tag_ancestor (tag_base_type &dest, const tag_base_type &src) const`
- `OSCL_IMPORT_REF size_type tag_depth (const tag_base_type &t) const`

7.75.1 Member Typedef Documentation

- 7.75.1.1 `typedef uint32 Oscl_Tag_Base::size_type`
- 7.75.1.2 `typedef tag_base_unit* Oscl_Tag_Base::tag_base_type`
- 7.75.1.3 `typedef char Oscl_Tag_Base::tag_base_unit`

7.75.2 Member Function Documentation

- 7.75.2.1 `bool Oscl_Tag_Base::operator() (const tag_base_type & x, const tag_base_type & y) const [inline]`
- 7.75.2.2 `OSCL_IMPORT_REF tag_base_type Oscl_Tag_Base::tag_ancestor (tag_base_type & dest, const tag_base_type & src) const`
- 7.75.2.3 `int32 Oscl_Tag_Base::tag_cmp (const tag_base_type & x, const tag_base_type & y) const [inline]`
- 7.75.2.4 `tag_base_type Oscl_Tag_Base::tag_copy (tag_base_type & dest, const tag_base_type & src) const [inline]`
- 7.75.2.5 `OSCL_IMPORT_REF size_type Oscl_Tag_Base::tag_depth (const tag_base_type & t) const`
- 7.75.2.6 `size_type Oscl_Tag_Base::tag_len (const tag_base_type & t) const [inline]`

The documentation for this struct was generated from the following file:

- [oscl_tagtree.h](#)

7.76 Oscl_TagTree< T, Alloc > Class Template Reference

```
#include <oscl_tagtree.h>
```

Public Types

- `typedef Oscl_Tag< Alloc > tag_type`
- `typedef tag_type::tag_base_type tag_base_type`
- `typedef Oscl_Vector< Node *, Alloc > children_type`
- `typedef Node node_type`
- `typedef node_type * node_ptr`
- `typedef Oscl_Map< const tag_base_type, node_ptr, Alloc, Oscl_Tag_Base > map_type`
- `typedef map_type::size_type size_type`
- `typedef map_type::value_type value_type`
- `typedef Oscl_Pair< iterator, bool > pair_iterator_bool`

Public Methods

- `Oscl_TagTree (size_type max_depth=0)`
- `Oscl_TagTree (const Oscl_TagTree< T, Alloc > &x)`
- `Oscl_TagTree< T, Alloc > & operator= (const Oscl_TagTree< T, Alloc > &x)`
- `~Oscl_TagTree ()`
- `iterator begin ()`
- `const_iterator begin () const`
- `iterator end ()`
- `const_iterator end () const`
- `bool empty () const`
- `size_type size () const`
- `T & operator[] (const tag_base_type &t)`
- `pair_iterator_bool insert (const tag_base_type &t, const T &x)`
- `void erase (iterator position)`
- `size_type erase (const tag_base_type &x)`
- `void clear ()`
- `iterator find (const tag_base_type &x)`
- `size_type count (const tag_base_type &x) const`

7.76.1 Detailed Description

`template<class T, class Alloc> class Oscl_TagTree< T, Alloc >`

Oscl_TagTree Class.

7.76.2 Member Typedef Documentation

- 7.76.2.1 template<class T, class Alloc> typedef **Oscl_Vector<Node*, Alloc>** Oscl_TagTree< T, Alloc >::children_type
- 7.76.2.2 template<class T, class Alloc> typedef **Oscl_Map<const tag_base_type, node_ptr, Alloc, Oscl_Tag_Base>** Oscl_TagTree< T, Alloc >::map_type
- 7.76.2.3 template<class T, class Alloc> typedef **node_type*** Oscl_TagTree< T, Alloc >::node_ptr
- 7.76.2.4 template<class T, class Alloc> typedef **Node** Oscl_TagTree< T, Alloc >::node_type
- 7.76.2.5 template<class T, class Alloc> typedef **Oscl_Pair<iterator, bool>** Oscl_TagTree< T, Alloc >::pair_iterator_bool
- 7.76.2.6 template<class T, class Alloc> typedef map_type::size_type Oscl_TagTree< T, Alloc >::size_type
- 7.76.2.7 template<class T, class Alloc> typedef tag_type::tag_base_type Oscl_TagTree< T, Alloc >::tag_base_type
- 7.76.2.8 template<class T, class Alloc> typedef **Oscl_Tag<Alloc>** Oscl_TagTree< T, Alloc >::tag_type
- 7.76.2.9 template<class T, class Alloc> typedef map_type::value_type Oscl_TagTree< T, Alloc >::value_type

7.76.3 Constructor & Destructor Documentation

- 7.76.3.1 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::Oscl_TagTree (**size_type max_depth = 0**) [inline]

Creates a tag tree with only a root node with tag ""

- 7.76.3.2 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::Oscl_TagTree (const Oscl_TagTree< T, Alloc > & x) [inline]

Copy constructor

- 7.76.3.3 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::~Oscl_TagTree () [inline]

Destructor

7.76.4 Member Function Documentation

- 7.76.4.1 template<class T, class Alloc> const_iterator Oscl_TagTree< T, Alloc >::begin () const [inline]

Returns an iterator pointing to the first node in the tree.

7.76.4.2 template<class T, class Alloc> iterator Oscl_TagTree< T, Alloc >::begin () [inline]

Returns an iterator pointing to the first node in the tree.

7.76.4.3 template<class T, class Alloc> void Oscl_TagTree< T, Alloc >::clear () [inline]

Erases the entire tag tree.

7.76.4.4 template<class T, class Alloc> size_type Oscl_TagTree< T, Alloc >::count (const tag_base_type & x) const [inline]

Returns the number of elements with key x. This can only be 0 or 1..

7.76.4.5 template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::empty () const [inline]

Returns true if tree size is 0

7.76.4.6 template<class T, class Alloc> const_iterator Oscl_TagTree< T, Alloc >::end () const [inline]

Returns a const iterator pointing to the end of the tree.

7.76.4.7 template<class T, class Alloc> iterator Oscl_TagTree< T, Alloc >::end () [inline]

Returns an iterator pointing to the end of the tree.

7.76.4.8 template<class T, class Alloc> size_type Oscl_TagTree< T, Alloc >::erase (const tag_base_type & x) [inline]

Erases the node with tag x. If the node has children, then the node will not be erased from the tree. It will be replaced with the default node value

Parameters:

x Tag of node to erase

Returns:

Returns the number of nodes erased. Since one-to-one mapping between nodes and tags, this will be either 0 or 1

7.76.4.9 template<class T, class Alloc> void Oscl_TagTree< T, Alloc >::erase (iterator position) [inline]

Erases the element pointed to by the iterator. If the node has children, then the node will not be erased from the tree. It will be replaced with the default node value.

Parameters:

position Iterator pointing to the node to be erased

7.76.4.10 template<class T, class Alloc> iterator Oscl_TagTree< T, Alloc >::find (const tag_base_type & x) [inline]

Finds an element whose key is x

Returns:

returns an iterator to the element with key x. If no element is found, returns [end\(\)](#)

7.76.4.11 template<class T, class Alloc> pair_iterator_bool Oscl_TagTree< T, Alloc >::insert (const tag_base_type & t, const T & x) [inline]

Inserts x into the tree and associates it with tag t. If the tag already exists x will not be inserted, and an iterator pointing to the existing node with tag t is returned.

Parameters:

t tag to use

x element to insert

Returns:

Returns a pair of parameters, iterator and bool. The iterator points to the inserted node containing x. If the tag t already existed, then the iterator points to the node associated with tag t. The bool is true if x was inserted and false if it was not inserted due to an existing node with tag t.

7.76.4.12 template<class T, class Alloc> Oscl_TagTree<T, Alloc>& Oscl_TagTree< T, Alloc >::operator= (const Oscl_TagTree< T, Alloc > & x) [inline]

Assignment operator

7.76.4.13]

template<class T, class Alloc> T& Oscl_TagTree< T, Alloc >::operator[] (const tag_base_type & t) [inline]

Returns a reference to the object that is associated with a particular tag. If the map does not already contain such an object, operator[] inserts the default object T().

7.76.4.14 template<class T, class Alloc> size_type Oscl_TagTree< T, Alloc >::size () const [inline]

Returns the number of nodes stored in the tree

The documentation for this class was generated from the following file:

- [oscl_tagtree.h](#)

7.77 Oscl_TagTree< T, Alloc >::const_iterator Struct Reference

```
#include <oscl_tagtree.h>
```

Public Types

- `typedef const node_type & reference`
- `typedef const node_type * pointer`
- `typedef map_type::const_iterator mapiter`
- `typedef const_iterator self`

Public Methods

- `const_iterator()`
- `const_iterator(mapiter x)`
- `const_iterator(const const_iterator &it)`
- `reference operator*() const`
- `pointer operator->() const`
- `bool operator==(const self &x)`
- `bool operator!=(const self &x)`
- `self & operator++()`
- `self operator++(int)`
- `self & operator--()`
- `self operator--(int)`

Data Fields

- `mapiter mapit`

template<class T, class Alloc> struct Oscl_TagTree< T, Alloc >::const_iterator

7.77.1 Member Typedef Documentation

- 7.77.1.1 template<class T, class Alloc> typedef map_type::const_iterator Oscl_TagTree< T, Alloc >::const_iterator::mapiter
- 7.77.1.2 template<class T, class Alloc> typedef const node_type* Oscl_TagTree< T, Alloc >::const_iterator::pointer
- 7.77.1.3 template<class T, class Alloc> typedef const node_type& Oscl_TagTree< T, Alloc >::const_iterator::reference
- 7.77.1.4 template<class T, class Alloc> typedef const_iterator Oscl_TagTree< T, Alloc >::const_iterator::self

7.77.2 Constructor & Destructor Documentation

- 7.77.2.1 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::const_iterator::const_iterator() [inline]
- 7.77.2.2 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::const_iterator::const_iterator(mapiter x) [inline]
- 7.77.2.3 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::const_iterator::const_iterator(const const_iterator & it) [inline]

7.77.3 Member Function Documentation

- 7.77.3.1 template<class T, class Alloc> reference Oscl_TagTree< T, Alloc >::const_iterator::operator *() const [inline]
- 7.77.3.2 template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::const_iterator::operator!= (const self & x) [inline]
- 7.77.3.3 template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::const_iterator::operator++(int) [inline]
- 7.77.3.4 template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::const_iterator::operator++() [inline]
- 7.77.3.5 template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::const_iterator::operator-(int) [inline]
- 7.77.3.6 template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::const_iterator::operator-() [inline]
- 7.77.3.7 template<class T, class Alloc> pointer Oscl_TagTree< T, Alloc >::const_iterator::operator -() const [inline]
- 7.77.3.8 template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::const_iterator::operator==(const self & x) [inline]

7.77.4 Field Documentation

- [oscl_tagtree.h](#)

7.78 Oscl_TagTree< T, Alloc >::iterator Struct Reference

```
#include <oscl_tagtree.h>
```

Public Types

- `typedef node_type & reference`
- `typedef node_type * pointer`
- `typedef map_type::iterator mapiter`
- `typedef iterator self`

Public Methods

- `iterator ()`
- `iterator (mapiter x)`
- `iterator (const iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator== (const self &x)`
- `bool operator!= (const self &x)`
- `self & operator++ ()`
- `self operator++ (int)`
- `self & operator– ()`
- `self operator– (int)`

Data Fields

- `mapiter mapit`

template<class T, class Alloc> struct Oscl_TagTree< T, Alloc >::iterator

7.78.1 Member Typedef Documentation

- 7.78.1.1 template<class T, class Alloc> typedef map_type::iterator Oscl_TagTree< T, Alloc >::iterator::mapiter
- 7.78.1.2 template<class T, class Alloc> typedef node_type* Oscl_TagTree< T, Alloc >::iterator::pointer
- 7.78.1.3 template<class T, class Alloc> typedef node_type& Oscl_TagTree< T, Alloc >::iterator::reference
- 7.78.1.4 template<class T, class Alloc> typedef iterator Oscl_TagTree< T, Alloc >::iterator::self

7.78.2 Constructor & Destructor Documentation

- 7.78.2.1 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::iterator::iterator () [inline]
- 7.78.2.2 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::iterator::iterator (mapiter x) [inline]
- 7.78.2.3 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::iterator::iterator (const iterator & it) [inline]

7.78.3 Member Function Documentation

- 7.78.3.1 template<class T, class Alloc> reference Oscl_TagTree< T, Alloc >::iterator::operator * () const [inline]
- 7.78.3.2 template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::iterator::operator!= (const self & x) [inline]
- 7.78.3.3 template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::iterator::operator++ (int) [inline]
- 7.78.3.4 template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::iterator::operator++ () [inline]
- 7.78.3.5 template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::iterator::operator- (int) [inline]
- 7.78.3.6 template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::iterator::operator- () [inline]
- 7.78.3.7 template<class T, class Alloc> pointer Oscl_TagTree< T, Alloc >::iterator::operator → () const [inline]
- 7.78.3.8 template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::iterator::operator== (const self & x) [inline]

7.78.4 Field Documentation

-
- 7.78.4.1 template<class T, class Alloc> mapiter Oscl_TagTree< T, Alloc >::iterator::mapit

- [oscl_tagtree.h](#)

7.79 Oscl_TagTree< T, Alloc >::Node Struct Reference

```
#include <oscl_tagtree.h>
```

Public Types

- `typedef Oscl_Vector< Node *, Alloc > children_type`

Public Methods

- `Node()`
- `void sort_children()`
- `tag_type::size_type depth()`

Data Fields

- `tag_type tag`
- `T value`
- `Node * parent`
- `children_type children`

template<class T, class Alloc> struct Oscl_TagTree< T, Alloc >::Node

7.79.1 Member Typedef Documentation

7.79.1.1 template<class T, class Alloc> typedef Oscl_Vector<Node*, Alloc> Oscl_TagTree< T, Alloc >::Node::children_type

7.79.2 Constructor & Destructor Documentation

7.79.2.1 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::Node::Node () [inline]

7.79.3 Member Function Documentation

7.79.3.1 template<class T, class Alloc> tag_type::size_type Oscl_TagTree< T, Alloc >::Node::depth () [inline]

7.79.3.2 template<class T, class Alloc> void Oscl_TagTree< T, Alloc >::Node::sort_children () [inline]

7.79.4 Field Documentation

7.79.4.1 template<class T, class Alloc> children_type Oscl_TagTree< T, Alloc >::Node::children

7.79.4.2 template<class T, class Alloc> Node* Oscl_TagTree< T, Alloc >::Node::parent

7.79.4.3 template<class T, class Alloc> tag_type Oscl_TagTree< T, Alloc >::Node::tag

7.79.4.4 template<class T, class Alloc> T Oscl_TagTree< T, Alloc >::Node::value

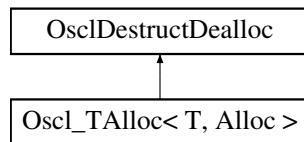
The documentation for this struct was generated from the following file:

- [oscl_tagtree.h](#)

7.80 Oscl_TAlloc< T, Alloc > Class Template Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oscl_TAlloc< T, Alloc >::



Public Types

- typedef T [value_type](#)
- typedef T * [pointer](#)
- typedef const T * [const_pointer](#)
- typedef uint32 [size_type](#)
- typedef T & [reference](#)
- typedef const T & [const_reference](#)

Public Methods

- virtual [~Oscl_TAlloc \(\)](#)
- [pointer allocate_fl \(uint32 size, const char *file_name, const int line_num\)](#)
- [pointer allocate \(uint32 size\)](#)
- [pointer alloc_and_construct_fl \(const_reference val, const char *file_name, const int line_num\)](#)
- [pointer alloc_and_construct \(const_reference val\)](#)
- void [deallocate \(OsclAny *p\)](#)
- void [deallocate \(OsclAny *p, size_type n\)](#)
- void [destruct_and_dealloc \(OsclAny *p\)](#)
- [pointer address \(reference r\)](#)
- [const_pointer address \(const_reference r\) const](#)
- void [construct \(pointer p, const_reference val\)](#)
- void [destroy \(pointer p\)](#)

template<class T, class Alloc> class Oscl_TAlloc< T, Alloc >

7.80.1 Member Typedef Documentation

7.80.1.1 template<class T, class Alloc> **typedef const T* Oscl_TAlloc< T, Alloc >::const_pointer**

7.80.1.2 template<class T, class Alloc> **typedef const T& Oscl_TAlloc< T, Alloc >::const_reference**

7.80.1.3 template<class T, class Alloc> **typedef T* Oscl_TAlloc< T, Alloc >::pointer**

7.80.1.4 template<class T, class Alloc> **typedef T& Oscl_TAlloc< T, Alloc >::reference**

7.80.1.5 template<class T, class Alloc> **typedef uint32 Oscl_TAlloc< T, Alloc >::size_type**

7.80.1.6 template<class T, class Alloc> **typedef T Oscl_TAlloc< T, Alloc >::value_type**

7.80.2 Constructor & Destructor Documentation

7.80.2.1 template<class T, class Alloc> **virtual Oscl_TAlloc< T, Alloc >::~Oscl_TAlloc ()**
[inline, virtual]

7.80.3 Member Function Documentation

7.80.3.1 template<class T, class Alloc> **const_pointer Oscl_TAlloc< T, Alloc >::address (const_reference r) const** [inline]

7.80.3.2 template<class T, class Alloc> **pointer Oscl_TAlloc< T, Alloc >::address (reference r)**
[inline]

7.80.3.3 template<class T, class Alloc> **pointer Oscl_TAlloc< T, Alloc >::alloc_and_construct (const_reference val)** [inline]

7.80.3.4 template<class T, class Alloc> **pointer Oscl_TAlloc< T, Alloc >::alloc_and_construct_fl (const_reference val, const char *file_name, const int line_num)** [inline]

7.80.3.5 template<class T, class Alloc> **pointer Oscl_TAlloc< T, Alloc >::allocate (uint32 size)**
[inline]

7.80.3.6 template<class T, class Alloc> **pointer Oscl_TAlloc< T, Alloc >::allocate_fl (uint32 size, const char *file_name, const int line_num)** [inline]

7.80.3.7 template<class T, class Alloc> **void Oscl_TAlloc< T, Alloc >::construct (pointer p, const_reference val)** [inline]

7.80.3.8 template<class T, class Alloc> **void Oscl_TAlloc< T, Alloc >::deallocate (OsclAny *p, size_type n)** [inline]

7.80.3.9 template<class T, class Alloc> **void Oscl_TAlloc< T, Alloc >::deallocate (OsclAny *p)**
[inline]

7.80.3.10 template<class T, class Alloc> **void Oscl_TAlloc< T, Alloc >::destroy (pointer p)**
[inline]

7.80.3.11 template<class T, class Alloc> **void Oscl_TAlloc< T, Alloc >::destruct_and_dealloc (OsclAny *p)** [inline, virtual]

The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

7.81 Oscl_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference

```
#include <oscl_defalloc.h>
```

Public Types

- `typedef Oscl_TAlloc< U, V > other`

```
template<class T, class Alloc>template<class U, class V> struct Oscl_TAlloc< T, Alloc >::rebind< U, V >
```

7.81.1 Member Typedef Documentation

```
7.81.1.1 template<class T, class Alloc> template<class U, class V> typedef Oscl_TAlloc<U, V>  
Oscl_TAlloc< T, Alloc >::rebind< U, V >::other
```

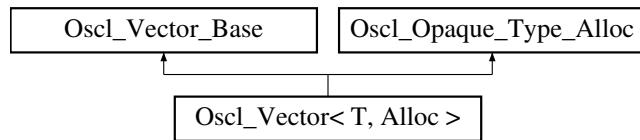
The documentation for this struct was generated from the following file:

- `oscl_defalloc.h`

7.82 Oscl_Vector< T, Alloc > Class Template Reference

```
#include <oscl_vector.h>
```

Inheritance diagram for Oscl_Vector< T, Alloc >::



Public Types

- `typedef T value_type`
- `typedef T * pointer`
- `typedef T & reference`
- `typedef const T & const_reference`
- `typedef T * iterator`
- `typedef const T * const_iterator`

Public Methods

- `Oscl_Vector()`
- `Oscl_Vector(uint32 n)`
- `Oscl_Vector(const Oscl_Vector< T, Alloc > &x)`
- `virtual ~Oscl_Vector()`
- `Oscl_Vector< T, Alloc > & operator=(const Oscl_Vector< T, Alloc > &x)`
- `void push_back(const T &x)`
- `void push_front(const T &x)`
- `iterator insert(iterator pos, const T &x)`
- `T & operator[](uint32 n)`
- `const T & operator[](uint32 n) const`
- `T & front()`
- `const T & front() const`
- `T & back()`
- `const T & back() const`
- `void pop_back()`
- `void clear()`
- `void destroy()`
- `iterator begin() const`
- `iterator end() const`
- `iterator erase(iterator pos)`
- `iterator erase(iterator first, iterator last)`

7.82.1 Detailed Description

template<class T, class Alloc> class Oscl_Vector< T, Alloc >

Oscl_Vector Class. A subset of STL::Vector methods. Oscl_Vector supports random access to elements, constant time insertion and removal of elements at the end of the vector, and linear time insertion and removal of elements at the beginning or middle of the vector. The number of elements in a vector can vary dynamically, and memory management is performed automatically.

7.82.2 Member Typedef Documentation

7.82.2.1 template<class T, class Alloc> typedef const T* Oscl_Vector< T, Alloc >::const_iterator

7.82.2.2 template<class T, class Alloc> typedef const T& Oscl_Vector< T, Alloc >::const_reference

7.82.2.3 template<class T, class Alloc> typedef T* Oscl_Vector< T, Alloc >::iterator

7.82.2.4 template<class T, class Alloc> typedef T* Oscl_Vector< T, Alloc >::pointer

7.82.2.5 template<class T, class Alloc> typedef T& Oscl_Vector< T, Alloc >::reference

7.82.2.6 template<class T, class Alloc> typedef T Oscl_Vector< T, Alloc >::value_type

7.82.3 Constructor & Destructor Documentation

7.82.3.1 template<class T, class Alloc> Oscl_Vector< T, Alloc >::Oscl_Vector () [inline]

Creates an empty vector.

7.82.3.2 template<class T, class Alloc> Oscl_Vector< T, Alloc >::Oscl_Vector (uint32 n) [inline]

Creates an empty vector with capacity n.

Parameters:

n creates a vector with n elements. The main reason for specifying n is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

7.82.3.3 template<class T, class Alloc> Oscl_Vector< T, Alloc >::Oscl_Vector (const Oscl_Vector< T, Alloc > & x) [inline]

Copy Constructor.

Parameters:

x vector class to copy.

**7.82.3.4 template<class T, class Alloc> virtual Oscl_Vector< T, Alloc >::~Oscl_Vector ()
[inline, virtual]**

The destructor.

7.82.4 Member Function Documentation

**7.82.4.1 template<class T, class Alloc> const T& Oscl_Vector< T, Alloc >::back () const
[inline]**

Returns the last element.

7.82.4.2 template<class T, class Alloc> T& Oscl_Vector< T, Alloc >::back () [inline]

Returns the last element.

**7.82.4.3 template<class T, class Alloc> iterator Oscl_Vector< T, Alloc >::begin () const
[inline]**

Returns an iterator pointing to the beginning of the vector.

Reimplemented from [Oscl_Vector_Base](#).

7.82.4.4 template<class T, class Alloc> void Oscl_Vector< T, Alloc >::clear () [inline]

Removes all elements.

7.82.4.5 template<class T, class Alloc> void Oscl_Vector< T, Alloc >::destroy () [inline]

Destroy – this is like an explicit destructor call.

Reimplemented from [Oscl_Vector_Base](#).

**7.82.4.6 template<class T, class Alloc> iterator Oscl_Vector< T, Alloc >::end () const
[inline]**

Returns an iterator pointing to the end of the vector..

Reimplemented from [Oscl_Vector_Base](#).

**7.82.4.7 template<class T, class Alloc> iterator Oscl_Vector< T, Alloc >::erase (iterator *first*,
iterator *last*) [inline]**

Erases elements in range [*first*, *last*). Erasing an element invalidates all iterators pointing to elements following the deletion point.

Parameters:

first starting position

last ending position, this position is not erased

7.82.4.8 template<class T, class Alloc> iterator Oscl_Vector< T, Alloc >::erase (iterator pos) [inline]

Erases the element pointed to by iterator pos. Erasing an element invalidates all iterators pointing to elements following the deletion point.

Parameters:

pos iterator at erase position

7.82.4.9 template<class T, class Alloc> const T& Oscl_Vector< T, Alloc >::front () const [inline]

Returns the first element.

7.82.4.10 template<class T, class Alloc> T& Oscl_Vector< T, Alloc >::front () [inline]

Returns the first element.

7.82.4.11 template<class T, class Alloc> iterator Oscl_Vector< T, Alloc >::insert (iterator pos, const T & x) [inline]

Inserts a new element before the one at pos.

Parameters:

pos position at which to insert the new element.

x new element

7.82.4.12 template<class T, class Alloc> Oscl_Vector<T, Alloc>& Oscl_Vector< T, Alloc >::operator= (const Oscl_Vector< T, Alloc > & x) [inline]

The assignment operator

7.82.4.13]

template<class T, class Alloc> const T& Oscl_Vector< T, Alloc >::operator[] (uint32 n) const [inline]

Returns the n'th element.

Parameters:

n element position to return

7.82.4.14]

template<class T, class Alloc> T& Oscl_Vector< T, Alloc >::operator[] (uint32 n) [inline]

Returns the n'th element.

Parameters:

n element position to return

7.82.4.15 template<class T, class Alloc> void Oscl_Vector< T, Alloc >::pop_back () [inline]

Removes the last element.

Reimplemented from [Oscl_Vector_Base](#).

7.82.4.16 template<class T, class Alloc> void Oscl_Vector< T, Alloc >::push_back (const T & x) [inline]

Inserts a new element at the end. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

Parameters:

x new element

7.82.4.17 template<class T, class Alloc> void Oscl_Vector< T, Alloc >::push_front (const T & x) [inline]

Inserts a new element at the front. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

Parameters:

x new element

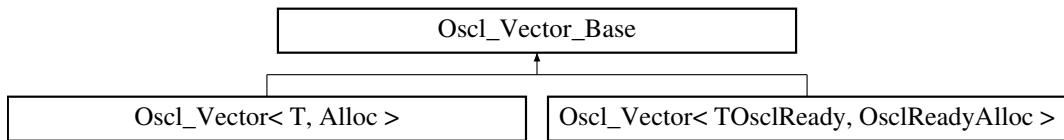
The documentation for this class was generated from the following file:

- [oscl_vector.h](#)

7.83 Oscl_Vector_Base Class Reference

```
#include <oscl_vector.h>
```

Inheritance diagram for Oscl_Vector_Base::



Public Methods

- uint32 [size \(\) const](#)
- uint32 [capacity \(\) const](#)
- bool [empty \(\) const](#)
- OSCL_IMPORT_REF void [reserve \(uint32 n\)](#)

Protected Methods

- OSCL_IMPORT_REF void [construct \(Oscl_Opaque_Type_Alloc *aType\)](#)
- OSCL_IMPORT_REF void [construct \(Oscl_Opaque_Type_Alloc *aType, uint32 n\)](#)
- OSCL_IMPORT_REF void [construct \(Oscl_Opaque_Type_Alloc *aType, const Oscl_Vector_Base &x\)](#)
- virtual [~Oscl_Vector_Base \(\)](#)
- OSCL_IMPORT_REF void [push_back \(const OsclAny *x\)](#)
- OSCL_IMPORT_REF void [pop_back \(\)](#)
- OSCL_IMPORT_REF void [push_front \(const OsclAny *x\)](#)
- OSCL_IMPORT_REF OsclAny * [insert \(OsclAny *pos, const OsclAny *x\)](#)
- OSCL_IMPORT_REF OsclAny * [erase \(OsclAny *pos\)](#)
- OSCL_IMPORT_REF OsclAny * [erase \(OsclAny *first, OsclAny *last\)](#)
- OSCL_IMPORT_REF void [assign_vector \(const Oscl_Vector_Base &x\)](#)
- OSCL_IMPORT_REF void [destroy \(\)](#)

Protected Attributes

- uint32 [numelems](#)
- uint32 [bufsize](#)
- [OsclAny * elems](#)
- uint32 [sizeof_T](#)

Friends

- class [OsclPriorityQueueBase](#)

7.83.1 Detailed Description

Oscl_Vector_Base is a non-templatized base class for [Oscl_Vector](#). The purpose of this base class is to avoid large inline routines in the [Oscl_Vector](#) implementation. This class is not intended for direct instantiation except by [Oscl_Vector](#).

7.83.2 Constructor & Destructor Documentation

7.83.2.1 virtual Oscl_Vector_Base::~Oscl_Vector_Base () [inline, protected, virtual]

The destructor.

7.83.3 Member Function Documentation

7.83.3.1 OSCL_IMPORT_REF void Oscl_Vector_Base::assign_vector (const Oscl_Vector_Base & x) [protected]

7.83.3.2 uint32 Oscl_Vector_Base::capacity () const [inline]

Returns the allocated memory of the vector in units of number of elements.

7.83.3.3 OSCL_IMPORT_REF void Oscl_Vector_Base::construct (Oscl_Opaque_Type_Alloc * aType, const Oscl_Vector_Base & x) [protected]

7.83.3.4 OSCL_IMPORT_REF void Oscl_Vector_Base::construct (Oscl_Opaque_Type_Alloc * aType, uint32 n) [protected]

7.83.3.5 OSCL_IMPORT_REF void Oscl_Vector_Base::construct (Oscl_Opaque_Type_Alloc * aType) [protected]

7.83.3.6 OSCL_IMPORT_REF void Oscl_Vector_Base::destroy () [protected]

Reimplemented in [Oscl_Vector< T, Alloc >](#), [Oscl_Vector< OsclComponentRegistryElement, OsclMemAllocator >](#), [Oscl_Vector< uint32, OsclMemAllocator >](#), [Oscl_Vector< OsclSocketServRequestQElem, OsclMemAllocator >](#), [Oscl_Vector< Node *, Alloc >](#), [Oscl_Vector< OsclFixedCacheParam, OsclMemAllocator >](#), [Oscl_Vector< OsclSocketRequest *, OsclMemAllocator >](#), [Oscl_Vector< entry_type *, Alloc >](#), [Oscl_Vector< OSCL_HeapString< OsclMemAllocator >, OsclMemAllocator >](#), [Oscl_Vector< OsclAsyncFileBuffer *, OsclMemAllocator >](#), [Oscl_Vector< TOsclFileOffset, OsclMemAllocator >](#), [Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator >](#), [Oscl_Vector< OsclSharedPtr< PVLoggerFilter >, alloc_type >](#), [Oscl_Vector< TOsclReady, OsclReadyAlloc >](#), [Oscl_Vector< OsclFileCacheBuffer, OsclMemAllocator >](#), [Oscl_Vector< OsclSharedPtr< PVLoggerAppender >, alloc_type >](#), [Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >](#), and [Oscl_Vector< OsclAny *, OsclMemAllocator >](#).

7.83.3.7 bool Oscl_Vector_Base::empty () const [inline]

True if the vector's size is 0.

7.83.3.8 OSCL_IMPORT_REF OsclAny* Oscl_Vector_Base::erase (OsclAny *first, OsclAny *last) [protected]

Erases elements in range [first, last). Erasing an element invalidates all iterators pointing to elements following the deletion point.

Parameters:

first starting position

last ending position, this position is not erased

7.83.3.9 OSCL_IMPORT_REF OsclAny* Oscl_Vector_Base::erase (OsclAny *pos) [protected]

Erases the element pointed to by iterator pos. Erasing an element invalidates all iterators pointing to elements following the deletion point.

Parameters:

pos iterator at erase position

7.83.3.10 OSCL_IMPORT_REF OsclAny* Oscl_Vector_Base::insert (OsclAny *pos, const OsclAny *x) [protected]

Inserts a new element at a specific position.

Parameters:

pos iterator at insert position.

x pointer to new element

7.83.3.11 OSCL_IMPORT_REF void Oscl_Vector_Base::pop_back () [protected]

Removes the last element.

Reimplemented in `Oscl_Vector< T, Alloc >`, `Oscl_Vector< OsclComponentRegistryElement, OsclMemAllocator >`, `Oscl_Vector< uint32, OsclMemAllocator >`, `Oscl_Vector< OsclSocketServRequestQElem, OsclMemAllocator >`, `Oscl_Vector< Node *, Alloc >`, `Oscl_Vector< OsclFixedCacheParam, OsclMemAllocator >`, `Oscl_Vector< OsclSocketRequest *, OsclMemAllocator >`, `Oscl_Vector< entry_type *, Alloc >`, `Oscl_Vector< OSCL_HeapString< OsclMemAllocator >, OsclMemAllocator >`, `Oscl_Vector< OsclAsyncFileBuffer *, OsclMemAllocator >`, `Oscl_Vector< TOscIFileOffset, OsclMemAllocator >`, `Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator >`, `Oscl_Vector< OsclSharedPtr< PVLoggerFilter >, alloc_type >`, `Oscl_Vector< TOscIReady, OsclReadyAlloc >`, `Oscl_Vector< OsclFileCacheBuffer, OsclMemAllocator >`, `Oscl_Vector< OsclSharedPtr< PVLoggerAppender >, alloc_type >`, `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >`, and `Oscl_Vector< OsclAny *, OsclMemAllocator >`.

7.83.3.12 OSCL_IMPORT_REF void Oscl_Vector_Base::push_back (const OsclAny *x) [protected]

Inserts a new element at the end. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

Parameters:

x pointer to the new element

7.83.3.13 OSCL_IMPORT_REF void Oscl_Vector_Base::push_front (const OsclAny * *x*) [protected]

Inserts a new element at the front. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

Parameters:

x pointer to new element

7.83.3.14 OSCL_IMPORT_REF void Oscl_Vector_Base::reserve (uint32 *n*)

Reallocates memory if necessary to a capacity of *n* elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

Parameters:

n size of vector

7.83.3.15 uint32 Oscl_Vector_Base::size () const [inline]

Returns the size of the vector in units of number of elements.

7.83.4 Friends And Related Function Documentation

7.83.4.1 friend class OsclPriorityQueueBase [friend]

7.83.5 Field Documentation

7.83.5.1 uint32 Oscl_Vector_Base::bufsize [protected]**7.83.5.2 OsclAny* Oscl_Vector_Base::elems [protected]****7.83.5.3 uint32 Oscl_Vector_Base::numelems [protected]****7.83.5.4 uint32 Oscl_Vector_Base::sizeof_T [protected]**

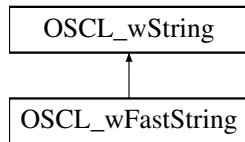
The documentation for this class was generated from the following file:

- [oscl_vector.h](#)

7.84 OSCL_wFastString Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wFastString::



Public Types

- `typedef OSCL_wString::chartype chartype`
- `typedef TOSCL_wStringOp optype`
- `typedef OSCL_String::chartype other_chartype`

Public Methods

- `OSCL_IMPORT_REF OSCL_wFastString()`
- `OSCL_IMPORT_REF OSCL_wFastString(const OSCL_wFastString &src)`
- `OSCL_IMPORT_REF OSCL_wFastString(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_wFastString(chartype *buf, uint32 maxlen)`
- `OSCL_IMPORT_REF ~OSCL_wFastString()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_wFastString & operator=(const OSCL_wFastString &src)`
- `OSCL_IMPORT_REF OSCL_wFastString & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(chartype *cstr, uint32 maxlen)`
- `OSCL_IMPORT_REF void set(const other_chartype *buf, uint32 numofbyte, optype op)`
- `OSCL_IMPORT_REF void set_length()`

Friends

- class `OSCL_wString`

7.84.1 Detailed Description

`OSCL_wFastString` is identical to `OSCL_FastString` except that it uses wide-character format. For descriptions, see `OSCL_FastString`.

7.84.2 Member Typedef Documentation

7.84.2.1 `typedef OSCL_wString::chartype OSCL_wFastString::chartype`

Reimplemented from `OSCL_wString`.

7.84.2.2 **typedef TOSCL_wStringOp OSCL_wFastString::optype**

7.84.2.3 **typedef OSCL_String::chartype OSCL_wFastString::other_chartype**

7.84.3 Constructor & Destructor Documentation

7.84.3.1 **OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString()**

7.84.3.2 **OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString(const OSCL_wFastString & src)**

7.84.3.3 **OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString(const chartype * cstr)**

7.84.3.4 **OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString(chartype * buf, uint32 maxlen)**

7.84.3.5 **OSCL_IMPORT_REF OSCL_wFastString::~OSCL_wFastString()**

7.84.4 Member Function Documentation

7.84.4.1 **OSCL_IMPORT_REF const chartype* OSCL_wFastString::get_cstr() [virtual]**

Implements [OSCL_wString](#).

7.84.4.2 **OSCL_IMPORT_REF uint32 OSCL_wFastString::get_maxsize() [virtual]**

Implements [OSCL_wString](#).

7.84.4.3 **OSCL_IMPORT_REF uint32 OSCL_wFastString::get_size() [virtual]**

Implements [OSCL_wString](#).

7.84.4.4 **OSCL_IMPORT_REF chartype* OSCL_wFastString::get_str() [virtual]**

Implements [OSCL_wString](#).

7.84.4.5 **OSCL_IMPORT_REF OSCL_wFastString& OSCL_wFastString::operator=(const chartype * cstr)**

Reimplemented from [OSCL_wString](#).

- 7.84.4.6 **OSCL_IMPORT_REF OSCL_wFastString& OSCL_wFastString::operator= (const OSCL_wFastString & src)**
- 7.84.4.7 **OSCL_IMPORT_REF void OSCL_wFastString::set (const other_chartype * buf, uint32 numofbyte, optype op)**
- 7.84.4.8 **OSCL_IMPORT_REF void OSCL_wFastString::set (chartype * cstr, uint32 maxlen)**
- 7.84.4.9 **OSCL_IMPORT_REF void OSCL_wFastString::set_length ()**

7.84.5 Friends And Related Function Documentation

- 7.84.5.1 **friend class OSCL_wString [friend]**

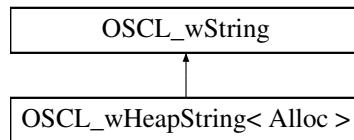
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.85 OSCL_wHeapString< Alloc > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wHeapString< Alloc >::



Public Types

- `typedef OSCL_wString::chartype chartype`
- `typedef TOSCL_wStringOp otype`
- `typedef OSCL_String::chartype other_chartype`

Public Methods

- `OSCL_wHeapString()`
- `OSCL_wHeapString(const OSCL_wHeapString &src)`
- `OSCL_wHeapString(const OSCL_wString &src)`
- `OSCL_wHeapString(const chartype *cstr)`
- `OSCL_wHeapString(const chartype *buf, uint32 length)`
- `~OSCL_wHeapString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype * get_cstr() const`
- `chartype * get_str() const`
- `OSCL_wHeapString & operator=(const OSCL_wHeapString &src)`
- `OSCL_wHeapString & operator=(const OSCL_wString &src)`
- `OSCL_wHeapString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`
- `void set(const other_chartype *buf, otype op)`
- `void set(const other_chartype *buf, uint32 length, otype op)`

Friends

- class `OSCL_wString`

7.85.1 Detailed Description

```
template<class Alloc> class OSCL_wHeapString< Alloc >
```

`OSCL_wHeapString` is identical to `OSCL_HeapString` except that it uses wide-character format. For descriptions, see `OSCL_HeapString`.

7.85.2 Member Typedef Documentation

7.85.2.1 template<class Alloc> typedef OSCL_wString::chartype OSCL_wHeapString< Alloc >::chartype

Reimplemented from [OSCL_wString](#).

7.85.2.2 template<class Alloc> typedef TOSCL_wStringOp OSCL_wHeapString< Alloc >::optype

7.85.2.3 template<class Alloc> typedef OSCL_String::chartype OSCL_wHeapString< Alloc >::other_chartype

7.85.3 Friends And Related Function Documentation

7.85.3.1 template<class Alloc> friend class OSCL_wString [friend]

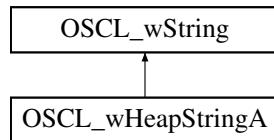
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.86 OSCL_wHeapStringA Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wHeapStringA::



Public Types

- `typedef OSCL_wString::chartype chartype`
- `typedef TOSCL_wStringOp optype`
- `typedef OSCL_String::chartype other_chartype`

Public Methods

- `OSCL_IMPORT_REF OSCL_wHeapStringA()`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wHeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wHeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wString &src, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const chartype *cstr, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const chartype *buf, uint32 length, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF ~OSCL_wHeapStringA()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const OSCL_wHeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(const chartype *buf, uint32 length)`
- `OSCL_IMPORT_REF void set(const other_chartype *buf, optype op)`
- `OSCL_IMPORT_REF void set(const other_chartype *buf, uint32 length, optype op)`

Friends

- class `OSCL_wString`

7.86.1 Detailed Description

OSCL_wHeapStringA is identical to [OSCL_HeapStringA](#) except that it uses wide-character format. For descriptions, see [OSCL_HeapStringA](#).

7.86.2 Member Typedef Documentation

7.86.2.1 `typedef OSCL_wString::chartype OSCL_wHeapStringA::chartype`

Reimplemented from [OSCL_wString](#).

7.86.2.2 `typedef TOSCL_wStringOp OSCL_wHeapStringA::optype`

7.86.2.3 `typedef OSCL_String::chartype OSCL_wHeapStringA::other_chartype`

7.86.3 Constructor & Destructor Documentation

7.86.3.1 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA()`

7.86.3.2 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA(Oscl_DefAlloc *alloc, OsclRefCounter *ref = NULL)`

7.86.3.3 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA(const OSCL_wHeapStringA &src)`

7.86.3.4 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA(const OSCL_wHeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref = NULL)`

7.86.3.5 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA(const OSCL_wString &src, Oscl_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

7.86.3.6 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA(const chartype *cstr, Oscl_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

7.86.3.7 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA(const chartype *buf, uint32 length, Oscl_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

7.86.3.8 `OSCL_IMPORT_REF OSCL_wHeapStringA::~OSCL_wHeapStringA()`

7.86.4 Member Function Documentation

7.86.4.1 `OSCL_IMPORT_REF const chartype* OSCL_wHeapStringA::get_cstr() [virtual]`

Implements [OSCL_wString](#).

7.86.4.2 `OSCL_IMPORT_REF uint32 OSCL_wHeapStringA::get_maxsize() [virtual]`

Implements [OSCL_wString](#).

7.86.4.3 OSCL_IMPORT_REF uint32 OSCL_wHeapStringA::get_size () [virtual]

Implements [OSCL_wString](#).

7.86.4.4 OSCL_IMPORT_REF chartype* OSCL_wHeapStringA::get_str () [virtual]

Implements [OSCL_wString](#).

7.86.4.5 OSCL_IMPORT_REF OSCL_wHeapStringA& OSCL_wHeapStringA::operator= (const chartype * cstr)

Reimplemented from [OSCL_wString](#).

7.86.4.6 OSCL_IMPORT_REF OSCL_wHeapStringA& OSCL_wHeapStringA::operator= (const OSCL_wString & src)

Reimplemented from [OSCL_wString](#).

7.86.4.7 OSCL_IMPORT_REF OSCL_wHeapStringA& OSCL_wHeapStringA::operator= (const OSCL_wHeapStringA & src)**7.86.4.8 OSCL_IMPORT_REF void OSCL_wHeapStringA::set (const other_chartype * buf, uint32 length, optype op)****7.86.4.9 OSCL_IMPORT_REF void OSCL_wHeapStringA::set (const other_chartype * buf, optype op)****7.86.4.10 OSCL_IMPORT_REF void OSCL_wHeapStringA::set (const chartype * buf, uint32 length)**

7.86.5 Friends And Related Function Documentation

7.86.5.1 friend class OSCL_wString [friend]

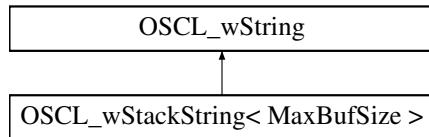
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.87 OSCL_wStackString< MaxBufSize > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wStackString< MaxBufSize >::



Public Types

- `typedef OSCL_wString::chartype chartype`
- `typedef TOSCL_wStringOp optype`
- `typedef OSCL_String::chartype other_chartype`

Public Methods

- `OSCL_wStackString()`
- `OSCL_wStackString(const OSCL_wStackString &src)`
- `OSCL_wStackString(const OSCL_wString &src)`
- `OSCL_wStackString(const chartype *cstr)`
- `OSCL_wStackString(const chartype *buf, uint32 length)`
- `~OSCL_wStackString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype *get_cstr() const`
- `chartype *get_str() const`
- `OSCL_wStackString & operator=(const OSCL_wStackString &src)`
- `OSCL_wStackString & operator=(const OSCL_wString &src)`
- `OSCL_wStackString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`
- `void set(const other_chartype *buf, optype op)`
- `void set(const other_chartype *buf, uint32 length, optype op)`

Friends

- class `OSCL_wString`

7.87.1 Detailed Description

```
template<uint32 MaxBufSize> class OSCL_wStackString< MaxBufSize >
```

`OSCL_wStackString` is identical to `OSCL_StackString` except that it uses wide-character format. For descriptions, see `OSCL_StackString`.

7.87.2 Member Typedef Documentation

7.87.2.1 template<uint32 MaxBufSize> typedef OSCL_wString::chartype OSCL_wStackString< MaxBufSize >::chartype

Reimplemented from [OSCL_wString](#).

7.87.2.2 template<uint32 MaxBufSize> typedef TOSCL_wStringOp OSCL_wStackString< MaxBufSize >::optype

7.87.2.3 template<uint32 MaxBufSize> typedef OSCL_String::chartype OSCL_wStackString< MaxBufSize >::other_chartype

7.87.3 Friends And Related Function Documentation

7.87.3.1 template<uint32 MaxBufSize> friend class OSCL_wString [friend]

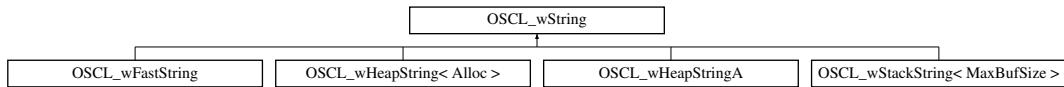
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

7.88 OSCL_wString Class Reference

```
#include <oscl_string.h>
```

Inheritance diagram for OSCL_wString::



Public Types

- `typedef oscl_wchar chartype`

Public Methods

- `virtual uint32 get_size () const=0`
- `virtual uint32 get_maxsize () const=0`
- `virtual const chartype * get_cstr () const=0`
- `virtual OSCL_IMPORT_REF bool is_writable () const`
- `virtual chartype * get_str () const=0`
- `OSCL_IMPORT_REF OSCL_wString & operator= (const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wString & operator= (const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_wString & operator+= (const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wString & operator+= (const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_wString & operator+= (const chartype c)`
- `OSCL_IMPORT_REF bool operator== (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator!= (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator< (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator<= (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator> (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator>= (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator== (const chartype *cstr) const`
- `OSCL_IMPORT_REF chartype operator[] (uint32 index) const`
- `virtual OSCL_IMPORT_REF chartype read (uint32 index) const`
- `virtual OSCL_IMPORT_REF uint32 setrep_to_wide_char (const char *src, uint32 len, TOSCL_wStringOp op, Oscl_DefAlloc *aAlloc)`
- `virtual OSCL_IMPORT_REF int8 hash () const`
- `virtual OSCL_IMPORT_REF void write (uint32 index, chartype c)`
- `virtual OSCL_IMPORT_REF void write (uint32 offset, uint32 length, const chartype *buf)`

Protected Methods

- `OSCL_IMPORT_REF OSCL_wString ()`
- `virtual OSCL_IMPORT_REF ~OSCL_wString ()`
- `virtual void set_rep (const chartype *cstr)=0`
- `virtual void append_rep (const chartype *cstr)=0`
- `virtual void set_rep (const OSCL_wString &src)=0`
- `virtual void append_rep (const OSCL_wString &src)=0`
- `virtual void set_len (uint32 len)=0`

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_IMPORT_REF OSCL_wString::OSCL_wString () [protected]`

7.88.3.2 `virtual OSCL_IMPORT_REF 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 OSCL_IMPORT_REF int8 OSCL_wString::hash () [virtual]**
- 7.88.4.8 **virtual OSCL_IMPORT_REF bool OSCL_wString::is_writable () [virtual]**
- 7.88.4.9 **OSCL_IMPORT_REF bool OSCL_wString::operator!= (const OSCL_wString & src) const**
- 7.88.4.10 **OSCL_IMPORT_REF OSCL_wString& OSCL_wString::operator+= (const chartype c)**
- 7.88.4.11 **OSCL_IMPORT_REF OSCL_wString& OSCL_wString::operator+= (const chartype * cstr)**
- 7.88.4.12 **OSCL_IMPORT_REF OSCL_wString& OSCL_wString::operator+= (const OSCL_wString & src)**
- 7.88.4.13 **OSCL_IMPORT_REF bool OSCL_wString::operator< (const OSCL_wString & src) const**
- 7.88.4.14 **OSCL_IMPORT_REF bool OSCL_wString::operator<= (const OSCL_wString & src) const**
- 7.88.4.15 **OSCL_IMPORT_REF OSCL_wString& OSCL_wString::operator= (const chartype * cstr)**

Reimplemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

- 7.88.4.16 **OSCL_IMPORT_REF OSCL_wString& OSCL_wString::operator= (const OSCL_wString & src)**

Reimplemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), and [OSCL_wStackString< MaxBufSize >](#).

- 7.88.4.17 **OSCL_IMPORT_REF bool OSCL_wString::operator== (const chartype * cstr) const**

- 7.88.4.18 **OSCL_IMPORT_REF bool OSCL_wString::operator== (const OSCL_wString & src) const**

- 7.88.4.19 **OSCL_IMPORT_REF bool OSCL_wString::operator> (const OSCL_wString & src) const**

- 7.88.4.20 **OSCL_IMPORT_REF bool OSCL_wString::operator>= (const OSCL_wString & src) const**

- 7.88.4.21]

OSCL_IMPORT_REF chartype OSCL_wString::operator[] (uint32 index) const

- 7.88.4.22 **virtual OSCL_IMPORT_REF chartype OSCL_wString::read (uint32 *index*) const [virtual]**
- 7.88.4.23 **virtual void OSCL_wString::set_len (uint32 *len*) [protected, pure virtual]**
- 7.88.4.24 **virtual void OSCL_wString::set_rep (const OSCL_wString & *src*) [protected, pure virtual]**
- 7.88.4.25 **virtual void OSCL_wString::set_rep (const chartype * *cstr*) [protected, pure virtual]**
- 7.88.4.26 **virtual OSCL_IMPORT_REF uint32 OSCL_wString::setrep_to_wide_char (const char * *src*, uint32 *len*, TOSCL_wStringOp *op*, Oscl_DefAlloc * *aAlloc*) [virtual]**
- 7.88.4.27 **virtual OSCL_IMPORT_REF void OSCL_wString::write (uint32 *offset*, uint32 *length*, const chartype * *buf*) [virtual]**
- 7.88.4.28 **virtual OSCL_IMPORT_REF void OSCL_wString::write (uint32 *index*, chartype *c*) [virtual]**

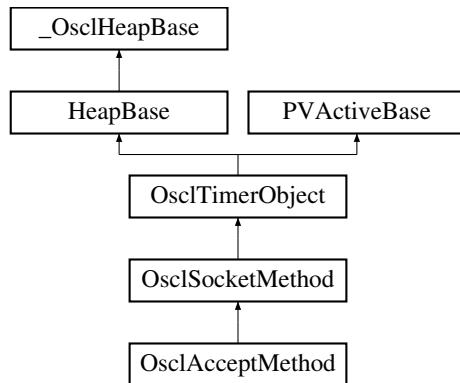
The documentation for this class was generated from the following file:

- [oscl_string.h](#)

7.89 OsclAcceptMethod Class Reference

```
#include <oscl_socket_accept.h>
```

Inheritance diagram for OsclAcceptMethod::



Public Methods

- [~OsclAcceptMethod \(\)](#)
- [TPVSocketEvent Accept \(int32 aTimeout\)](#)
- [void DiscardAcceptedSocket \(\)](#)
- [OsclSocketI * GetAcceptedSocket \(\)](#)
- [OsclAcceptRequest * AcceptRequest \(\)](#)

Static Public Methods

- [OsclAcceptMethod * NewL \(OsclIPSocketI &c\)](#)

7.89.1 Constructor & Destructor Documentation

7.89.1.1 OsclAcceptMethod::~OsclAcceptMethod ()

7.89.2 Member Function Documentation

7.89.2.1 TPVSocketEvent OsclAcceptMethod::Accept (int32 aTimeout)

7.89.2.2 OsclAcceptRequest* OsclAcceptMethod::AcceptRequest () [inline]

7.89.2.3 void OsclAcceptMethod::DiscardAcceptedSocket ()

7.89.2.4 OsclSocketI* OsclAcceptMethod::GetAcceptedSocket ()

7.89.2.5 OsclAcceptMethod* OsclAcceptMethod::NewL (OsclIPSocketI &c) [static]

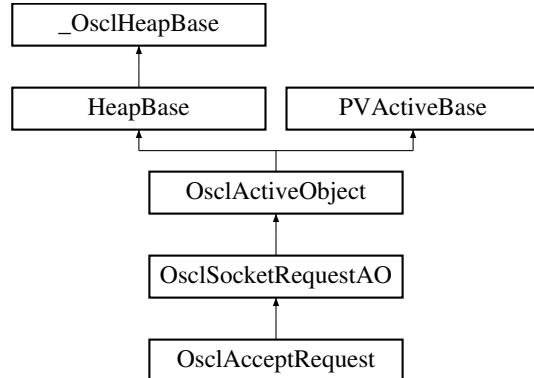
The documentation for this class was generated from the following file:

- [oscl_socket_accept.h](#)

7.90 OsclAcceptRequest Class Reference

```
#include <oscl_socket_accept.h>
```

Inheritance diagram for OsclAcceptRequest::



Public Methods

- [OsclAcceptRequest \(OsclSocketMethod &c\)](#)
- [void Accept \(OsclSocketI &aSocket\)](#)

7.90.1 Constructor & Destructor Documentation

7.90.1.1 OsclAcceptRequest::OsclAcceptRequest ([OsclSocketMethod & c](#)) [inline]

7.90.2 Member Function Documentation

7.90.2.1 void OsclAcceptRequest::Accept ([OsclSocketI & aSocket](#))

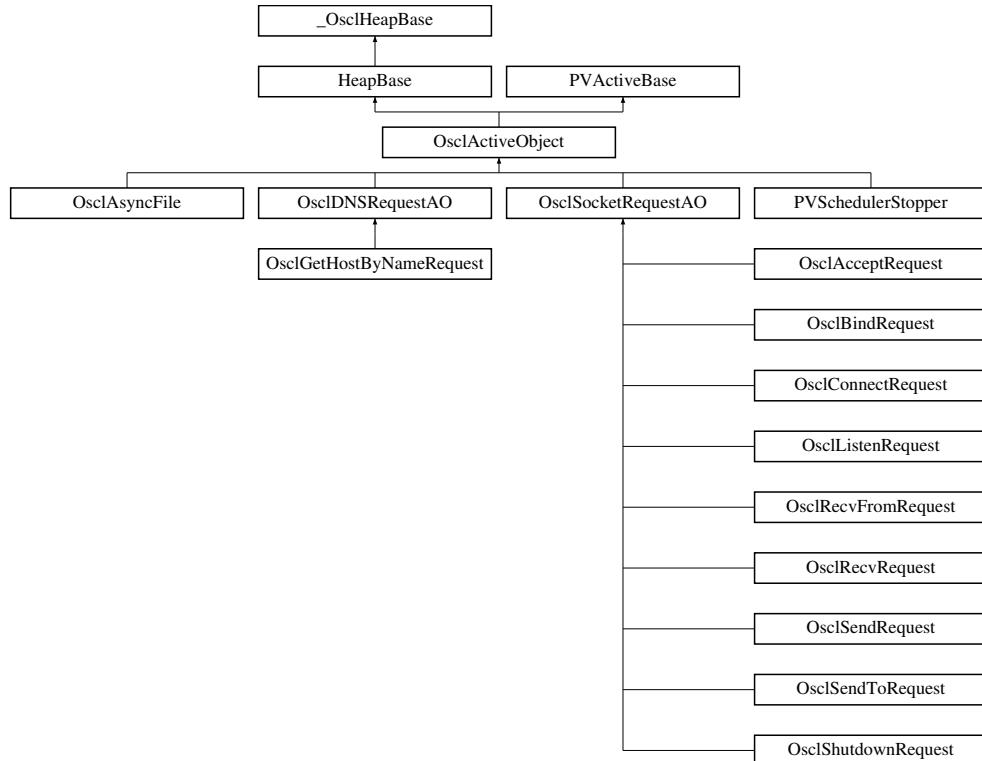
The documentation for this class was generated from the following file:

- [oscl_socket_accept.h](#)

7.91 OsclActiveObject Class Reference

```
#include <oscl_scheduler_ao.h>
```

Inheritance diagram for OsclActiveObject::



Public Types

- enum **OsclActivePriority** { **EPriorityIdle** = -100, **EPriorityLow** = -20, **EPriorityNominal** = 0, **EPriorityHigh** = 10, **EPriorityHighest** = 20 }

Public Methods

- OSCL_IMPORT_REF **OsclActiveObject** (int32 aPriority, const char name[])
- virtual OSCL_IMPORT_REF ~**OsclActiveObject** ()
- OSCL_IMPORT_REF void **SetBusy** ()
- OSCL_IMPORT_REF bool **IsBusy** () const
- OSCL_IMPORT_REF void **PendForExec** ()
- OSCL_IMPORT_REF void **PendComplete** (int32 aStatus)
- OSCL_IMPORT_REF void **AddToScheduler** ()
- OSCL_IMPORT_REF void **RemoveFromScheduler** ()
- OSCL_IMPORT_REF void **RunIfNotReady** ()
- OSCL_IMPORT_REF void **Cancel** ()
- OSCL_IMPORT_REF int32 **Priority** () const
- OSCL_IMPORT_REF int32 **Status** () const
- OSCL_IMPORT_REF void **SetStatus** (int32)
- OSCL_IMPORT_REF **OsclAOStatus & StatusRef** ()

Protected Methods

- virtual OSCL_IMPORT_REF void [DoCancel \(\)](#)
- virtual OSCL_IMPORT_REF int32 [RunError \(int32 aError\)](#)

7.91.1 Detailed Description

User base class for execution objects. OsclActiveObject defines an execution object without any timer. This AO can be used across threads, i.e. the request can be activated in one thread and completed in another.

7.91.2 Member Enumeration Documentation

7.91.2.1 enum OsclActiveObject::OsclActivePriority

Scheduling priorities.

Enumeration values:

- EPriorityIdle** A low priority, useful for execution objects representing background processing.
EPriorityLow A priority higher than EPriorityIdle but lower than EPriorityNominal.
EPriorityNominal Most exec objects will have this priority.
EPriorityHigh A priority higher than EPriorityNominal; useful for execution objects handling user input.
EPriorityHighest A priority higher than EPriorityHighest.

7.91.3 Constructor & Destructor Documentation

7.91.3.1 OSCL_IMPORT_REF OsclActiveObject::OsclActiveObject (int32 *aPriority*, const char *name*[])

Constructor.

Parameters:

- aPriority* (input param): scheduling priority
name (input param): optional name for this AO.

7.91.3.2 virtual OSCL_IMPORT_REF OsclActiveObject::~OsclActiveObject () [virtual]

Destructor.

7.91.4 Member Function Documentation

7.91.4.1 OSCL_IMPORT_REF void OsclActiveObject::AddToScheduler ()

Add this exec object to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

7.91.4.2 OSCL_IMPORT_REF void OsclActiveObject::Cancel ()

Cancel any pending request. If the request is readied, this will call the DoCancel routine, wait for the request to cancel, then set the request idle. The AO will not run. If the request is not readied, it does nothing. Request must be canceled from the same thread in which it is scheduled.

Reimplemented from [PVActiveBase](#).

7.91.4.3 virtual OSCL_IMPORT_REF void OsclActiveObject::DoCancel () [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Implements [PVActiveBase](#).

Reimplemented in [OsclDNSRequestAO](#), and [OsclSocketRequestAO](#).

7.91.4.4 OSCL_IMPORT_REF bool OsclActiveObject::IsBusy ()

Return true if this AO is pending, false otherwise.

7.91.4.5 OSCL_IMPORT_REF void OsclActiveObject::PendComplete (int32 *aStatus*)

Complete the active request for the AO. This API is thread-safe. If the request is not pending, this call will leave.

Parameters:

aStatus: request completion status.

7.91.4.6 OSCL_IMPORT_REF void OsclActiveObject::PendForExec ()

Set request active for this AO and set the status to pending. PendForExec is identical to SetActive, but it additionally sets the request status to OSCL_REQUEST_PENDING.

7.91.4.7 OSCL_IMPORT_REF int32 OsclActiveObject::Priority ()

Return scheduling priority of this exec object.

7.91.4.8 OSCL_IMPORT_REF void OsclActiveObject::RemoveFromScheduler ()

Remove this AO from its scheduler. Will leave if the calling thread context does not match the scheduling thread. Cancels any readied request before removing.

Reimplemented from [PVActiveBase](#).

**7.91.4.9 virtual OSCL_IMPORT_REF int32 OsclActiveObject::RunError (int32 *aError*)
[protected, virtual]**

Run Error handler. This gets called by scheduler when the Run routine leaves. The default implementation simply returns the leave code. If the derived class wants to handle errors from Run, it may override this. The RunError should return OsclErrNone if it handles the error, otherwise it should return the input error code.

Parameters:

aError: the leave code generated by the Run.

Implements [PVActiveBase](#).

7.91.4.10 OSCL_IMPORT_REF void OsclActiveObject::RunIfNotReady ()

Complete this AO's request immediately. If the AO is already active, this will do nothing. Will leave if the AO is not added to any scheduler, or if the calling thread context does not match the scheduling thread.

7.91.4.11 OSCL_IMPORT_REF void OsclActiveObject::SetBusy ()

Set object ready for this AO, additionally sets the request status to OSCL_REQUEST_PENDING. Will leave if the request is already readied, or the execution object is not added to any scheduler, or the calling thread context does not match the scheduler thread.

7.91.4.12 OSCL_IMPORT_REF void OsclActiveObject::SetStatus (int32)**7.91.4.13 OSCL_IMPORT_REF int32 OsclActiveObject::Status ()**

Request status access

7.91.4.14 OSCL_IMPORT_REF OsclAOStatus& OsclActiveObject::StatusRef ()

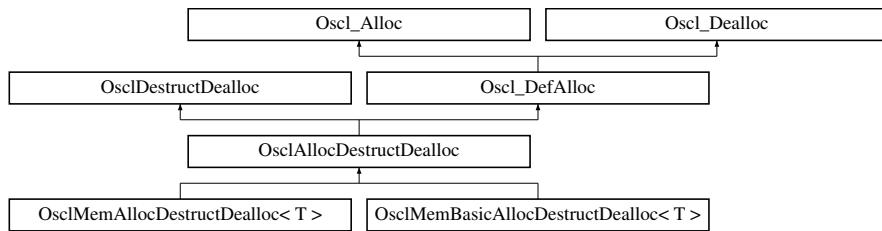
The documentation for this class was generated from the following file:

- [oscl_scheduler_ao.h](#)

7.92 OsclAllocDestructDealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for OsclAllocDestructDealloc::



Public Methods

- virtual ~OsclAllocDestructDealloc ()

7.92.1 Constructor & Destructor Documentation

7.92.1.1 virtual OsclAllocDestructDealloc::~OsclAllocDestructDealloc () [inline, virtual]

The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

7.93 OsclAOStatus Class Reference

```
#include <oscl_aostatus.h>
```

Public Methods

- OSCL_INLINE OsclAOStatus ()
- OSCL_INLINE OsclAOStatus (int32 aStatus)
- OSCL_INLINE int32 operator= (int32 aStatus)
- OSCL_INLINE int32 operator== (int32 aStatus) const
- OSCL_INLINE int32 operator!= (int32 aStatus) const
- OSCL_INLINE int32 operator>= (int32 aStatus) const
- OSCL_INLINE int32 operator<= (int32 aStatus) const
- OSCL_INLINE int32 operator> (int32 aStatus) const
- OSCL_INLINE int32 operator< (int32 aStatus) const
- OSCL_INLINE int32 Value () const

7.93.1 Constructor & Destructor Documentation

7.93.1.1 OSCL_INLINE OsclAOStatus::OsclAOStatus ()

7.93.1.2 OSCL_INLINE OsclAOStatus::OsclAOStatus (int32 *aStatus*)

7.93.2 Member Function Documentation

7.93.2.1 OSCL_INLINE int32 OsclAOStatus::operator!= (int32 *aStatus*) const

7.93.2.2 OSCL_INLINE int32 OsclAOStatus::operator< (int32 *aStatus*) const

7.93.2.3 OSCL_INLINE int32 OsclAOStatus::operator<= (int32 *aStatus*) const

7.93.2.4 OSCL_INLINE int32 OsclAOStatus::operator= (int32 *aStatus*)

7.93.2.5 OSCL_INLINE int32 OsclAOStatus::operator== (int32 *aStatus*) const

7.93.2.6 OSCL_INLINE int32 OsclAOStatus::operator> (int32 *aStatus*) const

7.93.2.7 OSCL_INLINE int32 OsclAOStatus::operator>= (int32 *aStatus*) const

7.93.2.8 OSCL_INLINE int32 OsclAOStatus::Value ()

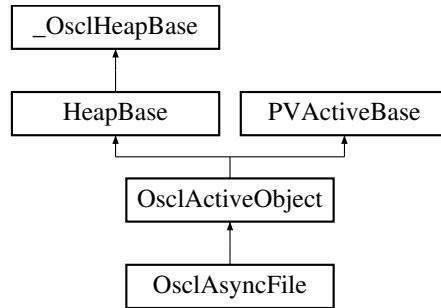
The documentation for this class was generated from the following file:

- [oscl_aostatus.h](#)

7.94 OsclAsyncFile Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclAsyncFile::



Public Methods

- [`~OsclAsyncFile \(\)`](#)
- [`int32 Open \(const oscl_wchar *filename, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv\)`](#)
- [`int32 Open \(const char *filename, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv\)`](#)
- [`int32 Seek \(TOsclFileOffset offset, Oscl_File::seek_type origin\)`](#)
- [`TOsclFileOffset Tell \(\)`](#)
- [`uint32 Read \(OsclAny *aBuffer1, uint32 aDataSize, uint32 aNumElements\)`](#)
- [`int32 EndOfFile \(\)`](#)
- [`TOsclFileOffset Size \(\)`](#)
- [`int32 Close \(\)`](#)
- [`uint32 Write \(const OsclAny *aBuffer1, uint32 aDataSize, uint32 aNumElements\)`](#)
- [`uint32 Flush \(\)`](#)

Static Public Methods

- [`OsclAsyncFile * NewL \(OsclNativeFile &aAsyncFile, int32 aCacheSize, PVLogger *\)`](#)
- [`void Delete \(OsclAsyncFile *\)`](#)

Data Fields

- [`uint32 iNumOfRun`](#)
- [`uint32 iNumOfRunErr`](#)

7.94.1 Detailed Description

OsclAsyncFile

7.94.2 Constructor & Destructor Documentation

7.94.2.1 OsclAsyncFile::~OsclAsyncFile ()

Destructor.

7.94.3 Member Function Documentation

7.94.3.1 int32 OsclAsyncFile::Close ()

7.94.3.2 void OsclAsyncFile::Delete (OsclAsyncFile *) [static]

7.94.3.3 int32 OsclAsyncFile::EndOfFile ()

7.94.3.4 uint32 OsclAsyncFile::Flush () [inline]

7.94.3.5 OsclAsyncFile* OsclAsyncFile::NewL (OsclNativeFile & aSyncFile, int32 aCacheSize, PVLogger *) [static]

Two-phased constructor.

Parameters:

aSyncFile: open handle for async file read. Note: it is the caller's job to open/close this file handle.

aSyncFile: duplicate open handle for sync file read. Note: it is the caller's job to open this file handle, but this class will close the handle.

aCacheSize: size of one of the individual cache buffers. The total cached data size will be larger, since multiple buffers are used.

aStartAsyncRead: When true, async file read will start immediately. When false, read will not begin until StartAsyncRead is called.

- 7.94.3.6 int32 OsclAsyncFile::Open (const char **filename*, uint32 *mode*, const OsclNativeFileParams & *params*, Oscl_FileServer & *fileserv*)
- 7.94.3.7 int32 OsclAsyncFile::Open (const oscl_wchar **filename*, uint32 *mode*, const OsclNativeFileParams & *params*, Oscl_FileServer & *fileserv*)
- 7.94.3.8 uint32 OsclAsyncFile::Read (OsclAny **aBuffer1*, uint32 *aDataSize*, uint32 *aNumElements*)
- 7.94.3.9 int32 OsclAsyncFile::Seek (TOsclFileOffset *offset*, Oscl_File::seek_type *origin*)
- 7.94.3.10 TOsclFileOffset OsclAsyncFile::Size ()
- 7.94.3.11 TOsclFileOffset OsclAsyncFile::Tell ()
- 7.94.3.12 uint32 OsclAsyncFile::Write (const OsclAny **aBuffer1*, uint32 *aDataSize*, uint32 *aNumElements*) [inline]

7.94.4 Field Documentation

- 7.94.4.1 uint32 OsclAsyncFile::iNumOfRun

- 7.94.4.2 uint32 OsclAsyncFile::iNumOfRunErr

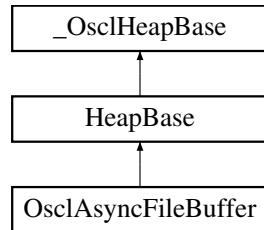
The documentation for this class was generated from the following file:

- [oscl_file_async_read.h](#)

7.95 OsclAsyncFileBuffer Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclAsyncFileBuffer::



Public Methods

- `~OsclAsyncFileBuffer ()`
- `void CleanInUse ()`
- `void SetInUse ()`
- `bool IsInUse ()`
- `bool IsValid ()`
- `TOsclFileOffset Offset ()`
- `void SetOffset (TOsclFileOffset aOffset)`
- `int32 Length ()`
- `bool HasThisOffset (TOsclFileOffset aOffset)`
- `int32 Id ()`
- `OsclBuf * Buffer ()`
- `void UpdateData ()`
- `void StartAsyncRead (bool aStartAsyncRead)`

Static Public Methods

- `OsclAsyncFileBuffer * NewL (int32 aBufferSize, int32 aId)`

7.95.1 Detailed Description

Buffer class used with async read. We keep an array of these, covering consecutive areas of the file. This allows for some seeking without requiring a full flush & refill each time.

7.95.2 Constructor & Destructor Documentation

7.95.2.1 `OsclAsyncFileBuffer::~OsclAsyncFileBuffer ()`

7.95.3 Member Function Documentation

7.95.3.1 `OsclBuf* OsclAsyncFileBuffer::Buffer ()`

7.95.3.2 `void OsclAsyncFileBuffer::CleanInUse () [inline]`

7.95.3.3 `bool OsclAsyncFileBuffer::HasThisOffset (TOsclFileOffset aOffset)`

7.95.3.4 `int32 OsclAsyncFileBuffer::Id () [inline]`

7.95.3.5 `bool OsclAsyncFileBuffer::IsInUse () [inline]`

7.95.3.6 `bool OsclAsyncFileBuffer::IsValid () [inline]`

7.95.3.7 `int32 OsclAsyncFileBuffer::Length () [inline]`

7.95.3.8 `OsclAsyncFileBuffer* OsclAsyncFileBuffer::NewL (int32 aBufferSize, int32 aId) [static]`

7.95.3.9 `TOsclFileOffset OsclAsyncFileBuffer::Offset () [inline]`

7.95.3.10 `void OsclAsyncFileBuffer::SetInUse () [inline]`

7.95.3.11 `void OsclAsyncFileBuffer::SetOffset (TOsclFileOffset aOffset) [inline]`

7.95.3.12 `void OsclAsyncFileBuffer::StartAsyncRead (bool aStartAsyncRead)`

7.95.3.13 `void OsclAsyncFileBuffer::UpdateData ()`

The documentation for this class was generated from the following file:

- `oscl_file_async_read.h`

7.96 OsclAuditCB Class Reference

```
#include <oscl_mem.h>
```

Public Methods

- [OsclAuditCB \(\)](#)
- [OsclAuditCB \(const OsclMemStatsNode *myStatsNode, OsclMemAudit *ptr\)](#)

Data Fields

- [const OsclMemStatsNode * pStatsNode](#)
- [OsclMemAudit * pAudit](#)

7.96.1 Constructor & Destructor Documentation

7.96.1.1 OsclAuditCB::OsclAuditCB () [inline]

7.96.1.2 OsclAuditCB::OsclAuditCB (const OsclMemStatsNode * *myStatsNode*, OsclMemAudit * *ptr*) [inline]

7.96.2 Field Documentation

7.96.2.1 OsclMemAudit* OsclAuditCB::pAudit

7.96.2.2 const OsclMemStatsNode* OsclAuditCB::pStatsNode

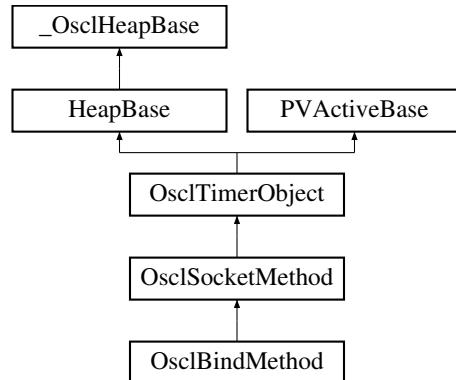
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

7.97 OsclBindMethod Class Reference

```
#include <oscl_socket_bind.h>
```

Inheritance diagram for OsclBindMethod::



Public Methods

- [~OsclBindMethod \(\)](#)
- [TPVSocketEvent Bind \(OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsclBindRequest * BindRequest \(\)](#)

Static Public Methods

- [OsclBindMethod * NewL \(OsclIPSocketI &c\)](#)

7.97.1 Constructor & Destructor Documentation

7.97.1.1 OsclBindMethod::~OsclBindMethod ()

7.97.2 Member Function Documentation

7.97.2.1 TPVSocketEvent OsclBindMethod::Bind (OsclNetworkAddress & aAddress, int32 aTimeout)

7.97.2.2 OsclBindRequest* OsclBindMethod::BindRequest () [inline]

7.97.2.3 OsclBindMethod* OsclBindMethod::NewL (OsclIPSocketI &c) [static]

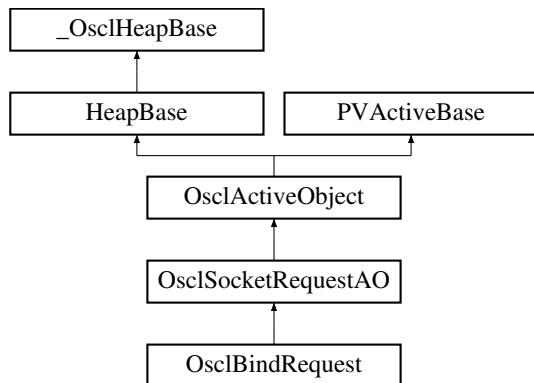
The documentation for this class was generated from the following file:

- [oscl_socket_bind.h](#)

7.98 OsclBindRequest Class Reference

```
#include <oscl_socket_bind.h>
```

Inheritance diagram for OsclBindRequest::



Public Methods

- [OsclBindRequest \(OsclSocketMethod &c\)](#)
- [void Bind \(OsclNetworkAddress &aAddress\)](#)

7.98.1 Detailed Description

This is the AO that interacts with the socket server

7.98.2 Constructor & Destructor Documentation

7.98.2.1 OsclBindRequest::OsclBindRequest ([OsclSocketMethod & c](#)) [inline]

7.98.3 Member Function Documentation

7.98.3.1 void OsclBindRequest::Bind ([OsclNetworkAddress & aAddress](#))

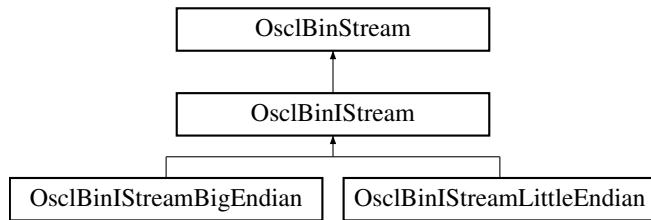
The documentation for this class was generated from the following file:

- [oscl_socket_bind.h](#)

7.99 OsclBinIStream Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStream::



Public Methods

- [OsclBinIStream \(\)](#)
- [~OsclBinIStream \(\)](#)
- uint8 [Read_uint8 \(\)](#)

This method reads an unsigned short from the stream.

- OsclBinIStream & [get \(int8 *data, int32 size\)](#)

This method reads 'length' number of bytes from the stream and places them in 'data'.

7.99.1 Constructor & Destructor Documentation

7.99.1.1 OsclBinIStream::OsclBinIStream () [inline]

7.99.1.2 OsclBinIStream::~OsclBinIStream () [inline]

7.99.2 Member Function Documentation

7.99.2.1 OsclBinIStream& OsclBinIStream::get (int8 * data, int32 size)

This method reads 'length' number of bytes from the stream and places them in 'data'.

Parameters:

data is a pointer to the place to store the bytes read

size is the number of bytes to read

7.99.2.2 uint8 OsclBinIStream::Read_uint8 ()

This method reads an unsigned short from the stream.

Returns:

Unsigned short read from the stream.

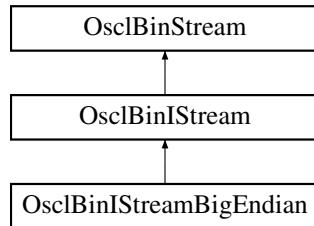
The documentation for this class was generated from the following file:

- [oscl_bin_stream.h](#)

7.100 OsclBinIStreamBigEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStreamBigEndian::



Public Methods

- [OsclBinIStreamBigEndian \(\)](#)
- [void Read \(int8 &data\)](#)
- [void Read \(uint8 &data\)](#)
- [void Read \(int16 &data\)](#)
- [void Read \(uint16 &data\)](#)
- [void Read \(int32 &data\)](#)
- [void Read \(uint32 &data\)](#)
- [OsclBinIStreamBigEndian & operator>> \(int8 &data\)](#)

This method reads a int8 from the stream and stores it in 'data'.

- [OsclBinIStream & operator>> \(uint8 &data\)](#)

This method reads a uint8 from the stream and stores it in 'data'.

- [OsclBinIStreamBigEndian & operator>> \(int16 &data\)](#)

This method reads a int16 from the stream and stores it in 'data'.

- [OsclBinIStreamBigEndian & operator>> \(uint16 &data\)](#)

This method reads a uint16 from the stream and stores it in 'data'.

- [OsclBinIStreamBigEndian & operator>> \(int32 &data\)](#)

This method reads a int32 from the stream and stores it in 'data'.

- [OsclBinIStreamBigEndian & operator>> \(uint32 &data\)](#)

This method reads a uint32 from the stream and stores it in 'data'.

- [uint16 Read_uint16 \(\)](#)

This method reads an unsigned short from the stream.

- [uint32 Read_uint32 \(\)](#)

This method reads an unsigned long from the stream.

7.100.1 Constructor & Destructor Documentation

7.100.1.1 OsclBinIStreamBigEndian::OsclBinIStreamBigEndian () [inline]

7.100.2 Member Function Documentation

7.100.2.1 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (uint32 & data)

This method reads a uint32 from the stream and stores it in 'data'.

7.100.2.2 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int32 & data)

This method reads a int32 from the stream and stores it in 'data'.

7.100.2.3 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (uint16 & data)

This method reads a uint16 from the stream and stores it in 'data'.

7.100.2.4 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int16 & data)

This method reads a int16 from the stream and stores it in 'data'.

7.100.2.5 OsclBinIStream& OsclBinIStreamBigEndian::operator>> (uint8 & data)

This method reads a uint8 from the stream and stores it in 'data'.

7.100.2.6 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int8 & data)

This method reads a int8 from the stream and stores it in 'data'.

7.100.2.7 void OsclBinIStreamBigEndian::Read (uint32 & data)

7.100.2.8 void OsclBinIStreamBigEndian::Read (int32 & data)

7.100.2.9 void OsclBinIStreamBigEndian::Read (uint16 & data)

7.100.2.10 void OsclBinIStreamBigEndian::Read (int16 & data)

7.100.2.11 void OsclBinIStreamBigEndian::Read (uint8 & data)

7.100.2.12 void OsclBinIStreamBigEndian::Read (int8 & data)

7.100.2.13 uint16 OsclBinIStreamBigEndian::Read_uint16 ()

This method reads an unsigned short from the stream.

Returns:

Unsigned short read from the stream.

7.100.2.14 uint32 OsclBinIStreamBigEndian::Read_uint32 ()

This method reads an unsigned long from the stream.

Returns:

unsigned long read from the stream.

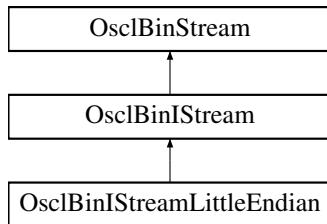
The documentation for this class was generated from the following file:

- [oscl_bin_stream.h](#)

7.101 OsclBinIStreamLittleEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStreamLittleEndian::



Public Methods

- [OsclBinIStreamLittleEndian \(\)](#)
[OsclBinIStreamLittleEndian & operator>> \(int8 &data\)](#)
This method reads a int8 from the stream and stores it in 'data'.
- [OsclBinIStreamLittleEndian & operator>> \(uint8 &data\)](#)
This method reads a uint8 from the stream and stores it in 'data'.
- [OsclBinIStreamLittleEndian & operator>> \(int16 &data\)](#)
This method reads a int16 from the stream and stores it in 'data'.
- [OsclBinIStreamLittleEndian & operator>> \(uint16 &data\)](#)
This method reads a uint16 from the stream and stores it in 'data'.
- [OsclBinIStreamLittleEndian & operator>> \(int32 &data\)](#)
This method reads a int32 from the stream and stores it in 'data'.
- [OsclBinIStreamLittleEndian & operator>> \(uint32 &data\)](#)
This method reads a uint32 from the stream and stores it in 'data'.

Protected Methods

- uint16 [Read_uint16 \(\)](#)
- uint32 [Read_uint32 \(\)](#)

7.101.1 Constructor & Destructor Documentation

7.101.1.1 OsclBinIStreamLittleEndian::OsclBinIStreamLittleEndian () [inline]

7.101.2 Member Function Documentation

7.101.2.1 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint32 & data)

This method reads a uint32 from the stream and stores it in 'data'.

7.101.2.2 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int32 & data)

This method reads a int32 from the stream and stores it in 'data'.

7.101.2.3 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint16 & data)

This method reads a uint16 from the stream and stores it in 'data'.

7.101.2.4 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int16 & data)

This method reads a int16 from the stream and stores it in 'data'.

7.101.2.5 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint8 & data)

This method reads a uint8 from the stream and stores it in 'data'.

7.101.2.6 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int8 & data)

This method reads a int8 from the stream and stores it in 'data'.

7.101.2.7 uint16 OsclBinIStreamLittleEndian::Read_uint16 () [protected]

7.101.2.8 uint32 OsclBinIStreamLittleEndian::Read_uint32 () [protected]

The documentation for this class was generated from the following file:

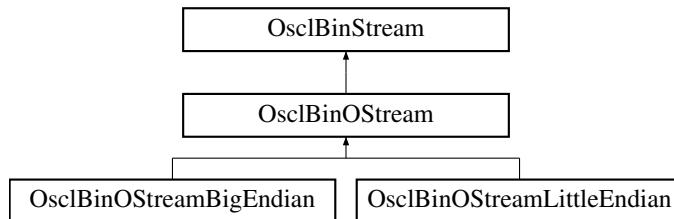
- [oscl_bin_stream.h](#)

7.102 OsclBinOStream Class Reference

Class OsclBinOStream implements the basic stream functions for an output stream.

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStream:::



Public Methods

- [OsclBinOStream \(\)](#)
- virtual [~OsclBinOStream \(\)](#)
- [OsclBinOStream & write \(const int8 *data, int32 size\)](#)

This method writes 'length' number of bytes stored in 'data' to the stream.

7.102.1 Detailed Description

Class OsclBinOStream implements the basic stream functions for an output stream.

7.102.2 Constructor & Destructor Documentation

7.102.2.1 [OsclBinOStream::OsclBinOStream \(\) \[inline\]](#)

7.102.2.2 [virtual OsclBinOStream::~OsclBinOStream \(\) \[inline, virtual\]](#)

7.102.3 Member Function Documentation

7.102.3.1 [OsclBinOStream& OsclBinOStream::write \(const int8 * data, int32 size\)](#)

This method writes 'length' number of bytes stored in 'data' to the stream.

The documentation for this class was generated from the following file:

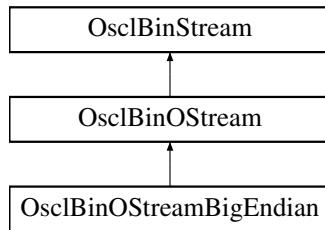
- [oscl_bin_stream.h](#)

7.103 OsclBinOStreamBigEndian Class Reference

Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering.

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStreamBigEndian::



Public Methods

- [OsclBinOStreamBigEndian \(\)](#)
- OsclBinOStreamBigEndian & [operator<< \(const int8 &data\)](#)
This method writes a int8 from 'data' to the stream.
- OsclBinOStreamBigEndian & [operator<< \(const uint8 &data\)](#)
This method writes a uint8 from 'data' to the stream.
- OsclBinOStreamBigEndian & [operator<< \(const int16 &data\)](#)
This method writes a int16 from 'data' to the stream.
- OsclBinOStreamBigEndian & [operator<< \(const uint16 &data\)](#)
This method writes a uint16 from 'data' to the stream.
- OsclBinOStreamBigEndian & [operator<< \(const int32 &data\)](#)
This method writes a int32 from 'data' to the stream.
- OsclBinOStreamBigEndian & [operator<< \(const uint32 &data\)](#)
This method writes a uint32 from 'data' to the stream.

Protected Methods

- void [WriteUnsignedShort \(const uint16 data\)](#)
- void [WriteUnsignedLong \(const uint32 data\)](#)

7.103.1 Detailed Description

Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering.

7.103.2 Constructor & Destructor Documentation

7.103.2.1 OsclBinOStreamBigEndian::OsclBinOStreamBigEndian () [inline]

7.103.3 Member Function Documentation

7.103.3.1 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint32 & data)

This method writes a uint32 from 'data' to the stream.

7.103.3.2 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int32 & data)

This method writes a int32 from 'data' to the stream.

7.103.3.3 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint16 & data)

This method writes a uint16 from 'data' to the stream.

7.103.3.4 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int16 & data)

This method writes a int16 from 'data' to the stream.

7.103.3.5 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint8 & data)

This method writes a uint8 from 'data' to the stream.

7.103.3.6 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int8 & data)

This method writes a int8 from 'data' to the stream.

7.103.3.7 void OsclBinOStreamBigEndian::WriteUnsignedLong (const uint32 data) [protected]

7.103.3.8 void OsclBinOStreamBigEndian::WriteUnsignedShort (const uint16 data) [protected]

The documentation for this class was generated from the following file:

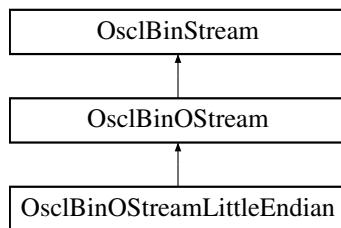
- [oscl_bin_stream.h](#)

7.104 OsclBinOStreamLittleEndian Class Reference

Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStreamLittleEndian::



Public Methods

- [OsclBinOStreamLittleEndian \(\)](#)
- [OsclBinOStreamLittleEndian & operator<< \(const int8 &data\)](#)
This method writes a int8 from 'data' to the stream.
- [OsclBinOStreamLittleEndian & operator<< \(const uint8 &data\)](#)
This method writes a uint8 from 'data' to the stream.
- [OsclBinOStreamLittleEndian & operator<< \(const int16 &data\)](#)
This method writes a int16 from 'data' to the stream.
- [OsclBinOStreamLittleEndian & operator<< \(const uint16 &data\)](#)
This method writes a uint16 from 'data' to the stream.
- [OsclBinOStreamLittleEndian & operator<< \(const int32 &data\)](#)
This method writes a int32 from 'data' to the stream.
- [OsclBinOStreamLittleEndian & operator<< \(const uint32 &data\)](#)
This method writes a uint32 from 'data' to the stream.

Protected Methods

- [void WriteUnsignedShort \(const uint16 data\)](#)
This method writes 'data' (unsigned short) to the stream.
- [void WriteUnsignedLong \(const uint32 data\)](#)
This method writes 'data' (unsigned long) to the stream.

7.104.1 Detailed Description

Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.

7.104.2 Constructor & Destructor Documentation

7.104.2.1 OsclBinOStreamLittleEndian::OsclBinOStreamLittleEndian () [inline]

7.104.3 Member Function Documentation

7.104.3.1 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint32 & data)

This method writes a uint32 from 'data' to the stream.

7.104.3.2 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int32 & data)

This method writes a int32 from 'data' to the stream.

7.104.3.3 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint16 & data)

This method writes a uint16 from 'data' to the stream.

7.104.3.4 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int16 & data)

This method writes a int16 from 'data' to the stream.

7.104.3.5 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint8 & data)

This method writes a uint8 from 'data' to the stream.

7.104.3.6 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int8 & data)

This method writes a int8 from 'data' to the stream.

7.104.3.7 void OsclBinOStreamLittleEndian::WriteUnsignedLong (const uint32 data) [protected]

This method writes 'data' (unsigned long) to the stream.

7.104.3.8 void OsclBinOStreamLittleEndian::WriteUnsignedShort (const uint16 data) [protected]

This method writes 'data' (unsigned short) to the stream.

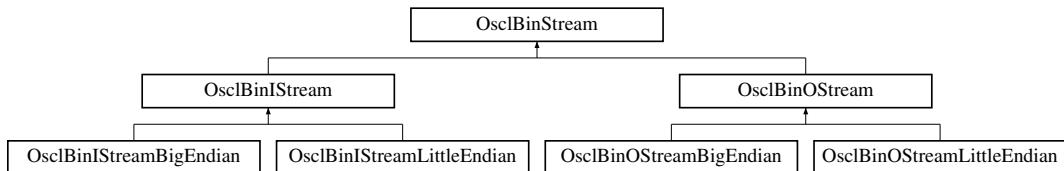
The documentation for this class was generated from the following file:

- [oscl_bin_stream.h](#)

7.105 OsclBinStream Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinStream::



Public Methods

- [OsclBinStream \(\)](#)
- [bool good \(\)](#)

This method determines if the stream is ok.

- [bool eof \(\)](#)

This method determines if end of stream has been reached.

- [bool fail \(\)](#)

This method determines if an error has occurred in the stream.

- [void Attach \(void *buffer, uint32 l_length\)](#)

This method specifies the data buffer to attach to the stream.

- [void Attach \(const uint32 numFragments, const OsclMemoryFragment *fragPtr\)](#)

This method specifies the memory fragment array to use for input.

- [uint32 tellg \(\)](#)

This method returns the current stream position.

- [void Seek \(uint32 absPosition\)](#)

This method seeks to the specified stream position.

- [uint32 PositionInBlock \(\)](#)

This method returns the current stream position.

- [void seekFromCurrentPosition \(int32 offset\)](#)

This method seeks to the specified offset from the current location.

Protected Types

- enum [state_t](#) { [GOOD_STATE](#), [EOF_STATE](#), [FAIL_STATE](#) }

Protected Methods

- bool [ReserveSpace](#) (uint32 size)
- bool [HaveRoomInCurrentBlock](#) (uint32 size)

Protected Attributes

- [state_t state](#)
- uint8 * [pBasePosition](#)
- uint8 * [pPosition](#)
- uint32 [length](#)
- const [OsclMemoryFragment](#) * [nextFragPtr](#)
- int [fragsLeft](#)
- const [OsclMemoryFragment](#) * [firstFragPtr](#)
- int [numFrags](#)
- [OsclMemoryFragment](#) [specialFragBuffer](#)

7.105.1 Member Enumeration Documentation

7.105.1.1 enum OsclBinStream::state_t [protected]

Enumeration values:

- GOOD_STATE**
- EOF_STATE**
- FAIL_STATE**

7.105.2 Constructor & Destructor Documentation

7.105.2.1 OsclBinStream::OsclBinStream () [inline]

7.105.3 Member Function Documentation

7.105.3.1 void OsclBinStream::Attach (const uint32 *numFragments*, const [OsclMemoryFragment](#) **fragPtr*)

This method specifies the memory fragment array to use for input.

This array should remain static while the stream refers to it.

Parameters:

numFragments is the number of elements in the array

fragPtr is the pointer to the MemoryFragment array

7.105.3.2 void OsclBinStream::Attach (void **buffer*, uint32 *l_length*)

This methods specifies the data buffer to attach to the stream.

Parameters:

buffer will provide the input

length of the buffer

7.105.3.3 bool OsclBinStream::eof ()

This method determines if end of stream has been reached.

Returns:

true if end of stream has been reached.

7.105.3.4 bool OsclBinStream::fail ()

This method determines if an error has occurred in the stream.

Returns:

true if an error occurred in the stream.

7.105.3.5 bool OsclBinStream::good ()

This method determines if the stream is ok.

Returns:

true if stream is ok.

7.105.3.6 bool OsclBinStream::HaveRoomInCurrentBlock (uint32 *size*) [protected]**7.105.3.7 uint32 OsclBinStream::PositionInBlock ()**

This method returns the current stream position.

Returns:

stream position.

7.105.3.8 bool OsclBinStream::ReserveSpace (uint32 *size*) [protected]**7.105.3.9 void OsclBinStream::Seek (uint32 *absPosition*)**

This method seeks to the specified stream position.

Returns:

Stream position.

7.105.3.10 void OsclBinStream::seekFromCurrentPosition (int32 *offset*)

This method seeks to the specified offset from the current location.

Parameters:

offset from current stream location

7.105.3.11 uint32 OsclBinStream::tellg ()

This method returns the current stream position.

This method is to be used if the input stream is a pointer to the MemoryFragment array

Returns:

Stream position.

7.105.4 Field Documentation

7.105.4.1 const OsclMemoryFragment* OsclBinStream::firstFragPtr [protected]

7.105.4.2 int OsclBinStream::fragsLeft [protected]

7.105.4.3 uint32 OsclBinStream::length [protected]

7.105.4.4 const OsclMemoryFragment* OsclBinStream::nextFragPtr [protected]

7.105.4.5 int OsclBinStream::numFrags [protected]

7.105.4.6 uint8* OsclBinStream::pBasePosition [protected]

7.105.4.7 uint8* OsclBinStream::pPosition [protected]

7.105.4.8 OsclMemoryFragment OsclBinStream::specialFragBuffer [protected]

7.105.4.9 state_t OsclBinStream::state [protected]

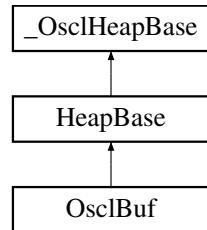
The documentation for this class was generated from the following file:

- [oscl_bin_stream.h](#)

7.106 OsclBuf Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclBuf::



Public Methods

- [OsclBuf](#) (int32 size)
- int32 [Length](#) ()
- [OsclPtr Des](#) ()
- [OsclPtrC DesC](#) ()

Static Public Methods

- OsclBuf * [NewL](#) (int32 size)
- void [Delete](#) (OsclBuf *a)

Data Fields

- uint8 * [iBuffer](#)
- int32 [iMaxLength](#)
- int32 [iLength](#)

7.106.1 Constructor & Destructor Documentation

7.106.1.1 `OsclBuf::OsclBuf (int32 size)` [inline]

7.106.2 Member Function Documentation

7.106.2.1 `void OsclBuf::Delete (OsclBuf * a)` [inline, static]

7.106.2.2 `OsclPtr OsclBuf::Des ()` [inline]

7.106.2.3 `OsclPtrC OsclBuf::DesC ()` [inline]

7.106.2.4 `int32 OsclBuf::Length ()` [inline]

7.106.2.5 `OsclBuf* OsclBuf::NewL (int32 size)` [inline, static]

7.106.3 Field Documentation

7.106.3.1 `uint8* OsclBuf::iBuffer`

7.106.3.2 `int32 OsclBuf::iLength`

7.106.3.3 `int32 OsclBuf::iMaxLength`

The documentation for this class was generated from the following file:

- [oscl_file_async_read.h](#)

7.107 OsclCompareLess< T > Class Template Reference

```
#include <oscl_priqueue.h>
```

Public Methods

- int [compare](#) (T &a, T &b) const

```
template<class T> class OsclCompareLess< T >
```

7.107.1 Member Function Documentation

**7.107.1.1 template<class T> int OsclCompareLess< T >::compare (T & a, T & b) const
[inline]**

The documentation for this class was generated from the following file:

- [oscl_priqueue.h](#)

7.108 OsclComponentRegistry Class Reference

```
#include <oscl_registry_serv_impl.h>
```

Public Methods

- [OsclComponentRegistry \(\)](#)
- [~OsclComponentRegistry \(\)](#)
- [int32 Register \(uint32 &aId, OSCL_String &, OsclComponentFactory\)](#)
- [int32 Unregister \(OSCL_String &\)](#)
- [int32 Unregister \(uint32\)](#)
- [OsclComponentFactory FindExact \(OSCL_String &\)](#)
- [void FindHierarchical \(OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &\)](#)
- [void OpenSession \(\)](#)
- [void CloseSession \(\)](#)

Data Fields

- [OsclComponentRegistryData iData](#)
- [OsclMutex iMutex](#)
- [uint32 iComponentIdCounter](#)
- [uint32 iNumSessions](#)

7.108.1 Detailed Description

Thread-safe singleton registry object.

7.108.2 Constructor & Destructor Documentation

7.108.2.1 `OsclComponentRegistry::OsclComponentRegistry ()`

7.108.2.2 `OsclComponentRegistry::~OsclComponentRegistry ()`

7.108.3 Member Function Documentation

7.108.3.1 `void OsclComponentRegistry::CloseSession ()`

7.108.3.2 `OsclComponentFactory OsclComponentRegistry::FindExact (OSCL_String &)`

7.108.3.3 `void OsclComponentRegistry::FindHierarchical (OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &)`

7.108.3.4 `void OsclComponentRegistry::OpenSession ()`

7.108.3.5 `int32 OsclComponentRegistry::Register (uint32 & aId, OSCL_String &, OsclComponentFactory)`

7.108.3.6 `int32 OsclComponentRegistry::Unregister (uint32)`

7.108.3.7 `int32 OsclComponentRegistry::Unregister (OSCL_String &)`

7.108.4 Field Documentation

7.108.4.1 `uint32 OsclComponentRegistry::iComponentIdCounter`

7.108.4.2 `OsclComponentRegistryData OsclComponentRegistry::iData`

7.108.4.3 `OsclMutex OsclComponentRegistry::iMutex`

7.108.4.4 `uint32 OsclComponentRegistry::iNumSessions`

The documentation for this class was generated from the following file:

- `oscl_registry_serv_impl.h`

7.109 OsclComponentRegistryData Class Reference

```
#include <oscl_registry_serv_impl.h>
```

Public Methods

- [OsclComponentRegistryElement * Find \(OSCL_String &, bool aExact\)](#)

Data Fields

- [Oscl_Vector< OsclComponentRegistryElement, OsclMemAllocator > iVec](#)

7.109.1 Detailed Description

Registry

7.109.2 Member Function Documentation

7.109.2.1 [OsclComponentRegistryElement* OsclComponentRegistryData::Find \(OSCL_String &, bool aExact\)](#)

7.109.3 Field Documentation

7.109.3.1 [Oscl_Vector<OsclComponentRegistryElement, OsclMemAllocator> OsclComponentRegistryData::iVec](#)

The documentation for this class was generated from the following file:

- [oscl_registry_serv_impl.h](#)

7.110 OsclComponentRegistryElement Class Reference

```
#include <oscl_registry_serv_impl.h>
```

Public Methods

- [OsclComponentRegistryElement \(OSCL_String &, OsclComponentFactory\)](#)
- [OsclComponentRegistryElement \(const OsclComponentRegistryElement &\)](#)
- [OsclComponentRegistryElement & operator= \(const OsclComponentRegistryElement &src\)](#)
- [~OsclComponentRegistryElement \(\)](#)
- [bool Match \(OSCL_String &aStr, bool aExact\)](#)

Data Fields

- [OSCL_String * iId](#)
- [OsclComponentFactory iFactory](#)
- [uint32 iComponentId](#)

7.110.1 Detailed Description

Data for each registered component.

7.110.2 Constructor & Destructor Documentation

7.110.2.1 OsclComponentRegistryElement::OsclComponentRegistryElement (OSCL_String &, OsclComponentFactory)

7.110.2.2 OsclComponentRegistryElement::OsclComponentRegistryElement (const OsclComponentRegistryElement &)

7.110.2.3 OsclComponentRegistryElement::~OsclComponentRegistryElement ()

7.110.3 Member Function Documentation

7.110.3.1 bool OsclComponentRegistryElement::Match (OSCL_String & aStr, bool aExact)

7.110.3.2 OsclComponentRegistryElement& OsclComponentRegistryElement::operator= (const OsclComponentRegistryElement & src)

7.110.4 Field Documentation

7.110.4.1 uint32 OsclComponentRegistryElement::iComponentId

7.110.4.2 OsclComponentFactory OsclComponentRegistryElement::iFactory

7.110.4.3 OSCL_String* OsclComponentRegistryElement::iId

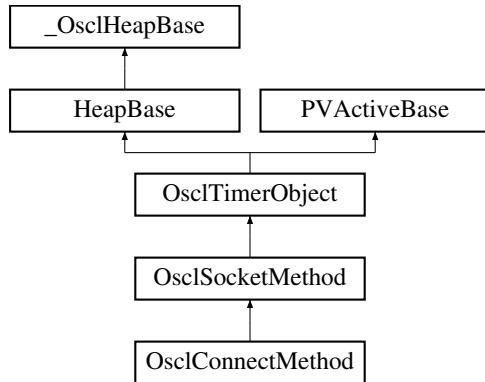
The documentation for this class was generated from the following file:

- [oscl_registry_serv_impl.h](#)

7.111 OsclConnectMethod Class Reference

```
#include <oscl_socket_connect.h>
```

Inheritance diagram for OsclConnectMethod::



Public Methods

- [~OsclConnectMethod \(\)](#)
- [TPVSocketEvent Connect \(OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsclConnectRequest * ConnectRequest \(\)](#)

Static Public Methods

- [OsclConnectMethod * NewL \(OsclIPSocketI &c\)](#)

7.111.1 Constructor & Destructor Documentation

7.111.1.1 OsclConnectMethod::~OsclConnectMethod ()

7.111.2 Member Function Documentation

7.111.2.1 TPVSocketEvent OsclConnectMethod::Connect (OsclNetworkAddress & aAddress, int32 aTimeout)

7.111.2.2 OsclConnectRequest* OsclConnectMethod::ConnectRequest () [inline]

7.111.2.3 OsclConnectMethod* OsclConnectMethod::NewL (OsclIPSocketI & c) [static]

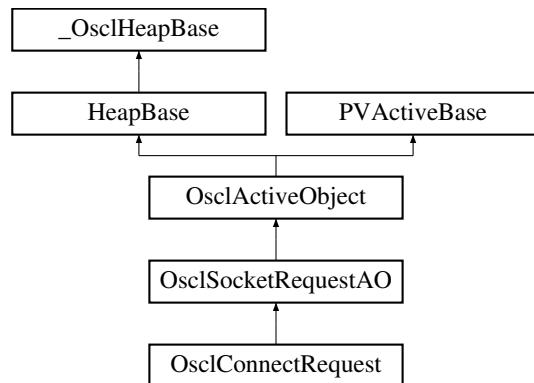
The documentation for this class was generated from the following file:

- [oscl_socket_connect.h](#)

7.112 OsclConnectRequest Class Reference

```
#include <oscl_socket_connect.h>
```

Inheritance diagram for OsclConnectRequest::



Public Methods

- [OsclConnectRequest \(OsclSocketMethod &c\)](#)
- [void Connect \(OsclNetworkAddress &aAddress\)](#)

7.112.1 Detailed Description

This is the AO that interacts with the socket server

7.112.2 Constructor & Destructor Documentation

7.112.2.1 OsclConnectRequest::OsclConnectRequest ([OsclSocketMethod & c](#)) [inline]

7.112.3 Member Function Documentation

7.112.3.1 void OsclConnectRequest::Connect ([OsclNetworkAddress & aAddress](#))

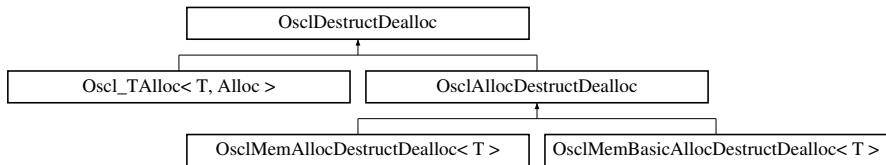
The documentation for this class was generated from the following file:

- [oscl_socket_connect.h](#)

7.113 OsclDestructDealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for OsclDestructDealloc::



Public Methods

- virtual void [destruct_and_dealloc \(OsclAny *ptr\)=0](#)

7.113.1 Member Function Documentation

7.113.1.1 virtual void OsclDestructDealloc::destruct_and_dealloc (OsclAny *ptr) [pure virtual]

Implemented in [Oscl_TAlloc< T, Alloc >](#), [OsclMemAllocDestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [Oscl_TAlloc< entry_type, Alloc >](#), [Oscl_TAlloc< node_type, TagTree_Allocator >](#), [Oscl_TAlloc< node_type, alloc_type >](#), [Oscl_TAlloc< MM_StatsNodeTagTreeType, OsclMemBasicAllocator >](#), [Oscl_TAlloc< char, alloc_type >](#), [Oscl_TAlloc< tag_base_unit, Alloc >](#), [Oscl_TAlloc< PVLogger, alloc_type >](#), and [Oscl_TAlloc< node_type, Alloc >](#).

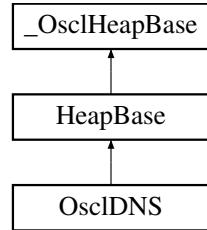
The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

7.114 OsclDNS Class Reference

```
#include <oscl_dns.h>
```

Inheritance diagram for OsclDNS::



Public Methods

- OSCL_IMPORT_REF ~OsclDNS ()
- OSCL_IMPORT_REF TPVDNSEvent GetHostByName (char *name, OsclNetworkAddress &addr, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelGetHostByName ()

Static Public Methods

- OSCL_IMPORT_REF OsclDNS * NewL (Oscl_DefAlloc &alloc, OsclSocketServ &aServ, OsclDNSObserver &aObserver, uint32 aId)

Friends

- class OsclDNSRequestAO

7.114.1 Detailed Description

The DNS class

7.114.2 Constructor & Destructor Documentation

7.114.2.1 OSCL_IMPORT_REF OsclDNS::~OsclDNS ()

Destructor.

Note: the application must de-allocate the DNS object using the same allocator that was passed in the NewL object creation call.

7.114.3 Member Function Documentation

7.114.3.1 OSCL_IMPORT_REF void OsclDNS::CancelGetHostByName ()

Cancel GetHostByName

This method will cancel any pending GetHostByName operation on the current object, causing the GetHostByName to complete with error EPVDNSCancel. If there is no pending GetHostByName operation, this method will have no effect.

7.114.3.2 OSCL_IMPORT_REF TPVDNSEvent OsclDNS::GetHostByName (char * *name*, OsclNetworkAddress & *addr*, int32 *aTimeoutMsec* = -1)

GetHostByName. This is an asynchronous method.

Parameters:

name: Null-terminated string containing the host name.

addr: The output address. The ipAddr field will contain the network address of the host in dotted decimal notation.

aTimeoutMsec: A timeout for the request in milliseconds, or (-1) to indicate infinite wait. @returns: EPVDNSPending for success, EPVDNSFailure for failure.

7.114.3.3 OSCL_IMPORT_REF OsclDNS* OsclDNS::NewL (Oscl_DefAlloc & *alloc*, OsclSocketServ & *aServ*, OsclDNSObserver & *aObserver*, uint32 *aId*) [static]

DNS object creation.

Parameters:

alloc: Memory allocator

aServ: Socket server.

aObserver: DNS Event observer

aId: Unique ID for this DNS object. This ID will be included in all callbacks associated with this DNS object.

7.114.4 Friends And Related Function Documentation

7.114.4.1 friend class OsclDNSRequestAO [friend]

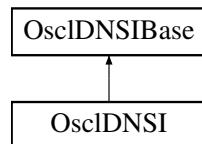
The documentation for this class was generated from the following file:

- [oscl_dns.h](#)

7.115 OsclDNSI Class Reference

```
#include <oscl_dns_imp_pv.h>
```

Inheritance diagram for OsclDNSI::



Public Methods

- [~OsclDNSI \(\)](#)
- int32 [Open \(OsclSocketServI &aServer\)](#)
- int32 [Close \(\)](#)
- void [GetHostByName \(GetHostNameParam &, OsclDNSRequestAO &\)](#)
- void [GetHostByNameSuccess \(GetHostNameParam &\)](#)

Static Public Methods

- OsclDNSI * [NewL \(Oscl_DefAlloc &a\)](#)

Friends

- class [OsclDNSRequest](#)
- class [DNSRequestParam](#)

7.115.1 Detailed Description

OsclDNSI, non-Symbian implementation

7.115.2 Constructor & Destructor Documentation

7.115.2.1 OsclDNSI::~OsclDNSI ()

7.115.3 Member Function Documentation

7.115.3.1 int32 OsclDNSI::Close () [virtual]

Implements [OsclDNSIBase](#).

7.115.3.2 void OsclDNSI::GetHostByName (GetHostNameParam &, OsclDNSRequestAO &) [virtual]

Implements [OsclDNSIBase](#).

7.115.3.3 void OsclDNSI::GetHostByNameSuccess ([GetHostNameParam](#) &) [virtual]

Implements [OsclDNSIBase](#).

7.115.3.4 OsclDNSI* OsclDNSI::NewL ([Oscl_DefAlloc](#) & *a*) [static]

7.115.3.5 int32 OsclDNSI::Open ([OsclSocketServI](#) & *aServer*) [virtual]

Implements [OsclDNSIBase](#).

7.115.4 Friends And Related Function Documentation

7.115.4.1 friend class DNSRequestParam [friend]

7.115.4.2 friend class OsclDNSRequest [friend]

Reimplemented from [OsclDNSIBase](#).

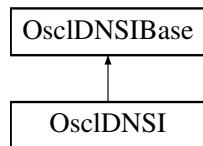
The documentation for this class was generated from the following file:

- [oscl_dns_imp_pv.h](#)

7.116 OsclDNSIBase Class Reference

```
#include <oscl_dns_imp_base.h>
```

Inheritance diagram for OsclDNSIBase::



Public Methods

- virtual ~OsclDNSIBase ()
- virtual int32 Open (OsclSocketServI &aServer)=0
- virtual int32 Close ()=0
- virtual void GetHostByName (GetHostNameParam &, OsclDNSRequestAO &)=0
- virtual void GetHostByNameSuccess (GetHostNameParam &)=0
- void CancelFxn (TPVDNSFxn)

Protected Methods

- OsclDNSIBase (Oscl_DefAlloc &a)
- virtual bool IsReady (OsclDNSRequestAO &aObject)=0
- virtual void CancelGetHostName ()=0

Protected Attributes

- Oscl_DefAlloc & iAlloc
- OsclSocketServI * iSocketServ

Friends

- class OsclDNSRequest
- class OsclGetHostNameRequest

7.116.1 Detailed Description

OsclDNSIBase is a common base class for all implementations.

7.116.2 Constructor & Destructor Documentation

7.116.2.1 `virtual OsclDNSIBase::~OsclDNSIBase () [virtual]`

7.116.2.2 `OsclDNSIBase::OsclDNSIBase (Oscl_DefAlloc & a) [protected]`

7.116.3 Member Function Documentation

7.116.3.1 `void OsclDNSIBase::CancelFxn (TPVDNSFxn)`

7.116.3.2 `virtual void OsclDNSIBase::CancelGetHostByName () [protected, pure virtual]`

7.116.3.3 `virtual int32 OsclDNSIBase::Close () [pure virtual]`

Implemented in [OsclDNSI](#).

7.116.3.4 `virtual void OsclDNSIBase::GetHostByName (GetHostNameParam &, OsclDNSRequestAO &) [pure virtual]`

Implemented in [OsclDNSI](#).

7.116.3.5 `virtual void OsclDNSIBase::GetHostByNameSuccess (GetHostNameParam &) [pure virtual]`

Implemented in [OsclDNSI](#).

7.116.3.6 `virtual bool OsclDNSIBase::IsReady (OsclDNSRequestAO & aObject) [protected, pure virtual]`

7.116.3.7 `virtual int32 OsclDNSIBase::Open (OsclSocketServI & aServer) [pure virtual]`

Implemented in [OsclDNSI](#).

7.116.4 Friends And Related Function Documentation

7.116.4.1 `friend class OsclDNSRequest [friend]`

Reimplemented in [OsclDNSI](#).

7.116.4.2 `friend class OsclGetHostByNameRequest [friend]`

7.116.5 Field Documentation

7.116.5.1 `Oscl_DefAlloc& OsclDNSIBase::iAlloc [protected]`

7.116.5.2 `OsclSocketServI* OsclDNSIBase::iSocketServ [protected]`

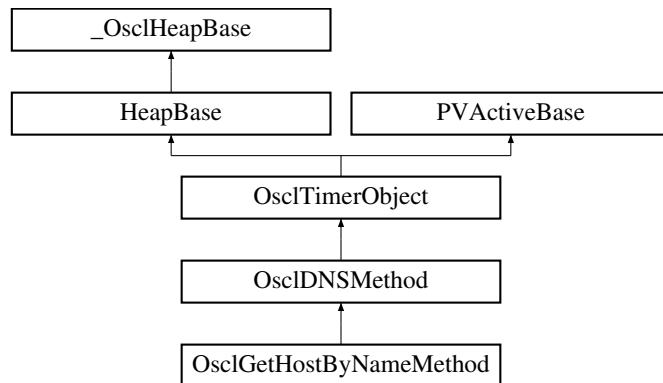
The documentation for this class was generated from the following file:

- [oscl_dns_imp_base.h](#)

7.117 OsclDNSMethod Class Reference

```
#include <oscl_dns_method.h>
```

Inheritance diagram for OsclDNSMethod::



Public Methods

- [OsclDNSMethod \(Oscl_DefAlloc &a, const char *name, TPVDNSFxn fxn\)](#)
- void [Abort \(\)](#)
- void [AbortAll \(\)](#)
- void [CancelMethod \(\)](#)
- void [Run \(\)](#)

Data Fields

- [OsclDNSObserver * iDNSObserver](#)
- uint32 [iId](#)
- [Oscl_DefAlloc & iAlloc](#)
- [TPVDNSFxn iDNSFxn](#)
- [PVLogger * iLogger](#)

Protected Methods

- void [ConstructL \(OsclDNSObserver *aObserver, OsclDNSRequestAO *aAO, uint32 aId\)](#)
- bool [StartMethod \(int32 aTimeoutMsec\)](#)
- void [MethodDone \(\)](#)

Protected Attributes

- [OsclDNSRequestAO * iDNSRequestAO](#)

7.117.1 Detailed Description

This is the base class for all socket methods. It provides the timeout on socket requests.

7.117.2 Constructor & Destructor Documentation

7.117.2.1 OsclDNSMethod::OsclDNSMethod ([Oscl_DefAlloc](#) & *a*, const char * *name*, [TPVDNSFxn](#) *fxn*) [inline]

7.117.3 Member Function Documentation

7.117.3.1 void OsclDNSMethod::Abort ()

7.117.3.2 void OsclDNSMethod::AbortAll ()

7.117.3.3 void OsclDNSMethod::CancelMethod ()

7.117.3.4 void OsclDNSMethod::ConstructL ([OsclDNSObserver](#) * *aObserver*, [OsclDNSRequestAO](#) * *aAO*, uint32 *aId*) [protected]

7.117.3.5 void OsclDNSMethod::MethodDone () [protected]

7.117.3.6 void OsclDNSMethod::Run () [virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request

2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implements [PVActiveBase](#).

7.117.3.7 **bool OsclDNSMethod::StartMethod (int32 *aTimeoutMsec*)** [protected]

7.117.4 Field Documentation

7.117.4.1 **Oscl_DefAlloc& OsclDNSMethod::iAlloc**

7.117.4.2 **TPVDNSFxn OsclDNSMethod::iDNSFxn**

7.117.4.3 **OsclDNSObserver* OsclDNSMethod::iDNSObserver**

7.117.4.4 **OsclDNSRequestAO* OsclDNSMethod::iDNSRequestAO** [protected]

7.117.4.5 **uint32 OsclDNSMethod::iId**

7.117.4.6 **PVLogger* OsclDNSMethod::iLogger**

The documentation for this class was generated from the following file:

- [oscl_dns_method.h](#)

7.118 OsclDNSObserver Class Reference

```
#include <oscl_dns.h>
```

Public Methods

- virtual OSCL_IMPORT_REF void [HandleDNSEvent](#) (int32 aId, [TPVDNSFxn](#) aFxn, [TPVDNSEvent](#) aEvent, int32 aError)=0
- virtual [~OsclDNSObserver](#) ()

7.118.1 Detailed Description

DNS event observer. The client implements this to get asynchronous command completion.

7.118.2 Constructor & Destructor Documentation

7.118.2.1 virtual OsclDNSObserver::~OsclDNSObserver () [inline, virtual]

7.118.3 Member Function Documentation

**7.118.3.1 virtual OSCL_IMPORT_REF void OsclDNSObserver::HandleDNSEvent (int32 *aId*,
[TPVDNSFxn](#) *aFxn*, [TPVDNSEvent](#) *aEvent*, int32 *aError*) [pure virtual]**

DNS Event callback.

Parameters:

***aId*:** The ID that was supplied when the DNS object was created.

***aEvent*:** Function completion event. Will be EPVDNSSuccess, EPVDNSTimeout, or EPVDNSFailure.

***aError*:** When the event is EPVDNSFailure, this may contain a platform-specific error code, or zero if none is available.

The documentation for this class was generated from the following file:

- [oscl_dns.h](#)

7.119 OsclDNSRequest Class Reference

```
#include <oscl_dns_request.h>
```

Public Methods

- [OsclDNSRequest \(\)](#)
- [~OsclDNSRequest \(\)](#)
- void [CancelRequest \(\)](#)
- void [Complete \(bool, int32 aStatus, int32 aSockErr\)](#)
- void [Activate \(DNSRequestParam *iParam, OsclDNSRequestAO &a\)](#)

Data Fields

- [OsclDNSRequestAO * iDNSRequestAO](#)
- [DNSRequestParam * iDNSRequestParam](#)
- bool [iActive](#)

7.119.1 Detailed Description

This class defines the interface to the dns implementation threads.

7.119.2 Constructor & Destructor Documentation

7.119.2.1 [OsclDNSRequest::OsclDNSRequest \(\) \[inline\]](#)

7.119.2.2 [OsclDNSRequest::~OsclDNSRequest \(\) \[inline\]](#)

7.119.3 Member Function Documentation

7.119.3.1 void [OsclDNSRequest::Activate \(DNSRequestParam * iParam, OsclDNSRequestAO & a\)](#)

7.119.3.2 void [OsclDNSRequest::CancelRequest \(\)](#)

7.119.3.3 void [OsclDNSRequest::Complete \(bool, int32 aStatus, int32 aSockErr\)](#)

7.119.4 Field Documentation

7.119.4.1 bool [OsclDNSRequest::iActive](#)

7.119.4.2 [OsclDNSRequestAO* OsclDNSRequest::iDNSRequestAO](#)

7.119.4.3 [DNSRequestParam* OsclDNSRequest::iDNSRequestParam](#)

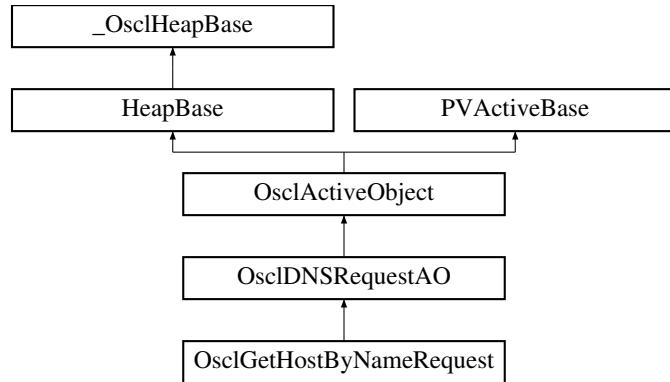
The documentation for this class was generated from the following file:

- [oscl_dns_request.h](#)

7.120 OsclDNSRequestAO Class Reference

```
#include <oscl_dns_method.h>
```

Inheritance diagram for OsclDNSRequestAO::



Protected Methods

- [OsclDNSRequestAO](#) (const char *name)
- void [ConstructL](#) ([OsclDNSI](#) *aDNS, [OsclDNSMethod](#) *aMethod)
- void [Abort](#) ()
- void [NewRequest](#) ()
- void [RequestDone](#) ()
- int [GetSocketError](#) ()
- [OsclSocketServI](#) * [Serv](#) ()
- void [DoCancel](#) ()
- void [Run](#) ()
- virtual void [Success](#) ()

Protected Attributes

- [OsclDNSI](#) * [iDNSI](#)
- [OsclDNSMethod](#) * [iDNSMethod](#)
- int32 [iSocketError](#)
- [PVLogger](#) * [iLogger](#)

Friends

- class [OsclDNSI](#)
- class [OsclDNSMethod](#)
- class [OsclDNSRequest](#)
- class [DNSRequestParam](#)

7.120.1 Detailed Description

This is the base class for all requests to the socket server.

7.120.2 Constructor & Destructor Documentation

7.120.2.1 OsclDNSRequestAO::OsclDNSRequestAO (const char * *name*) [inline, protected]

7.120.3 Member Function Documentation

7.120.3.1 void OsclDNSRequestAO::Abort () [inline, protected]

7.120.3.2 void OsclDNSRequestAO::ConstructL (OsclDNSI * *aDNS*, OsclDNSMethod * *aMethod*) [inline, protected]

7.120.3.3 void OsclDNSRequestAO::DoCancel () [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Reimplemented from [OsclActiveObject](#).

7.120.3.4 int OsclDNSRequestAO::GetSocketError () [protected]

7.120.3.5 void OsclDNSRequestAO::NewRequest () [protected]

7.120.3.6 void OsclDNSRequestAO::RequestDone () [protected]

7.120.3.7 void OsclDNSRequestAO::Run () [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's [Run\(\)](#) function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

7.120.3.8 **OsclSocketServI* OsclDNSRequestAO::Serv ()** [protected]

7.120.3.9 **virtual void OsclDNSRequestAO::Success ()** [inline, protected, virtual]

7.120.4 Friends And Related Function Documentation

7.120.4.1 **friend class DNSRequestParam** [friend]

7.120.4.2 **friend class OsclDNSI** [friend]

7.120.4.3 **friend class OsclDNSMethod** [friend]

7.120.4.4 **friend class OsclDNSRequest** [friend]

7.120.5 Field Documentation

7.120.5.1 **OsclDNSI* OsclDNSRequestAO::iDNSI** [protected]

7.120.5.2 **OsclDNSMethod* OsclDNSRequestAO::iDNSMethod** [protected]

7.120.5.3 **PVLogger* OsclDNSRequestAO::iLogger** [protected]

7.120.5.4 **int32 OsclDNSRequestAO::iSocketError** [protected]

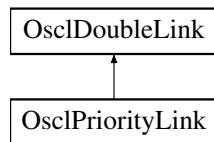
The documentation for this class was generated from the following file:

- [oscl_dns_method.h](#)

7.121 OsclDoubleLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleLink::



Public Methods

- [OsclDoubleLink \(\)](#)
- void [Remove \(\)](#)
- void [InsertAfter \(OsclDoubleLink *aLink\)](#)
- void [InsertBefore \(OsclDoubleLink *aLink\)](#)

Data Fields

- OsclDoubleLink * [iNext](#)
- OsclDoubleLink * [iPrev](#)

7.121.1 Constructor & Destructor Documentation

7.121.1.1 [OsclDoubleLink::OsclDoubleLink \(\) \[inline\]](#)

7.121.2 Member Function Documentation

7.121.2.1 [void OsclDoubleLink::InsertAfter \(OsclDoubleLink * *aLink*\)](#)

7.121.2.2 [void OsclDoubleLink::InsertBefore \(OsclDoubleLink * *aLink*\)](#)

7.121.2.3 [void OsclDoubleLink::Remove \(\)](#)

7.121.3 Field Documentation

7.121.3.1 [OsclDoubleLink* OsclDoubleLink::iNext](#)

7.121.3.2 [OsclDoubleLink* OsclDoubleLink::iPrev](#)

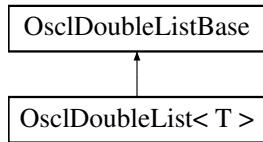
The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

7.122 OsclDoubleList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleList< T >::



Public Methods

- OSCL_INLINE OsclDoubleList()
- OSCL_INLINE OsclDoubleList(int32 anOffset)
- OSCL_INLINE void InsertHead(T &aRef)
- OSCL_INLINE void InsertTail(T &aRef)
- OSCL_INLINE bool IsHead(const T *aPtr) const
- OSCL_INLINE bool IsTail(const T *aPtr) const
- OSCL_INLINE T * Head() const
- OSCL_INLINE T * Tail() const

```
template<class T> class OsclDoubleList< T >
```

7.122.1 Constructor & Destructor Documentation

7.122.1.1 template<class T> OSCL_INLINE OsclDoubleList< T >::OsclDoubleList()

**7.122.1.2 template<class T> OSCL_INLINE OsclDoubleList< T >::OsclDoubleList(int32
anOffset)**

7.122.2 Member Function Documentation

7.122.2.1 template<class T> OSCL_INLINE T* OsclDoubleList< T >::Head()

7.122.2.2 template<class T> OSCL_INLINE void OsclDoubleList< T >::InsertHead(T & aRef)

7.122.2.3 template<class T> OSCL_INLINE void OsclDoubleList< T >::InsertTail(T & aRef)

**7.122.2.4 template<class T> OSCL_INLINE bool OsclDoubleList< T >::IsHead(const T * aPtr)
const**

**7.122.2.5 template<class T> OSCL_INLINE bool OsclDoubleList< T >::IsTail(const T * aPtr)
const**

7.122.2.6 template<class T> OSCL_INLINE T* OsclDoubleList< T >::Tail()

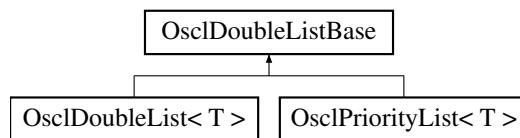
The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

7.123 OsclDoubleListBase Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleListBase::



Public Methods

- bool [IsEmpty \(\) const](#)
- void [SetOffset \(int32 anOffset\)](#)
- void [Reset \(\)](#)
- [OsclDoubleLink * getHead \(\)](#)
- int32 [getOffset \(\)](#)

Protected Methods

- [OsclDoubleListBase \(\)](#)
- [OsclDoubleListBase \(int32 anOffset\)](#)
- void [InsertHead \(OsclAny *aPtr\)](#)
- void [InsertTail \(OsclAny *aPtr\)](#)
- void [Insert \(OsclAny *aPtr\)](#)

Protected Attributes

- [OsclDoubleLink iHead](#)
- int32 [iOffset](#)

7.123.1 Constructor & Destructor Documentation

7.123.1.1 **OsclDoubleListBase::OsclDoubleListBase ()** [protected]

7.123.1.2 **OsclDoubleListBase::OsclDoubleListBase (int32 *anOffset*)** [protected]

7.123.2 Member Function Documentation

7.123.2.1 **OsclDoubleLink* OsclDoubleListBase::getHead ()** [inline]

7.123.2.2 **int32 OsclDoubleListBase::getOffset ()** [inline]

7.123.2.3 **void OsclDoubleListBase::Insert (OsclAny * *aPtr*)** [protected]

7.123.2.4 **void OsclDoubleListBase::InsertHead (OsclAny * *aPtr*)** [protected]

7.123.2.5 **void OsclDoubleListBase::InsertTail (OsclAny * *aPtr*)** [protected]

7.123.2.6 **bool OsclDoubleListBase::IsEmpty ()**

7.123.2.7 **void OsclDoubleListBase::Reset ()**

7.123.2.8 **void OsclDoubleListBase::SetOffset (int32 *anOffset*)**

7.123.3 Field Documentation

7.123.3.1 **OsclDoubleLink OsclDoubleListBase::iHead** [protected]

7.123.3.2 **int32 OsclDoubleListBase::iOffset** [protected]

The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

7.124 OsclDoubleRunner< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Public Methods

- [OsclDoubleRunner \(OsclDoubleListBase &aQue\)](#)
- [void Set \(T &aLink\)](#)
- [operator T * \(\)](#)
- [T * operator++ \(int\)](#)
- [T * operator- \(int\)](#)
- [void SetToHead \(\)](#)
- [void SetToTail \(\)](#)

Protected Attributes

- [int32 iOffset](#)
- [OsclDoubleLink * iHead](#)
- [OsclDoubleLink * iNext](#)

template<class T> class OsclDoubleRunner< T >

7.124.1 Constructor & Destructor Documentation

7.124.1.1 template<class T> OsclDoubleRunner< T >::OsclDoubleRunner ([OsclDoubleListBase & aQue](#)) [inline]

7.124.2 Member Function Documentation

7.124.2.1 template<class T> OsclDoubleRunner< T >::operator T * () [inline]

7.124.2.2 template<class T> T* OsclDoubleRunner< T >::operator++ (int) [inline]

7.124.2.3 template<class T> T* OsclDoubleRunner< T >::operator- (int)

7.124.2.4 template<class T> void OsclDoubleRunner< T >::Set (T & aLink) [inline]

7.124.2.5 template<class T> void OsclDoubleRunner< T >::SetToHead () [inline]

7.124.2.6 template<class T> void OsclDoubleRunner< T >::SetToTail () [inline]

7.124.3 Field Documentation

7.124.3.1 template<class T> [OsclDoubleLink* OsclDoubleRunner< T >::iHead](#) [protected]

7.124.3.2 template<class T> [OsclDoubleLink* OsclDoubleRunner< T >::iNext](#) [protected]

7.124.3.3 template<class T> int32 [OsclDoubleRunner< T >::iOffset](#) [protected]

The documentation for this class was generated from the following file:

-
- [oscl_double_list.h](#)

7.125 OsclError Class Reference

```
#include <oscl_error.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [PushL \(_OsclHeapBase *aPtr\)](#)
- OSCL_IMPORT_REF void [PushL \(OsclAny *aPtr\)](#)
- OSCL_IMPORT_REF void [PushL \(OsclTrapItem anItem\)](#)
- OSCL_IMPORT_REF void [Pop \(\)](#)
- OSCL_IMPORT_REF void [Pop \(int32 aCount\)](#)
- OSCL_IMPORT_REF void [PopDealloc \(\)](#)
- OSCL_IMPORT_REF void [PopDealloc \(int32 aCount\)](#)
- OSCL_IMPORT_REF void [Leave \(int32 aReason\)](#)
- OSCL_IMPORT_REF void [LeaveIfNull \(OsclAny *a\)](#)
- OSCL_IMPORT_REF void [LeaveIfError \(int32 aReason\)](#)

7.125.1 Detailed Description

User Error class

7.125.2 Member Function Documentation

7.125.2.1 OSCL_IMPORT_REF void OsclError::Leave (int32 *aReason*) [static]

Do a Leave error, with the given reason code. When a leave occurs, all items on the cleanup stack for the current trap level will be destroyed, and execution will jump to the trap handler.

7.125.2.2 OSCL_IMPORT_REF void OsclError::LeaveIfError (int32 *aReason*) [static]

Evaluate the input parameter, and if it is an error code (non-zero), then do a Leave with the provided reason code.

7.125.2.3 OSCL_IMPORT_REF void OsclError::LeaveIfNull (OsclAny * *a*) [static]

Evaluate the input parameter, and if it is null, do a Leave with OsclErrNoMemory reason code.

7.125.2.4 OSCL_IMPORT_REF void OsclError::Pop (int32 *aCount*) [static]

Pop the cleanup stack N times

7.125.2.5 OSCL_IMPORT_REF void OsclError::Pop () [static]

Pop the cleanup stack

7.125.2.6 OSCL_IMPORT_REF void OsclError::PopDealloc (int32 *aCount*) [static]

PopDealloc N times

7.125.2.7 OSCL_IMPORT_REF void OsclError::PopDealloc () [static]

Destroy the item on the top of the cleanup stack and pop it

7.125.2.8 OSCL_IMPORT_REF void OsclError::PushL ([OsclTrapItem](#) *anItem*) [static]

Push an [OsclTrapItem](#) onto the cleanup stack

7.125.2.9 OSCL_IMPORT_REF void OsclError::PushL ([OsclAny](#) * *aPtr*) [static]

Push an OsclAny item onto the cleanup stack.

7.125.2.10 OSCL_IMPORT_REF void OsclError::PushL ([_OsclHeapBase](#) * *aPtr*) [static]

Push an [_OsclHeapBase](#) item onto the cleanup stack.

The documentation for this class was generated from the following file:

- [oscl_error.h](#)

7.126 OsclErrorAllocator Class Reference

This class provides static methods to invoke the user defined memory allocation routines.

```
#include <oscl_error_allocator.h>
```

Public Methods

- **OsclErrorAllocator (Oscl_DefAlloc *allocator)**
constructor method
- **void * operator new (uint32 size, OsclAny *aPtr)**
placement new operator that allocates memory using the user defined methods
- **void operator delete (OsclAny *aPtr, OsclAny *aPtr2)**
delete operator that doesn't do anything, user has to deallocate manually

Static Public Methods

- **OsclAny * allocate (uint32 aSize)**
static method to allocate a block of memory on heap
- **OsclAny deallocate (OsclAny *aPointer)**
static method to deallocate a block of memory on heap

7.126.1 Detailed Description

This class provides static methods to invoke the user defined memory allocation routines.

This class must be instantiated before the static methods are called, else asserts will happen

7.126.2 Constructor & Destructor Documentation

7.126.2.1 OsclErrorAllocator::OsclErrorAllocator (**Oscl_DefAlloc * allocator**) [inline]

constructor method

Parameters:

allocator - a pointer to the concrete object that provides the allocator/deallocator

7.126.3 Member Function Documentation

7.126.3.1 OsclAny* OsclErrorAllocator::allocate (uint32 **aSize**) [inline, static]

static method to allocate a block of memory on heap

Parameters:

aSize - number of bytes to allocate

7.126.3.2 OsclAny OsclErrorAllocator::deallocate (OsclAny * aPointer) [inline, static]

static method to deallocate a block of memory on heap

Parameters:

aPointer - pointer to block of memory to be deallocated

7.126.3.3 void OsclErrorAllocator::operator delete (OsclAny * aPtr, OsclAny * aPtr2) [inline]

delete operator that doesn't do anything, user has to deallocate manually

7.126.3.4 void* OsclErrorAllocator::operator new (uint32 size, OsclAny * aPtr) [inline]

placement new operator that allocates memory using the user defined methods

The documentation for this class was generated from the following file:

- [oscl_error_allocator.h](#)

7.127 OsclErrorTrap Class Reference

```
#include <oscl_error.h>
```

Static Public Methods

- OSCL_IMPORT_REF int32 [Init \(Oscl_DefAlloc *aAlloc=NULL\)](#)
- OSCL_IMPORT_REF int32 [Cleanup \(\)](#)
- OSCL_IMPORT_REF [OsclErrorTrapImp * GetErrorTrapImp \(\)](#)

7.127.1 Member Function Documentation

7.127.1.1 OSCL_IMPORT_REF int32 OsclErrorTrap::Cleanup () [static]

Cleanup and destroy error trap for the calling thread.

Returns:

0 for success, or an error

7.127.1.2 OSCL_IMPORT_REF [OsclErrorTrapImp*](#) OsclErrorTrap::GetErrorTrapImp () [static]

Get the ErrorTrapImp for the current thread. Leaves on error.

7.127.1.3 OSCL_IMPORT_REF int32 OsclErrorTrap::Init ([Oscl_DefAlloc * aAlloc = NULL](#)) [static]

Allocate and initialize error trap for the calling thread.

Parameters:

aAlloc: optional, allocator to use for the internal implementation.

Returns:

0 for success, or an error

The documentation for this class was generated from the following file:

- [oscl_error.h](#)

7.128 OsclErrorTrapImp Class Reference

```
#include <oscl_error_trapcleanup.h>
```

Public Methods

- OSCL_IMPORT_REF void [UnTrap \(\)](#)

Static Public Methods

- OSCL_IMPORT_REF OsclErrorTrapImp * [Trap \(\)](#)
- OSCL_IMPORT_REF OsclErrorTrapImp * [TrapNoTls \(OsclErrorTrapImp *\)](#)

Data Fields

- [OsclJump * iJumpData](#)
- int32 [iLeave](#)
- [OsclTrapStack * iTrapStack](#)

Friends

- class [OsclErrorTrap](#)
- class [OsclError](#)
- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclJump](#)
- class [OsclJumpMark](#)
- class [OsclTrapStack](#)
- class [CPVInterfaceProxy](#)
- class [OsclScheduler](#)

7.128.1 Detailed Description

A per-thread cleanup stack with nested trap support.

7.128.2 Member Function Documentation

7.128.2.1 OSCL_IMPORT_REF OsclErrorTrapImp* OsclErrorTrapImp::Trap () [static]

PV trap cleanup. Public for use in macros only.

7.128.2.2 OSCL_IMPORT_REF OsclErrorTrapImp* OsclErrorTrapImp::TrapNoTls (OsclErrorTrapImp *) [static]

7.128.2.3 OSCL_IMPORT_REF void OsclErrorTrapImp::UnTrap ()

these are used in public macros, but aren't intended as public methods or members.

7.128.3 Friends And Related Function Documentation

7.128.3.1 **friend class CPVInterfaceProxy [friend]**

7.128.3.2 **friend class OsclError [friend]**

7.128.3.3 **friend class OsclErrorTrap [friend]**

7.128.3.4 **friend class OsclExecScheduler [friend]**

7.128.3.5 **friend class OsclExecSchedulerCommonBase [friend]**

7.128.3.6 **friend class OsclJump [friend]**

7.128.3.7 **friend class OsclJumpMark [friend]**

7.128.3.8 **friend class OsclScheduler [friend]**

7.128.3.9 **friend class OsclTrapStack [friend]**

7.128.4 Field Documentation

7.128.4.1 **OsclJump* OsclErrorTrapImp::iJumpData**

7.128.4.2 **int32 OsclErrorTrapImp::iLeave**

7.128.4.3 **OsclTrapStack* OsclErrorTrapImp::iTrapStack**

The documentation for this class was generated from the following file:

- [oscl_error_trapcleanup.h](#)

7.129 OsclException< LeaveCode > Class Template Reference

`oscl_exception.h` contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

```
#include <oscl_exception.h>
```

Public Methods

- `OsclException ()`

Static Public Methods

- `int getLeaveCode ()`

7.129.1 Detailed Description

`template<int LeaveCode> class OsclException< LeaveCode >`

`oscl_exception.h` contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

All PacketVideo exception classes will be derived from the OsclException class. Each derived class will have a static function where the leave code can be obtained. This avoids the issue of having static members in a DLL. The function needs to be static so it can be called without an instance of the class

7.129.2 Constructor & Destructor Documentation

7.129.2.1 `template<int LeaveCode> OsclException< LeaveCode >::OsclException ()`
[inline]

7.129.3 Member Function Documentation

7.129.3.1 `template<int LeaveCode> int OsclException< LeaveCode >::getLeaveCode ()`
[inline, static]

The documentation for this class was generated from the following file:

- `oscl_exception.h`

7.130 OsclExclusiveArrayPtr< T > Class Template Reference

The OsclExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory.

```
#include <oscl_exclusive_ptr.h>
```

Public Methods

- **OsclExclusiveArrayPtr (T *inPtr=0)**
Default constructor Initializes the pointer and takes ownership.
- **OsclExclusiveArrayPtr (OsclExclusiveArrayPtr< T > &_Y)**
Copy constructor.
- **OsclExclusiveArrayPtr< T > & operator= (OsclExclusiveArrayPtr< T > &_Y)**
Assignment operator from an another OsclExclusiveArrayPtr.
- **virtual ~OsclExclusiveArrayPtr ()**
Destructor.
- **T & operator* () const**
The indirection operator () accesses a value indirectly, through a pointer.*
- **T * operator-> () const**
The indirection operator (->) accesses a value indirectly, through a pointer.
- **T * get () const**
get() method returns the pointer, currently owned by the class.
- **T * release ()**
release() method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.
- **bool set (T *ptr)**
set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

Protected Attributes

- **T * _Ptr**

7.130.1 Detailed Description

template<class T> class OsclExclusiveArrayPtr< T >

The OsclExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an [OsclExclusivePtr](#) object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The [OsclExclusivePtr](#) is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

7.130.2 Constructor & Destructor Documentation

7.130.2.1 template<class T> OsclExclusiveArrayPtr< T >::OsclExclusiveArrayPtr (T * *inPtr* = 0) [inline, explicit]

Default constructor Initializes the pointer and takes ownership.

7.130.2.2 template<class T> OsclExclusiveArrayPtr< T >::OsclExclusiveArrayPtr (OsclExclusiveArrayPtr< T > & *_Y*) [inline]

Copy constructor.

Initializes the pointer and takes ownership from another OsclExclusiveArrayPtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

7.130.2.3 template<class T> virtual OsclExclusiveArrayPtr< T >::~OsclExclusiveArrayPtr () [inline, virtual]

Destructor.

The pointer is deleted in case this class still has ownership

7.130.3 Member Function Documentation

7.130.3.1 template<class T> T* OsclExclusiveArrayPtr< T >::get () const [inline]

[get\(\)](#) method returns the pointer, currently owned by the class.

7.130.3.2 template<class T> T& OsclExclusiveArrayPtr< T >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusiveArrayPtr can be used like the regular pointer that it was initialized with.

7.130.3.3 template<class T> T* OsclExclusiveArrayPtr< T >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusiveArrayPtr can be used like the regular pointer that it was initialized with.

7.130.3.4 template<class T> OsclExclusiveArrayPtr<T>& OsclExclusiveArrayPtr< T >::operator= (OsclExclusiveArrayPtr< T > & _Y) [inline]

Assignment operator from an another OsclExclusiveArrayPtr.

Parameters:

_Y The value parameter should be another OsclExclusiveArrayPtr

Returns:

Returns a reference to this OsclExclusiveArrayPtr instance with pointer initialized.

Precondition:

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the OsclExclusiveArrayPtr given as the input parameter. The ownership of the pointer is transferred.

7.130.3.5 template<class T> T* OsclExclusiveArrayPtr< T >::release () [inline]

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

7.130.3.6 template<class T> bool OsclExclusiveArrayPtr< T >::set (T *ptr) [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

7.130.4 Field Documentation

7.130.4.1 template<class T> T* OsclExclusiveArrayPtr< T >::_Ptr [protected]

The documentation for this class was generated from the following file:

- [oscl_exclusive_ptr.h](#)

7.131 OsclExclusivePtr< T > Class Template Reference

The OsclExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory.

```
#include <oscl_exclusive_ptr.h>
```

Public Methods

- **OsclExclusivePtr** (T *inPtr=0)
Default constructor Initializes the pointer and takes ownership.
- **OsclExclusivePtr** (OsclExclusivePtr< T > &_Y)
Copy constructor.
- OsclExclusivePtr< T > & **operator=** (OsclExclusivePtr< T > &_Y)
Assignment operator from an another OsclExclusivePtr.
- virtual ~**OsclExclusivePtr** ()
Destructor.
- T & **operator *** () const
The indirection operator () accesses a value indirectly, through a pointer.*
- T * **operator ->** () const
The indirection operator (->) accesses a value indirectly, through a pointer.
- T * **get** () const
***get()** method returns the pointer, currently owned by the class.*
- T * **release** ()
***release()** method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool **set** (T *ptr)
***set()** method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

Protected Attributes

- T * **_Ptr**

7.131.1 Detailed Description

template<class T> class OsclExclusivePtr< T >

The OsclExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an OsclExclusivePtr object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The OsclExclusivePtr is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

7.131.2 Constructor & Destructor Documentation

7.131.2.1 template<class T> OsclExclusivePtr< T >::OsclExclusivePtr (T * *inPtr* = 0) [inline, explicit]

Default constructor Initializes the pointer and takes ownership.

7.131.2.2 template<class T> OsclExclusivePtr< T >::OsclExclusivePtr (OsclExclusivePtr< T > & *_Y*) [inline]

Copy constructor.

Initializes the pointer and takes ownership from another OsclExclusivePtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

7.131.2.3 template<class T> virtual OsclExclusivePtr< T >::~OsclExclusivePtr () [inline, virtual]

Destructor.

The pointer is deleted in case this class still has ownership

7.131.3 Member Function Documentation

7.131.3.1 template<class T> T* OsclExclusivePtr< T >::get () const [inline]

[get\(\)](#) method returns the pointer, currently owned by the class.

7.131.3.2 template<class T> T& OsclExclusivePtr< T >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusivePtr can be used like the regular pointer that it was initialized with.

7.131.3.3 template<class T> T* OsclExclusivePtr< T >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusivePtr can be used like the regular pointer that it was initialized with.

7.131.3.4 template<class T> OsclExclusivePtr<T>& OsclExclusivePtr< T >::operator= (OsclExclusivePtr< T > & _Y) [inline]

Assignment operator from an another OsclExclusivePtr.

Parameters:

_Y The value parameter should be another OsclExclusivePtr

Returns:

Returns a reference to this OsclExclusivePtr instance with pointer initialized.

Precondition:

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the OsclExclusivePtr given as the input parameter. The ownership of the pointer is transferred.

7.131.3.5 template<class T> T* OsclExclusivePtr< T >::release () [inline]

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

7.131.3.6 template<class T> bool OsclExclusivePtr< T >::set (T *ptr) [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

7.131.4 Field Documentation

7.131.4.1 template<class T> T* OsclExclusivePtr< T >::_Ptr [protected]

The documentation for this class was generated from the following file:

- [oscl_exclusive_ptr.h](#)

7.132 OsclExclusivePtrA< T, Alloc > Class Template Reference

The OsclExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory.

```
#include <oscl_exclusive_ptr.h>
```

Public Methods

- **OsclExclusivePtrA** (T *inPtr=0)
Default constructor Initializes the pointer and takes ownership.
- **OsclExclusivePtrA** (OsclExclusivePtrA< T, Alloc > &_Y)
Copy constructor.
- OsclExclusivePtrA< T, Alloc > & **operator=** (OsclExclusivePtrA< T, Alloc > &_Y)
Assignment operator from an another OsclExclusiveArrayPtr.
- virtual ~**OsclExclusivePtrA** ()
Destructor.
- T & **operator *** () const
The indirection operator () accesses a value indirectly, through a pointer.*
- T * **operator ->** () const
The indirection operator (->) accesses a value indirectly, through a pointer.
- T * **get** () const
***get()** method returns the pointer, currently owned by the class.*
- T * **release** ()
***release()** method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool **set** (T *ptr)
***set()** method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

Protected Attributes

- T * **_Ptr**

7.132.1 Detailed Description

template<class T, class Alloc> class OsclExclusivePtrA< T, Alloc >

The OsclExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an [OsclExclusivePtr](#) object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The [OsclExclusivePtr](#) is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

7.132.2 Constructor & Destructor Documentation

7.132.2.1 template<class T, class Alloc> OsclExclusivePtrA< T, Alloc >::OsclExclusivePtrA (T * *inPtr* = 0) [inline, explicit]

Default constructor Initializes the pointer and takes ownership.

**7.132.2.2 template<class T, class Alloc> OsclExclusivePtrA< T, Alloc >::OsclExclusivePtrA
(OsclExclusivePtrA< T, Alloc > & *_Y*) [inline]**

Copy constructor.

Initializes the pointer and takes ownership from another [OsclExclusiveArrayPtr](#). Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

7.132.2.3 template<class T, class Alloc> virtual OsclExclusivePtrA< T, Alloc >::~OsclExclusivePtrA () [inline, virtual]

Destructor.

The pointer is deleted in case this class still has ownership

7.132.3 Member Function Documentation

7.132.3.1 template<class T, class Alloc> T* OsclExclusivePtrA< T, Alloc >::get () const [inline]

[get\(\)](#) method returns the pointer, currently owned by the class.

7.132.3.2 template<class T, class Alloc> T& OsclExclusivePtrA< T, Alloc >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the [OsclExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

7.132.3.3 template<class T, class Alloc> T* OsclExclusivePtrA< T, Alloc >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the [OsclExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

7.132.3.4 template<class T, class Alloc> OsclExclusivePtrA<T, Alloc>& OsclExclusivePtrA< T, Alloc >::operator= (OsclExclusivePtrA< T, Alloc > & _Y) [inline]

Assignment operator from an another [OsclExclusiveArrayPtr](#).

Parameters:

_Y The value parameter should be another [OsclExclusiveArrayPtr](#)

Returns:

Returns a reference to this [OsclExclusiveArrayPtr](#) instance with pointer initialized.

Precondition:

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the [OsclExclusiveArrayPtr](#) given as the input parameter. The ownership of the pointer is transferred.

7.132.3.5 template<class T, class Alloc> T* OsclExclusivePtrA< T, Alloc >::release () [inline]

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

7.132.3.6 template<class T, class Alloc> bool OsclExclusivePtrA< T, Alloc >::set (T *ptr) [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

7.132.4 Field Documentation

7.132.4.1 template<class T, class Alloc> T* OsclExclusivePtrA< T, Alloc >::_Ptr [protected]

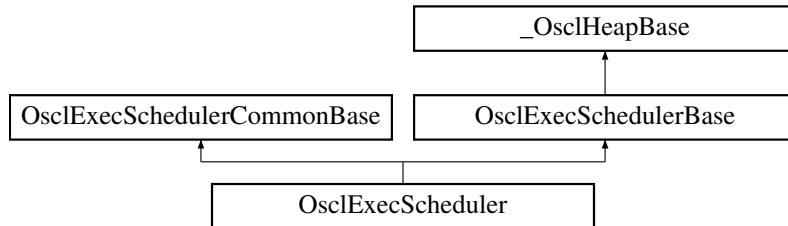
The documentation for this class was generated from the following file:

- [oscl_exclusive_ptr.h](#)

7.133 OsclExecScheduler Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for OsclExecScheduler::



Public Methods

- OSCL_IMPORT_REF void [RunSchedulerNonBlocking](#) (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)
- OSCL_IMPORT_REF void [RegisterForCallback](#) ([OsclSchedulerObserver](#) *aCallback, [OsclAny](#) *aCallbackContext)

Static Public Methods

- OSCL_IMPORT_REF [OsclExecScheduler](#) * [Current](#) ()

Friends

- class [OsclScheduler](#)

7.133.1 Member Function Documentation

7.133.1.1 OSCL_IMPORT_REF OsclExecScheduler* OsclExecScheduler::Current () [static]

Get currently installed scheduler for calling thread, or NULL if no scheduler is installed.

7.133.1.2 OSCL_IMPORT_REF void OsclExecScheduler::RegisterForCallback ([OsclSchedulerObserver](#) * aCallback, [OsclAny](#) * aCallbackContext)

Register for a notification when non-blocking scheduler needs to run again.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

7.133.1.3 OSCL_IMPORT_REF void OsclExecScheduler::RunSchedulerNonBlocking (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)

Run PV scheduler in non-blocking mode. This call returns when the desired number of Run calls have been made, or when there are no more active objects that are ready to run.

Parameters:

aTargetCount: (input param) the maximum number of Run calls to make.

aReady: (output param) tells the number of active objects that are currently ready to run.

aDelayMsec: (output param) If no active objects are ready to run, but one or more active objects are waiting on timers, this parameter will tell the time interval from the current time until the first of the pending timer objects will be ready to run, in milliseconds.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

7.133.2 Friends And Related Function Documentation

7.133.2.1 friend class OsclScheduler [friend]

Reimplemented from [OsclExecSchedulerCommonBase](#).

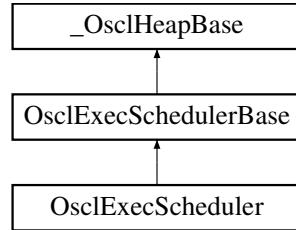
The documentation for this class was generated from the following file:

- [oscl_scheduler.h](#)

7.134 OsclExecSchedulerBase Class Reference

```
#include <oscl_scheduler_types.h>
```

Inheritance diagram for OsclExecSchedulerBase::



Friends

- class [OsclExecScheduler](#)
- class [OsclCoeActiveScheduler](#)
- class [PVActiveBase](#)

7.134.1 Detailed Description

OsclActiveSchedulerBase is the base for [OsclExecScheduler](#). The non-Symbian OsclActiveSchedulerBase class is functionally similar to a subset of Symbian CActiveScheduler class.

7.134.2 Friends And Related Function Documentation

7.134.2.1 friend class OsclCoeActiveScheduler [friend]

7.134.2.2 friend class OsclExecScheduler [friend]

7.134.2.3 friend class PVActiveBase [friend]

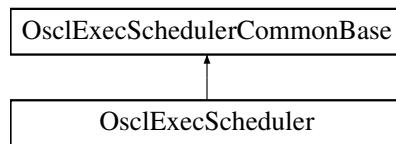
The documentation for this class was generated from the following file:

- [oscl_scheduler_types.h](#)

7.135 OsclExecSchedulerCommonBase Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for OsclExecSchedulerCommonBase::



Public Methods

- OSCL_IMPORT_REF void [StartScheduler](#) (OsclSemaphore *sem=NULL)
- OSCL_IMPORT_REF void [StopScheduler](#) ()
- OSCL_IMPORT_REF void [SuspendScheduler](#) ()
- OSCL_IMPORT_REF void [ResumeScheduler](#) ()
- OSCL_IMPORT_REF void [StartNativeScheduler](#) ()

Static Public Methods

- OSCL_IMPORT_REF [OsclNameString< PVSCHEDNAMELEN > *](#) [GetName](#) ()
- OSCL_IMPORT_REF uint32 [GetId](#) ()

Protected Types

- enum [TOtherExecStats](#) { [EOtherExecStats_WaitTime](#), [EOtherExecStats_QueueTime](#), [EOtherExecStats_NativeOS](#), [EOtherExecStats_ReleaseTime](#), [EOtherExecStats_Last](#) }

Protected Methods

- virtual ~[OsclExecSchedulerCommonBase](#) ()
- void [InstallScheduler](#) ()
- void [UninstallScheduler](#) ()
- void [Error](#) (int32 anError) const
- [OsclExecSchedulerCommonBase](#) ([Oscl_DefAlloc](#) *)
- virtual void [ConstructL](#) (const char *name, int)
- void [BeginScheduling](#) (bool blocking, bool native)
- void [EndScheduling](#) ()
- void [BlockingLoopL](#) ()
- bool [IsStarted](#) ()
- bool [IsInstalled](#) ()
- void [AddToExecTimerQ](#) ([PVActiveBase](#) *active, uint32)
- void [PendComplete](#) ([PVActiveBase](#) *, int32 aReason, [TPVThreadContext](#) aContext)
- void [RequestCanceled](#) ([PVActiveBase](#) *)
- [PVActiveBase](#) * [UpdateTimers](#) (uint32 &aDelay)
- [PVActiveBase](#) * [UpdateTimersMsec](#) (uint32 &aDelay)
- [PVActiveBase](#) * [WaitForReadyAO](#) ()

- void [CallRunExec \(PVActiveBase *\)](#)
- void [ConstructStatQ \(\)](#)
- void [BeginStats \(\)](#)
- void [EndStats \(\)](#)
- void [CleanupStatQ \(\)](#)
- PVActiveBase * [FindPVBase \(PVActiveBase *active, OsclDoubleList< PVActiveBase > &\)](#)
- void [CleanupExecQ \(\)](#)
- void [InitExecQ \(int\)](#)
- void [ResetLogPerf \(\)](#)
- void [IncLogPerf \(uint32\)](#)

Static Protected Methods

- OsclExecSchedulerCommonBase * [GetScheduler \(\)](#)
- OsclExecSchedulerCommonBase * [SetScheduler \(OsclExecSchedulerCommonBase *\)](#)
- void [ShowStats \(PVActiveStats *active\)](#)
- void [ShowSummaryStats \(PVActiveStats *active, PVLogger *, int64, int64 &, float &\)](#)

Protected Attributes

- bool [iBlockingMode](#)
- bool [iNativeMode](#)
- PVSchedulerStopper * [iStopper](#)
- OsclNoYieldMutex [iStopperCrit](#)
- PVThreadContext [iThreadContext](#)
- OsclNameString< PVSCHEDNAMELEN > [iName](#)
- bool [iDoStop](#)
- bool [iDoSuspend](#)
- bool [iSuspended](#)
- OsclSemaphore [iResumeSem](#)
- OsclErrorTrapImp * [iErrorTrapImp](#)
- OsclReadyQ [iReadyQ](#)
- OsclTimerQ [iExecTimerQ](#)
- uint32 [iNumAOAdded](#)
- OsclDoubleList< PVActiveStats > [iPVStatQ](#)
- PVActiveStats * [iOtherExecStats \[EOtherExecStats_Last\]](#)
- uint8 * [iTTotalTicksTemp](#)
- int64 [iGrandTotalTicks](#)
- float [iTTotalPercent](#)
- uint32 [iTime](#)
- int32 [iDelta](#)
- PVActiveStats * [iPVStats](#)
- PVLogger * [iLogger](#)
- PVLogger * [iDebugLogger](#)
- char * [iLogPerfIndentStr](#)
- int32 [iLogPerfIndentStrLen](#)
- uint32 [iLogPerfTotal](#)
- Oscl_DefAlloc * [iAlloc](#)
- OsclMemAllocator [iDefAlloc](#)

Static Protected Attributes

- const uint32 [iTTimeCompareThreshold](#)

Friends

- class [OsclScheduler](#)
- class [PVThreadContext](#)
- class [OsclCoeActiveScheduler](#)
- class [OsclTimerCompare](#)
- class [OsclReadyQ](#)
- class [OsclError](#)
- class [PVActiveStats](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [PVActiveBase](#)
- class [PVSchedulerStopper](#)
- class [OsclExecScheduler](#)

7.135.1 Member Enumeration Documentation

7.135.1.1 enum OsclExecSchedulerCommonBase::TOtherExecStats [protected]

Enumeration values:

- EOtherExecStats_WaitTime**
- EOtherExecStats_QueueTime**
- EOtherExecStats_NativeOS**
- EOtherExecStats_ReleaseTime**
- EOtherExecStats_Last**

7.135.2 Constructor & Destructor Documentation

- 7.135.2.1 **virtual OsclExecSchedulerCommonBase::~OsclExecSchedulerCommonBase ()**
[protected, virtual]
- 7.135.2.2 **OsclExecSchedulerCommonBase::OsclExecSchedulerCommonBase (Oscl_DefAlloc *)**
[protected]

7.135.3 Member Function Documentation

- 7.135.3.1 **void OsclExecSchedulerCommonBase::AddToExecTimerQ (PVActiveBase * *active*, uint32)** [protected]
- 7.135.3.2 **void OsclExecSchedulerCommonBase::BeginScheduling (bool *blocking*, bool *native*)**
[protected]
- 7.135.3.3 **void OsclExecSchedulerCommonBase::BeginStats ()** [protected]
- 7.135.3.4 **void OsclExecSchedulerCommonBase::BlockingLoopL ()** [protected]
- 7.135.3.5 **void OsclExecSchedulerCommonBase::CallRunExec (PVActiveBase *)** [protected]
- 7.135.3.6 **void OsclExecSchedulerCommonBase::CleanupExecQ ()** [protected]
- 7.135.3.7 **void OsclExecSchedulerCommonBase::CleanupStatQ ()** [protected]
- 7.135.3.8 **virtual void OsclExecSchedulerCommonBase::ConstructL (const char * *name*, int)**
[protected, virtual]
- 7.135.3.9 **void OsclExecSchedulerCommonBase::ConstructStatQ ()** [protected]
- 7.135.3.10 **void OsclExecSchedulerCommonBase::EndScheduling ()** [protected]
- 7.135.3.11 **void OsclExecSchedulerCommonBase::EndStats ()** [protected]
- 7.135.3.12 **void OsclExecSchedulerCommonBase::Error (int32 *anError*) const** [protected]
- 7.135.3.13 **PVActiveBase* OsclExecSchedulerCommonBase::FindPVBase (PVActiveBase * *active*, OsclDoubleList< PVActiveBase > &)** [protected]
- 7.135.3.14 **OSCL_IMPORT_REF uint32 OsclExecSchedulerCommonBase::GetId ()** [static]

Get numeric ID of current thread.

- 7.135.3.15 **OSCL_IMPORT_REF OsclNameString< PVSCHEDNAMELEN >* OsclExecSchedulerCommonBase::GetName ()** [static]

Get name of scheduler for current thread.

- 7.135.3.16** `OsclExecSchedulerCommonBase* OsclExecSchedulerCommonBase::GetScheduler ()` [static, protected]
- 7.135.3.17** `void OsclExecSchedulerCommonBase::IncLogPerf (uint32) [protected]`
- 7.135.3.18** `void OsclExecSchedulerCommonBase::InitExecQ (int) [protected]`
- 7.135.3.19** `void OsclExecSchedulerCommonBase::InstallScheduler () [protected]`
- 7.135.3.20** `bool OsclExecSchedulerCommonBase::IsInstalled () [inline, protected]`
- 7.135.3.21** `bool OsclExecSchedulerCommonBase::IsStarted () [protected]`
- 7.135.3.22** `void OsclExecSchedulerCommonBase::PendComplete (PVActiveBase *, int32 aReason, TPVThreadContext aContext) [protected]`
- 7.135.3.23** `void OsclExecSchedulerCommonBase::RequestCanceled (PVActiveBase *) [protected]`
- 7.135.3.24** `void OsclExecSchedulerCommonBase::ResetLogPerf () [protected]`
- 7.135.3.25** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::ResumeScheduler ()`

Resume scheduling immediately. This API only applies to a blocking loop scheduler.

- 7.135.3.26** `OsclExecSchedulerCommonBase* OsclExecSchedulerCommonBase::SetScheduler (OsclExecSchedulerCommonBase *) [static, protected]`
- 7.135.3.27** `void OsclExecSchedulerCommonBase::ShowStats (PVActiveStats * active) [static, protected]`
- 7.135.3.28** `void OsclExecSchedulerCommonBase::ShowSummaryStats (PVActiveStats * active, PVLogger *, int64, int64 &, float &) [static, protected]`
- 7.135.3.29** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::StartNativeScheduler ()`

Start the OS native scheduling loop. This is an alternative to the PV scheduling loop. To stop the native scheduler, use the StopScheduler API.

- 7.135.3.30** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::StartScheduler (OsclSemaphore * sem = NULL)`

Start scheduling. This call blocks until scheduler is stopped or an error occurs.

Parameters:

sem: optional startup semaphore. If provided, the scheduler will signal this semaphore when the startup has progressed to the point that it's safe to call StopScheduler or SuspendScheduler from another thread.

7.135.3.31 OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::StopScheduler ()

Stop scheduling. This API may be called from the scheduling thread or some other thread.

7.135.3.32 OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::SuspendScheduler ()

Suspend scheduling when the current Run is complete. This API only applies to a blocking loop scheduler.

7.135.3.33 void OsclExecSchedulerCommonBase::UninstallScheduler () [protected]**7.135.3.34 PVActiveBase* OsclExecSchedulerCommonBase::UpdateTimers (uint32 & aDelay) [protected]****7.135.3.35 PVActiveBase* OsclExecSchedulerCommonBase::UpdateTimersMsec (uint32 & aDelay) [protected]****7.135.3.36 PVActiveBase* OsclExecSchedulerCommonBase::WaitForReadyAO () [protected]****7.135.4 Friends And Related Function Documentation****7.135.4.1 friend class OsclActiveObject [friend]****7.135.4.2 friend class OsclCoeActiveScheduler [friend]****7.135.4.3 friend class OsclError [friend]****7.135.4.4 friend class OsclExecScheduler [friend]****7.135.4.5 friend class OsclReadyQ [friend]****7.135.4.6 friend class OsclScheduler [friend]**

Reimplemented in [OsclExecScheduler](#).

7.135.4.7 friend class OsclTimerCompare [friend]

7.135.4.8 friend class OsclTimerObject [friend]

7.135.4.9 friend class PVActiveBase [friend]

7.135.4.10 friend class PVActiveStats [friend]

7.135.4.11 friend class PVSchedulerStopper [friend]

7.135.4.12 friend class PVThreadContext [friend]

7.135.5 Field Documentation

7.135.5.1 **Oscl_DefAlloc*** OsclExecSchedulerCommonBase::iAlloc [protected]

7.135.5.2 bool OsclExecSchedulerCommonBase::iBlockingMode [protected]

7.135.5.3 **PVLogger*** OsclExecSchedulerCommonBase::iDebugLogger [protected]

7.135.5.4 **OsclMemAllocator** OsclExecSchedulerCommonBase::iDefAlloc [protected]

7.135.5.5 int32 OsclExecSchedulerCommonBase::iDelta [protected]

7.135.5.6 bool OsclExecSchedulerCommonBase::iDoStop [protected]

7.135.5.7 bool OsclExecSchedulerCommonBase::iDoSuspend [protected]

7.135.5.8 **OsclErrorTrapImp*** OsclExecSchedulerCommonBase::iErrorTrapImp
[protected]

7.135.5.9 **OsclTimerQ** OsclExecSchedulerCommonBase::iExecTimerQ [protected]

7.135.5.10 **int64** OsclExecSchedulerCommonBase::iGrandTotalTicks [protected]

7.135.5.11 **PVLogger*** OsclExecSchedulerCommonBase::iLogger [protected]

7.135.5.12 char* OsclExecSchedulerCommonBase::iLogPerfIndentStr [protected]

7.135.5.13 int32 OsclExecSchedulerCommonBase::iLogPerfIndentStrLen [protected]

7.135.5.14 uint32 OsclExecSchedulerCommonBase::iLogPerfTotal [protected]

7.135.5.15 **OsclNameString<PVSCHEDEXNAMELEN>** OsclExecSchedulerCommonBase::iName
[protected]

7.135.5.16 bool OsclExecSchedulerCommonBase::iNativeMode [protected]

7.135.5.17 uint32 OsclExecSchedulerCommonBase::iNumAOAdded [protected]

7.135.5.18 **PVActiveStats*** OsclExecSchedulerCommonBase::iOtherExecStats[EOtherExecStats_-
Last] [protected]

7.135.5.19 **OsclDoubleList<PVActiveStats>** OsclExecSchedulerCommonBase::iPVStatQ

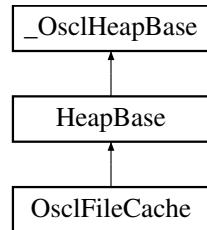
[protected]

- [oscl_scheduler.h](#)

7.136 OsclFileCache Class Reference

```
#include <oscl_file_cache.h>
```

Inheritance diagram for OsclFileCache::



Public Methods

- [OsclFileCache \(Oscl_File &aContainer\)](#)
- [~OsclFileCache \(\)](#)
- int32 [Open \(uint32 mode, uint32 cache_size\)](#)
- void [Close \(\)](#)
- uint32 [Read \(void *outputBuffer, uint32 size, uint32 numelements\)](#)
- uint32 [Write \(const void *inputBuffer, uint32 size, uint32 numelements\)](#)
- [TOsclFileOffset FileSize \(\)](#)
- int32 [Seek \(TOsclFileOffset offset, Oscl_File::seek_type origin\)](#)
- [TOsclFileOffset Tell \(\)](#)
- int32 [Flush \(\)](#)
- int32 [EndOfFile \(\)](#)
- OSCL_IMPORT_REF [OsclFileCacheBuffer * AddFixedCache \(const Oscl_File::OsclFixedCacheParam &\)](#)

Data Fields

- [OsclFileCacheBuffer _movableCache](#)
- [Oscl_Vector< OsclFileCacheBuffer, OsclMemAllocator > _fixedCaches](#)

Friends

- class [OsclFileCacheBuffer](#)

7.136.1 Constructor & Destructor Documentation

7.136.1.1 **OsclFileCache::OsclFileCache ([Oscl_File](#) & *aContainer*)**

7.136.1.2 **OsclFileCache::~OsclFileCache ()**

7.136.2 Member Function Documentation

7.136.2.1 **OSCL_IMPORT_REF [OsclFileCacheBuffer](#)* OsclFileCache::AddFixedCache (const [Oscl_File::OsclFixedCacheParam](#) &)**

7.136.2.2 **void OsclFileCache::Close ()**

7.136.2.3 **int32 OsclFileCache::EndOfFile () [inline]**

7.136.2.4 **[TOsclFileOffset](#) OsclFileCache::FileSize () [inline]**

7.136.2.5 **int32 OsclFileCache::Flush ()**

7.136.2.6 **int32 OsclFileCache::Open (uint32 *mode*, uint32 *cache_size*)**

7.136.2.7 **uint32 OsclFileCache::Read (void * *outputBuffer*, uint32 *size*, uint32 *numelements*)**

7.136.2.8 **int32 OsclFileCache::Seek ([TOsclFileOffset](#) *offset*, [Oscl_File::seek_type](#) *origin*)**

7.136.2.9 **[TOsclFileOffset](#) OsclFileCache::Tell () [inline]**

7.136.2.10 **uint32 OsclFileCache::Write (const void * *inputBuffer*, uint32 *size*, uint32 *numelements*)**

7.136.3 Friends And Related Function Documentation

7.136.3.1 **friend class OsclFileCacheBuffer [friend]**

7.136.4 Field Documentation

7.136.4.1 **[Oscl_Vector](#)<[OsclFileCacheBuffer](#), [OsclMemAllocator](#)> OsclFileCache::_fixedCaches**

7.136.4.2 **[OsclFileCacheBuffer](#) OsclFileCache::_movableCache**

The documentation for this class was generated from the following file:

- [oscl_file_cache.h](#)

7.137 OsclFileCacheBuffer Class Reference

```
#include <oscl_file_cache.h>
```

Public Methods

- [OsclFileCacheBuffer \(\)](#)
- int32 [SetPosition \(TOsclFileOffset pos\)](#)
- int32 [PrepRead \(\)](#)
- int32 [PrepWrite \(\)](#)
- int32 [WriteUpdatesToFile \(\)](#)
- int32 [FillFromFile \(uint32, uint32\)](#)
- bool [IsUpdated \(\)](#)
- bool [Contains \(TOsclFileOffset pos\)](#)
- bool [Preceeds \(TOsclFileOffset pos\)](#)

Data Fields

- [OsclFileCache * iContainer](#)
- bool [isFixed](#)
- uint32 [capacity](#)
- uint32 [usableSize](#)
- uint8 * [pBuffer](#)
- [TOsclFileOffset filePosition](#)
- uint32 [currentPos](#)
- uint32 [endPos](#)
- uint32 [updateStart](#)
- uint32 [updateEnd](#)

7.137.1 Constructor & Destructor Documentation

7.137.1.1 OsclFileCacheBuffer::OsclFileCacheBuffer () [inline]

7.137.2 Member Function Documentation

7.137.2.1 bool OsclFileCacheBuffer::Contains (TOsclFileOffset *pos*) [inline]

7.137.2.2 int32 OsclFileCacheBuffer::FillFromFile (uint32, uint32)

7.137.2.3 bool OsclFileCacheBuffer::IsUpdated () [inline]

7.137.2.4 bool OsclFileCacheBuffer::Preceeds (TOsclFileOffset *pos*) [inline]

7.137.2.5 int32 OsclFileCacheBuffer::PreRead ()

7.137.2.6 int32 OsclFileCacheBuffer::PrepWrite ()

7.137.2.7 int32 OsclFileCacheBuffer::SetPosition (TOsclFileOffset *pos*)

7.137.2.8 int32 OsclFileCacheBuffer::WriteUpdatesToFile ()

7.137.3 Field Documentation

7.137.3.1 uint32 OsclFileCacheBuffer::capacity

7.137.3.2 uint32 OsclFileCacheBuffer::currentPos

7.137.3.3 uint32 OsclFileCacheBuffer::endPos

7.137.3.4 TOsclFileOffset OsclFileCacheBuffer::filePosition

7.137.3.5 OsclFileCache* OsclFileCacheBuffer::iContainer

7.137.3.6 bool OsclFileCacheBuffer::isFixed

7.137.3.7 uint8* OsclFileCacheBuffer::pBuffer

7.137.3.8 uint32 OsclFileCacheBuffer::updateEnd

7.137.3.9 uint32 OsclFileCacheBuffer::updateStart

7.137.3.10 uint32 OsclFileCacheBuffer::usableSize

The documentation for this class was generated from the following file:

- [oscl_file_cache.h](#)

7.138 OsclFileHandle Class Reference

```
#include <oscl_file_handle.h>
```

Public Methods

- [OsclFileHandle \(TOsclFileHandle aHandle\)](#)
- [OsclFileHandle \(const OsclFileHandle &aHandle\)](#)
- [TOsclFileHandle Handle \(\) const](#)

Friends

- class [Oscl_File](#)

7.138.1 Detailed Description

OsclFileHandle is a container for a handle to a previously-opened file.

7.138.2 Constructor & Destructor Documentation

7.138.2.1 OsclFileHandle::OsclFileHandle (TOsclFileHandle *aHandle*) [inline]

7.138.2.2 OsclFileHandle::OsclFileHandle (const OsclFileHandle & *aHandle*) [inline]

7.138.3 Member Function Documentation

7.138.3.1 TOsclFileHandle OsclFileHandle::Handle () const [inline]

7.138.4 Friends And Related Function Documentation

7.138.4.1 friend class Oscl_File [friend]

The documentation for this class was generated from the following file:

- [oscl_file_handle.h](#)

7.139 OsclFileStats Class Reference

```
#include <oscl_file_stats.h>
```

Public Methods

- [OsclFileStats \(Oscl_File *c\)](#)
- void [Start \(uint32 &aTicks\)](#)
- void [End \(TOsclFileOp aOp, uint32 aStart, uint32 aParam=0, TOsclFileOffset aParam2=0\)](#)
- void [Log \(TOsclFileOp, PVLogger *, uint32\)](#)
- void [LogAll \(PVLogger *, uint32\)](#)

7.139.1 Constructor & Destructor Documentation

7.139.1.1 OsclFileStats::OsclFileStats ([Oscl_File](#) * *c*)

7.139.2 Member Function Documentation

7.139.2.1 void OsclFileStats::End ([TOsclFileOp](#) *aOp*, [uint32](#) *aStart*, [uint32](#) *aParam* = 0, [TOsclFileOffset](#) *aParam2* = 0)

7.139.2.2 void OsclFileStats::Log ([TOsclFileOp](#), [PVLogger](#) *, [uint32](#))

7.139.2.3 void OsclFileStats::LogAll ([PVLogger](#) *, [uint32](#))

7.139.2.4 void OsclFileStats::Start ([uint32](#) & *aTicks*)

The documentation for this class was generated from the following file:

- [oscl_file_stats.h](#)

7.140 OsclFileStatsItem Class Reference

```
#include <oscl_file_stats.h>
```

Data Fields

- uint32 [iOpCount](#)
- uint32 [iParam](#)
- [TOsclFileOffset](#) [iParam2](#)
- uint32 [iStartTick](#)
- uint32 [iTTotalTicks](#)

7.140.1 Field Documentation

7.140.1.1 uint32 OsclFileStatsItem::iOpCount

7.140.1.2 uint32 OsclFileStatsItem::iParam

7.140.1.3 TOsclFileOffset OsclFileStatsItem::iParam2

7.140.1.4 uint32 OsclFileStatsItem::iStartTick

7.140.1.5 uint32 OsclFileStatsItem::iTTotalTicks

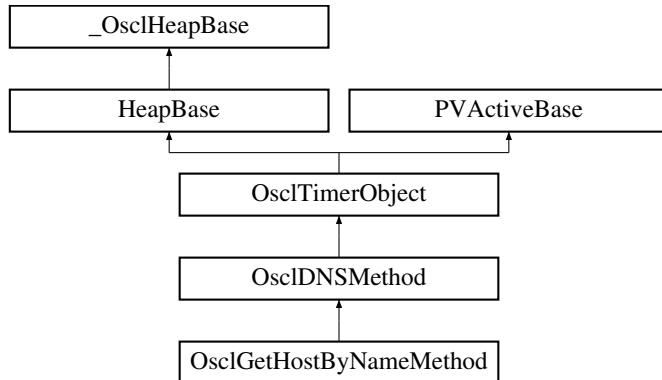
The documentation for this class was generated from the following file:

- [oscl_file_stats.h](#)

7.141 OsclGetHostByNameMethod Class Reference

```
#include <oscl_dns_gethostbyname.h>
```

Inheritance diagram for OsclGetHostByNameMethod::



Public Methods

- [~OsclGetHostByNameMethod \(\)](#)
- [TPVDNSEvent GetHostByName \(char *name, OsclNetworkAddress *addr, int32 aTimeout\)](#)

Static Public Methods

- [OsclGetHostByNameMethod * NewL \(Oscl_DefAlloc &a, OsclDNSI *aDNS, OsclDNSObserver *aObserver, uint32 aId\)](#)

7.141.1 Constructor & Destructor Documentation

7.141.1.1 OsclGetHostByNameMethod::~OsclGetHostByNameMethod ()

7.141.2 Member Function Documentation

7.141.2.1 [TPVDNSEvent OsclGetHostByNameMethod::GetHostByName \(char * name, OsclNetworkAddress * addr, int32 aTimeout\)](#)

7.141.2.2 [OsclGetHostByNameMethod* OsclGetHostByNameMethod::NewL \(Oscl_DefAlloc &a, OsclDNSI *aDNS, OsclDNSObserver *aObserver, uint32 aId\) \[static\]](#)

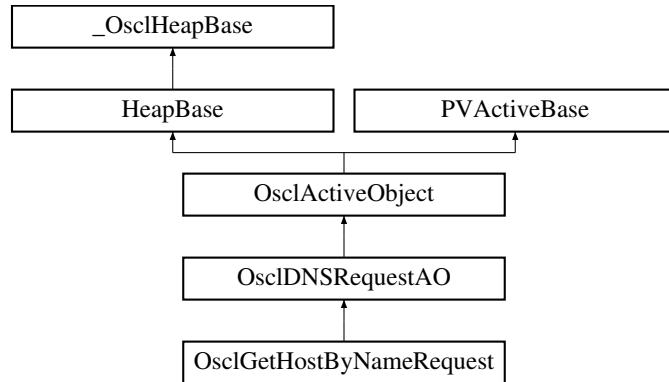
The documentation for this class was generated from the following file:

- [oscl_dns_gethostbyname.h](#)

7.142 OsclGetHostByNameRequest Class Reference

```
#include <oscl_dns_gethostbyname.h>
```

Inheritance diagram for OsclGetHostByNameRequest::



Friends

- class [OsclGetHostByNameMethod](#)

7.142.1 Friends And Related Function Documentation

7.142.1.1 friend class OsclGetHostByNameMethod [friend]

The documentation for this class was generated from the following file:

- [oscl_dns_gethostbyname.h](#)

7.143 OsclInit Class Reference

```
#include <oscl_init.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [Init](#) (int32 &aError, const [OsclSelect](#) *aSelect=NULL)
- OSCL_IMPORT_REF void [Cleanup](#) (int32 &aError, const [OsclSelect](#) *aSelect=NULL)

7.143.1 Detailed Description

Per-thread oscl initialization and cleanup.

7.143.2 Member Function Documentation

7.143.2.1 OSCL_IMPORT_REF void OsclInit::Cleanup (int32 & aError, const [OsclSelect](#) * aSelect = NULL) [static]

This routine cleans up the Oscl modules in the calling thread.

Parameters:

err: (output) error code of any leave that occurs in initialization.

config: (input param) optional set of initialization parameters. If null, then full initialization with default parameters will be performed. For proper cleanup, the parameters should match the ones used during the Init call.

7.143.2.2 OSCL_IMPORT_REF void OsclInit::Init (int32 & aError, const [OsclSelect](#) * aSelect = NULL) [static]

This routine initializes the Oscl modules in the calling thread.

Parameters:

err: (output) error code of any leave that occurs in initialization.

config: (input param) optional set of initialization parameters. If null, then full initialization with default parameters will be performed.

The documentation for this class was generated from the following file:

- [oscl_init.h](#)

7.144 OsclInteger64Transport Struct Reference

```
#include <oscl_int64_utils.h>
```

Data Fields

- uint32 [iHigh](#)
- uint32 [iLow](#)

7.144.1 Detailed Description

OsclInteger64Transport Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

7.144.2 Field Documentation

7.144.2.1 uint32 OsclInteger64Transport::iHigh

7.144.2.2 uint32 OsclInteger64Transport::iLow

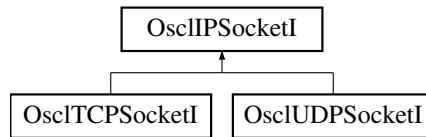
The documentation for this struct was generated from the following file:

- [oscl_int64_utils.h](#)

7.145 OsclIPSocketI Class Reference

```
#include <oscl_ip_socket.h>
```

Inheritance diagram for OsclIPSocketI::



Public Methods

- int32 [Bind \(OsclNetworkAddress &aAddress\)](#)
- int32 [Join \(OsclNetworkAddress &aAddress\)](#)
- int32 [SetRecvBufferSize \(uint32 size\)](#)
- virtual int32 [Close \(\)=0](#)
- virtual uint8 * [GetRecvData \(int32 *aLength\)=0](#)
- virtual uint8 * [GetSendData \(int32 *aLength\)=0](#)
- virtual ~[OsclIPSocketI \(\)](#)
- [OsclSocketServI * SocketServ \(\)](#)
- [Oscl_DefAlloc & Alloc \(\)](#)

Protected Methods

- [OsclIPSocketI \(Oscl_DefAlloc &a\)](#)
- void [ConstructL \(OsclSocketObserver *aObs, OsclSocketI *aSock, OsclSocketServI *aServ, uint32 aId\)](#)

Protected Attributes

- [Oscl_DefAlloc & iAlloc](#)
- [OsclNetworkAddress iAddress](#)
- uint32 [iId](#)
- [OsclSocketObserver * iObserver](#)
- [OsclSocketI * iSocket](#)
- [OsclSocketServI * iSocketServ](#)
- [PVLogger * iLogger](#)

Friends

- class [OsclSocketRequestAO](#)
- class [OsclSocketMethod](#)

7.145.1 Constructor & Destructor Documentation

7.145.1.1 `virtual OsclIPSocketI::~OsclIPSocketI ()` [inline, virtual]

7.145.1.2 `OsclIPSocketI::OsclIPSocketI (Oscl_DefAlloc & a)` [inline, protected]

7.145.2 Member Function Documentation

7.145.2.1 `Oscl_DefAlloc& OsclIPSocketI::Alloc ()` [inline]

7.145.2.2 `int32 OsclIPSocketI::Bind (OsclNetworkAddress & aAddress)`

7.145.2.3 `virtual int32 OsclIPSocketI::Close ()` [pure virtual]

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

7.145.2.4 `void OsclIPSocketI::ConstructL (OsclSocketObserver * aObs, OsclSocketI * aSock, OsclSocketServI * aServ, uint32 aId)` [protected]

7.145.2.5 `virtual uint8* OsclIPSocketI::GetRecvData (int32 * aLength)` [pure virtual]

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

7.145.2.6 `virtual uint8* OsclIPSocketI::GetSendData (int32 * aLength)` [pure virtual]

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

7.145.2.7 int32 OsclIPSocketI::Join ([OsclNetworkAddress](#) & *aAddress*)

7.145.2.8 int32 OsclIPSocketI::SetRecvBufferSize (uint32 *size*)

7.145.2.9 [OsclSocketServI](#)* OsclIPSocketI::SocketServ () [inline]

7.145.3 Friends And Related Function Documentation

7.145.3.1 friend class OsclSocketMethod [friend]

7.145.3.2 friend class OsclSocketRequestAO [friend]

7.145.4 Field Documentation

7.145.4.1 [OsclNetworkAddress](#) OsclIPSocketI::iAddress [protected]

7.145.4.2 [Oscl_DefAlloc](#)& OsclIPSocketI::iAlloc [protected]

7.145.4.3 uint32 OsclIPSocketI::iId [protected]

7.145.4.4 [PVLogger](#)* OsclIPSocketI::iLogger [protected]

7.145.4.5 [OsclSocketObserver](#)* OsclIPSocketI::iObserver [protected]

7.145.4.6 [OsclSocketI](#)* OsclIPSocketI::iSocket [protected]

7.145.4.7 [OsclSocketServI](#)* OsclIPSocketI::iSocketServ [protected]

The documentation for this class was generated from the following file:

- [oscl_ip_socket.h](#)

7.146 OsclJump Class Reference

```
#include <oscl_error_imp_jumps.h>
```

Public Methods

- void [Jump](#) (int a)
- jmp_buf * [Top](#) ()
- [~OsclJump](#) ()

Static Public Methods

- OSCL_IMPORT_REF void [StaticJump](#) (int a)

Friends

- class [OsclErrorTrapImp](#)

7.146.1 Constructor & Destructor Documentation

7.146.1.1 OsclJump::~OsclJump () [inline]

7.146.2 Member Function Documentation

7.146.2.1 void OsclJump::Jump (int a) [inline]

7.146.2.2 OSCL_IMPORT_REF void OsclJump::StaticJump (int a) [static]

7.146.2.3 jmp_buf* OsclJump::Top () [inline]

7.146.3 Friends And Related Function Documentation

7.146.3.1 friend class OsclErrorTrapImp [friend]

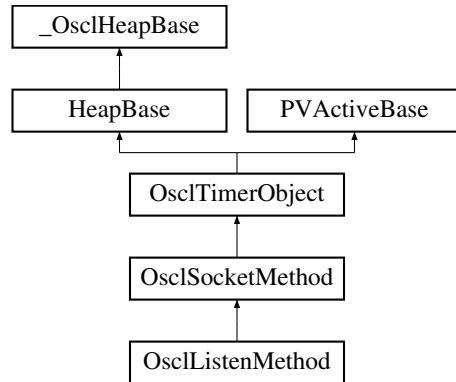
The documentation for this class was generated from the following file:

- [oscl_error_imp_jumps.h](#)

7.147 OsclListenMethod Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OsclListenMethod::



Public Methods

- [~OsclListenMethod \(\)](#)
- [TPVSocketEvent Listen \(uint32 qsize, int32 aTimeout\)](#)
- [OsclListenRequest * ListenRequest \(\)](#)

Static Public Methods

- [OsclListenMethod * NewL \(OsclIPSocketI &c\)](#)

7.147.1 Constructor & Destructor Documentation

7.147.1.1 OsclListenMethod::~OsclListenMethod ()

7.147.2 Member Function Documentation

7.147.2.1 TPVSocketEvent OsclListenMethod::Listen (uint32 qsize, int32 aTimeout)

7.147.2.2 OsclListenRequest* OsclListenMethod::ListenRequest () [inline]

7.147.2.3 OsclListenMethod* OsclListenMethod::NewL (OsclIPSocketI &c) [static]

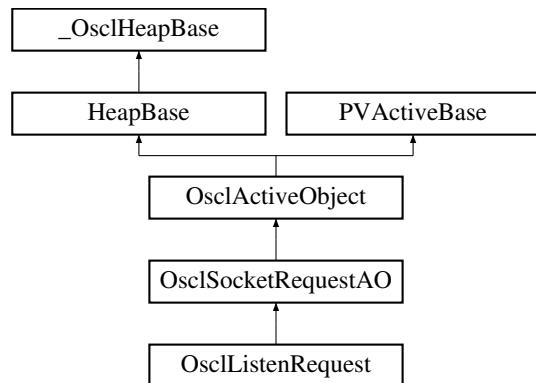
The documentation for this class was generated from the following file:

- [oscl_socket_listen.h](#)

7.148 OsclListenRequest Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OsclListenRequest::



Public Methods

- [OsclListenRequest \(OsclSocketMethod &c\)](#)
- void [Listen \(uint32 qsize\)](#)

7.148.1 Detailed Description

This is the AO that interacts with the socket server

7.148.2 Constructor & Destructor Documentation

7.148.2.1 OsclListenRequest::OsclListenRequest ([OsclSocketMethod & c](#)) [inline]

7.148.3 Member Function Documentation

7.148.3.1 void OsclListenRequest::Listen (uint32 *qsize*)

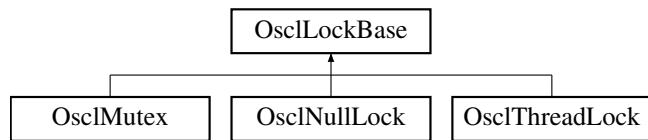
The documentation for this class was generated from the following file:

- [oscl_socket_listen.h](#)

7.149 OsclLockBase Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclLockBase::



Public Methods

- virtual void [Lock \(\)=0](#)
- virtual void [Unlock \(\)=0](#)
- virtual [~OsclLockBase \(\)](#)

7.149.1 Constructor & Destructor Documentation

7.149.1.1 virtual OsclLockBase::~OsclLockBase () [inline, virtual]

7.149.2 Member Function Documentation

7.149.2.1 virtual void OsclLockBase::Lock () [pure virtual]

Implemented in [OsclNullLock](#), [OsclMutex](#), and [OsclThreadLock](#).

7.149.2.2 virtual void OsclLockBase::Unlock () [pure virtual]

Implemented in [OsclNullLock](#), [OsclMutex](#), and [OsclThreadLock](#).

The documentation for this class was generated from the following file:

- [oscl_lock_base.h](#)

7.150 OsclMem Class Reference

```
#include <oscl_mem.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [Init \(\)](#)
- OSCL_IMPORT_REF void [Cleanup \(\)](#)

7.150.1 Member Function Documentation

7.150.1.1 OSCL_IMPORT_REF void OsclMem::Cleanup () [static]

Per-thread cleanup of Oscl Memory @exception: Leaves on error;

7.150.1.2 OSCL_IMPORT_REF void OsclMem::Init () [static]

Per-thread initialization of Oscl Memory

Parameters:

lock: A lock class for use with multi-threaded applications. The lock is needed in use cases where memory may be allocated in one thread and freed in another. In this case, there must be a single lock object, and its pointer must be passed to the [OsclMem::Init](#) call in each thread. If no lock is provided, the memory manager will not be thread-safe. @exception: Leaves on error

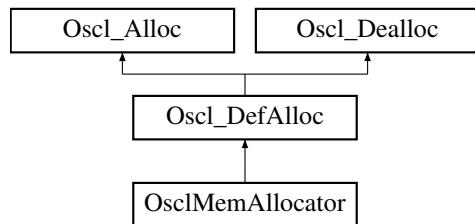
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

7.151 OsclMemAllocator Class Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemAllocator::



Public Methods

- [OsclAny * allocate \(const uint32 n\)](#)
- [OsclAny * allocate_fl \(const uint32 n, const char *file_name, const int line_num\)](#)
- void [deallocate \(OsclAny *p\)](#)

7.151.1 Detailed Description

A simple allocator class. Configurable as to whether this goes through the memory manager or not.

7.151.2 Member Function Documentation

7.151.2.1 [OsclAny* OsclMemAllocator::allocate \(const uint32 n\)](#) [inline, virtual]

This API throws an exception when malloc returns NULL. n must be greater than 0.

Returns:

pointer (or Leave with OsclErrNoMemory)

Implements [Oscl_DefAlloc](#).

7.151.2.2 [OsclAny* OsclMemAllocator::allocate_fl \(const uint32 n, const char *file_name, const int line_num\)](#) [inline, virtual]

Reimplemented from [Oscl_DefAlloc](#).

7.151.2.3 void [OsclMemAllocator::deallocate \(OsclAny *p\)](#) [inline, virtual]

Implements [Oscl_DefAlloc](#).

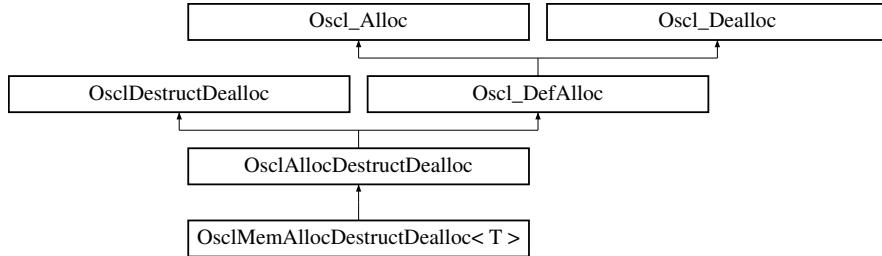
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

7.152 OsclMemAllocDestructDealloc< T > Class Template Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemAllocDestructDealloc< T >::



Public Methods

- [OsclAny * allocate_f1](#) (const uint32 size, const char *file_name, const int line_num)
- [OsclAny * allocate](#) (const uint32 size)
- void [deallocate](#) (OsclAny *p)
- void [destruct_and_dealloc](#) (OsclAny *p)

7.152.1 Detailed Description

`template<class T> class OsclMemAllocDestructDealloc< T >`

An [OsclAllocDestructDealloc](#) class that uses [OsclMemAllocator](#).

7.152.2 Member Function Documentation

7.152.2.1 `template<class T> OsclAny* OsclMemAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]`

Implements [Oscl_DefAlloc](#).

7.152.2.2 `template<class T> OsclAny* OsclMemAllocDestructDealloc< T >::allocate_f1 (const uint32 size, const char *file_name, const int line_num) [inline, virtual]`

Reimplemented from [Oscl_DefAlloc](#).

7.152.2.3 `template<class T> void OsclMemAllocDestructDealloc< T >::deallocate (OsclAny * p) [inline, virtual]`

Implements [Oscl_DefAlloc](#).

**7.152.2.4 template<class T> void OsclMemAllocDestructDealloc< T >::destruct_and_dealloc
(OsclAny * p) [inline, virtual]**

Implements [OsclDestructDealloc](#).

The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

7.153 OsclMemAudit Class Reference

```
#include <oscl_mem_audit.h>
```

Public Methods

- [OsclMemAudit \(\)](#)
- [~OsclMemAudit \(\)](#)
- [void * MM_allocate \(const OsclMemStatsNode *statsNode, uint32 sizeIn, const char *pFileName, uint32 lineNumber, bool allocNodeTracking=false\)](#)
- [bool MM_deallocate \(void *pMemBlockIn\)](#)
- [MM_Stats_t * MM_GetStats \(const char *const tagIn\)](#)
- [uint32 MM_GetStatsInDepth \(const char *tagIn, MM_Stats_CB *array_ptr, uint32 max_nodes\)](#)
- [uint32 MM_GetTreeNodes \(const char *tagIn\)](#)
- [bool MM_AddTag \(const char *tagIn\)](#)
- [const OsclMemStatsNode * MM_GetTagName \(const char *tagIn\)](#)
- [const OsclMemStatsNode * MM_GetExistingTag \(const char *tagIn\)](#)
- [const OsclMemStatsNode * MM_GetRootNode \(\)](#)
- [uint32 MM_GetAllocNodeInfo \(MM_AllocQueryInfo *output_array, uint32 max_array_size, uint32 offset\)](#)
- [MM_AllocQueryInfo * MM_CreateAllocNodeInfo \(uint32 max_array_size\)](#)
- [void MM_ReleaseAllocNodeInfo \(MM_AllocQueryInfo *info\)](#)
- [bool MM_Validate \(const void *ptrIn\)](#)
- [uint32 MM_GetAllocNo \(void\)](#)
- [void MM_GetOverheadStats \(MM_AuditOverheadStats &stats\)](#)
- [uint32 MM_GetNumAllocNodes \(\)](#)
- [uint32 MM_GetMode \(void\)](#)
- [uint8 MM_GetPrefillPattern \(void\)](#)
- [uint32 MM_GetPostfillPattern \(void\)](#)
- [void MM_SetMode \(uint32 inMode\)](#)
- [void MM_SetPrefillPattern \(uint8 pattern\)](#)
- [void MM_SetPostfillPattern \(uint8 pattern\)](#)
- [void MM_SetTagLevel \(uint32 level\)](#)
- [bool MM_SetFailurePoint \(const char *tagIn, uint32 alloc_number\)](#)
- [void MM_UnsetFailurePoint \(const char *tagIn\)](#)
- [int32 MM_GetRefCount \(\)](#)
- [OsclLockBase * GetLock \(\)](#)

Friends

- class [OsclMemGlobalAuditObject](#)

7.153.1 Constructor & Destructor Documentation

7.153.1.1 OsclMemAudit::OsclMemAudit () [inline]

Constructor, create the root node in statistics table

7.153.1.2 OsclMemAudit::~OsclMemAudit () [inline]

A destructor, remove all the nodes in allocation andstatistics table

7.153.2 Member Function Documentation**7.153.2.1 OsclLockBase* OsclMemAudit::GetLock () [inline]**

API to obtain mem lock ptr

7.153.2.2 bool OsclMemAudit::MM_AddTag (const char * tagIn) [inline]

API to add a node and zero out its counters; Note that this tag should be re-used

Parameters:

tagIn input tag

Returns:

true if operation succeeds;

7.153.2.3 void* OsclMemAudit::MM_allocate (const OsclMemStatsNode * statsNode, uint32 sizeIn, const char * pFileName, uint32 lineNumber, bool allocNodeTracking = false) [inline]

The following are APIs t __nothrow_ / const __nothrow_

Returns:

the memory pointer if operation succeeds.

7.153.2.4 MM_AllocQueryInfo* OsclMemAudit::MM_CreateAllocNodeInfo (uint32 max_array_size) [inline]**7.153.2.5 bool OsclMemAudit::MM_deallocate (void * pMemBlockIn) [inline]****Returns:**

true if operation succeeds;

7.153.2.6 uint32 OsclMemAudit::MM_GetAllocNo (void) [inline]

API to get the current allocation number

Returns:

the current allocation number

7.153.2.7 `uint32 OsclMemAudit::MM_GetAllocNodeInfo (MM_AllocQueryInfo * output_array, uint32 max_array_size, uint32 offset) [inline]`

API to query the list of alloc nodes. It copies the information into the provided output array.

Parameters:

output_array the array where the data will be written

max_array_size the max number of output array elements

offset the offset into the alloc node list from which the data should begin.

Returns:

the number of valid nodes in the output array

7.153.2.8 `const OsclMemStatsNode* OsclMemAudit::MM_GetExistingTag (const char * tagIn) [inline]`

API to add a node and zero out its counters; Note that this tag should be re-used

Parameters:

tagIn input tag

Returns:

true if operation succeeds;

7.153.2.9 `uint32 OsclMemAudit::MM_GetMode (void) [inline]`

API to get the operating mode of the mm_audit class.

7.153.2.10 `uint32 OsclMemAudit::MM_GetNumAllocNodes () [inline]`

API to get the number of allocation nodes (records) for allocations that are being tracked individually.

7.153.2.11 `void OsclMemAudit::MM_GetOverheadStats (MM_AuditOverheadStats & stats) [inline]`

API to get the overhead statistics for the memory used by the mm_audit class.

7.153.2.12 `uint32 OsclMemAudit::MM_GetPostfillPattern (void) [inline]`

API to get the postfill pattern. The pattern is used to fill the memory before freeing it.

7.153.2.13 `uint8 OsclMemAudit::MM_GetPrefillPattern (void) [inline]`

API to get the prefill pattern. The pattern is used to fill the memory before returning it to the caller.

7.153.2.14 int32 OsclMemAudit::MM_GetRefCount () [inline]

7.153.2.15 const OsclMemStatsNode* OsclMemAudit::MM_GetRootNode () [inline]

7.153.2.16 MM_Stats_t* OsclMemAudit::MM_GetStats (const char *const tagIn) [inline]

API to get memory statistics through context string(tag)

Returns:

statistics pointer if operation succeeds

7.153.2.17 uint32 OsclMemAudit::MM_GetStatsInDepth (const char * tagIn, MM_Stats_CB * array_ptr, uint32 max_nodes) [inline]

API to get memory statistics in detail through context string(tag) including its subtree

Returns:

statistics pointer array and actual number of nodes if operation succeeds

7.153.2.18 const OsclMemStatsNode* OsclMemAudit::MM_GetTagName (const char * tagIn) [inline]

API to add a node and zero out its counters; Note that this tag should be re-used

Parameters:

tagIn input tag

Returns:

pointer to [OsclMemStatsNode](#) which should be passed to MM_allocate

7.153.2.19 uint32 OsclMemAudit::MM_GetTreeNodes (const char * tagIn) [inline]

API to get the number of tree nodes including the tag node and its subtree

Parameters:

tagIn input tag

Returns:

the number of tree nodes ; 0 means no tag node

7.153.2.20 void OsclMemAudit::MM_ReleaseAllocNodeInfo (MM_AllocQueryInfo * info) [inline]

7.153.2.21 bool OsclMemAudit::MM_SetFailurePoint (const char * tagIn, uint32 alloc_number) [inline]

API to insert allocation failure deterministically according to allocation number associated with tag

Parameters:

tagIn input tag
alloc_number allocation number associated with tag

Returns:

true if operation succeeds;

7.153.2.22 void OsclMemAudit::MM_SetMode (uint32 *inMode*) [inline]

API to set the operating mode of the mm_audit class.

7.153.2.23 void OsclMemAudit::MM_SetPostfillPattern (uint8 *pattern*) [inline]

API to set the postfill pattern.

7.153.2.24 void OsclMemAudit::MM_SetPrefillPattern (uint8 *pattern*) [inline]

API to set the prefill pattern.

7.153.2.25 void OsclMemAudit::MM_SetTagLevel (uint32 *level*) [inline]

API to set the maximum tag level,i.e. tag level for a.b.c.d = 4

Parameters:

level input tag level to be set

7.153.2.26 void OsclMemAudit::MM_UnsetFailurePoint (const char * *tagIn*) [inline]

API to cancel the allocation failure point associated with tag

Parameters:

tagIn input tag

7.153.2.27 bool OsclMemAudit::MM_Validate (const void * *ptrIn*) [inline]

API to check the input pointer is a valid pointer to a chunk of memory

Parameters:

ptrIn input pointer to be validated

Returns:

true if operation succeeds;

7.153.3 Friends And Related Function Documentation

7.153.3.1 friend class OsclMemGlobalAuditObject [friend]

The documentation for this class was generated from the following file:

- [oscl_mem_audit.h](#)

7.154 OSCLMemAutoPtr< T, _Allocator > Class Template Reference

The oscl_auto_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the oscl_auto_ptr expires, its destructor uses delete to free the memory.

```
#include <oscl_mem_auto_ptr.h>
```

Public Methods

- **OSCLMemAutoPtr** (T *inPtr=0)
Default constructor Initializes the pointer and takes ownership.
- **OSCLMemAutoPtr** (const OSCLMemAutoPtr< T > &_Y)
Copy constructor.
- **OSCLMemAutoPtr< T, _Allocator > & operator=** (const OSCLMemAutoPtr< T, _Allocator > &_Y)
Assignment operator from an another oscl_auto_ptr.
- **~OSCLMemAutoPtr** ()
Destructor.
- **T & operator *** () const
The indirection operator () accesses a value indirectly, through a pointer.*
- **T * operator ->** () const
The indirection operator (->) accesses a value indirectly, through a pointer.
- **void takeOwnership** (T *ptr)
The takeOwnership function assigns the value with ownership.
- **void allocate** (**oscl_memsize_t** size)
- **void setWithoutOwnership** (T *ptr)
The takeOwnership function assigns the value with ownership.
- **T * get** () const
get() method returns the pointer, currently owned by the class.
- **T * release** () const
release() method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

Static Public Methods

- **void deallocate** (T *ptr)

Data Fields

- bool [_Ownership](#)

7.154.1 Detailed Description

```
template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> class OSCLMemAuto-  
Ptr< T, _Allocator >
```

The oscl_auto_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the oscl_auto_ptr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an oscl_auto_ptr object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The oscl_auto_ptr is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

7.154.2 Constructor & Destructor Documentation

```
7.154.2.1 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::OSCLMemAutoPtr (T * inPtr = 0) [inline,  
explicit]
```

Default constructor Initializes the pointer and takes ownership.

```
7.154.2.2 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::OSCLMemAutoPtr (const OSCLMemAutoPtr<  
T > & _Y) [inline]
```

Copy constructor.

Initializes the pointer and takes ownership from another oscl_auto_ptr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

```
7.154.2.3 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::~OSCLMemAutoPtr () [inline]
```

Destructor.

The pointer is deleted in case this class still has ownership

7.154.3 Member Function Documentation

7.154.3.1 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, _Allocator >::allocate (oscl_memsize_t size) [inline]

7.154.3.2 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, _Allocator >::deallocate (T *ptr) [inline, static]

7.154.3.3 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> T* OSCLMemAutoPtr< T, _Allocator >::get () const [inline]

[get\(\)](#) method returns the pointer, currently owned by the class.

7.154.3.4 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> T& OSCLMemAutoPtr< T, _Allocator >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OSCLMemAutoPtr can be used like the regular pointer that it was initialized with.

7.154.3.5 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> T* OSCLMemAutoPtr< T, _Allocator >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OSCLMemAutoPtr can be used like the regular pointer that it was initialized with.

7.154.3.6 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> OSCLMemAutoPtr< T, _Allocator >& OSCLMemAutoPtr< T, _Allocator >::operator=(const OSCLMemAutoPtr< T, _Allocator > & _Y) [inline]

Assignment operator from an another oscl_auto_ptr.

Parameters:

_Y The value parameter should be another oscl_auto_ptr

Returns:

Returns a reference to this oscl_auto_ptr instance with pointer initialized.

Precondition:

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the oscl_auto_ptr given as the input parameter. The ownership of the pointer is transferred.

7.154.3.7 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> T* OSCLMemAutoPtr< T, _Allocator >::release () const [inline]

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

**7.154.3.8 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> void
OSCLMemAutoPtr< T, _Allocator >::setWithoutOwnership (T *ptr) [inline]**

The takeOwnership function assigns the value with ownership.

**7.154.3.9 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> void
OSCLMemAutoPtr< T, _Allocator >::takeOwnership (T *ptr) [inline]**

The takeOwnership function assigns the value with ownership.

7.154.4 Field Documentation

**7.154.4.1 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> bool
OSCLMemAutoPtr< T, _Allocator >::_Ownership**

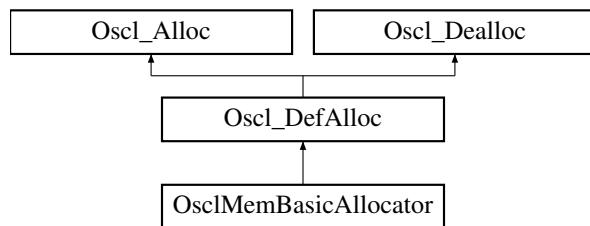
The documentation for this class was generated from the following file:

- [oscl_mem_auto_ptr.h](#)

7.155 OsclMemBasicAllocator Class Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemBasicAllocator::



Public Methods

- [OsclAny * allocate \(const uint32 n\)](#)
- [void deallocate \(OsclAny *p\)](#)

7.155.1 Detailed Description

A simple allocator class that does not use the memory management.

Note: this allocator is for internal use by Oscl only. Higher level code should use [OsclMemAllocator](#).

7.155.2 Member Function Documentation

7.155.2.1 [OsclAny* OsclMemBasicAllocator::allocate \(const uint32 n\) \[inline, virtual\]](#)

This API throws an exception when malloc returns NULL. n must be greater than 0.

Returns:

pointer (or Leave with OsclErrNoMemory)

Implements [Oscl_DefAlloc](#).

7.155.2.2 [void OsclMemBasicAllocator::deallocate \(OsclAny *p\) \[inline, virtual\]](#)

Implements [Oscl_DefAlloc](#).

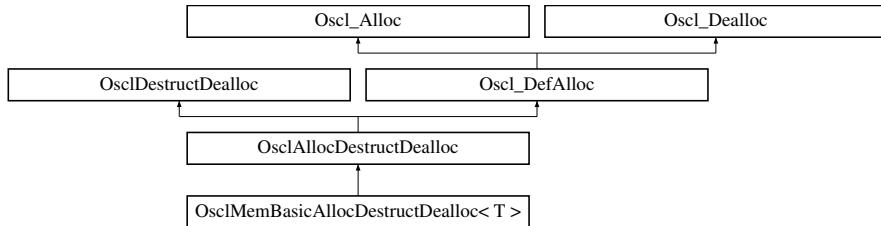
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

7.156 OsclMemBasicAllocDestructDealloc< T > Class Template Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemBasicAllocDestructDealloc< T >::



Public Methods

- [OsclAny * allocate \(const uint32 size\)](#)
- [void deallocate \(OsclAny *p\)](#)
- [void destruct_and_dealloc \(OsclAny *p\)](#)

7.156.1 Detailed Description

template<class T> class OsclMemBasicAllocDestructDealloc< T >

An [OsclAllocDestructDealloc](#) class that uses [OsclMemBasicAllocator](#).

7.156.2 Member Function Documentation

7.156.2.1 template<class T> [OsclAny*](#) OsclMemBasicAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]

Implements [Oscl_DefAlloc](#).

7.156.2.2 template<class T> void OsclMemBasicAllocDestructDealloc< T >::deallocate ([OsclAny](#) * p) [inline, virtual]

Implements [Oscl_DefAlloc](#).

7.156.2.3 template<class T> void OsclMemBasicAllocDestructDealloc< T >::destruct_and_dealloc ([OsclAny](#) * p) [inline, virtual]

Implements [OsclDestructDealloc](#).

The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

7.157 OsclMemGlobalAuditObject Class Reference

```
#include <oscl_mem.h>
```

Public Types

- `typedef OsclMemAudit audit_type`

Static Public Methods

- `OSCL_IMPORT_REF audit_type * getGlobalMemAuditObject ()`

Friends

- class `OsclMem`

7.157.1 Member Typedef Documentation

7.157.1.1 `typedef OsclMemAudit OsclMemGlobalAuditObject::audit_type`

7.157.2 Member Function Documentation

7.157.2.1 `OSCL_IMPORT_REF audit_type* OsclMemGlobalAuditObject::getGlobalMemAuditObject () [static]`

returns the global audit object. For use in macros only– not a public API.

7.157.3 Friends And Related Function Documentation

7.157.3.1 `friend class OsclMem [friend]`

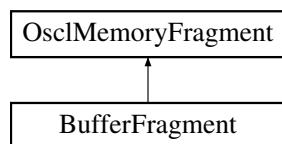
The documentation for this class was generated from the following file:

- `oscl_mem.h`

7.158 OsclMemoryFragment Struct Reference

```
#include <oscl_types.h>
```

Inheritance diagram for OsclMemoryFragment::



Data Fields

- `uint32 len`
- `void * ptr`

7.158.1 Field Documentation

7.158.1.1 `uint32 OsclMemoryFragment::len`

7.158.1.2 `void* OsclMemoryFragment::ptr`

The documentation for this struct was generated from the following file:

- [oscl_types.h](#)

7.159 OsclMemPoolAllocator Class Reference

```
#include <oscl_mempool_allocator.h>
```

Public Methods

- [OsclMemPoolAllocator \(Oscl_DefAlloc *gen_alloc=NULL\)](#)
- [virtual ~OsclMemPoolAllocator \(\)](#)
- [OsclAny * CreateMemPool \(const uint32 aNumChunk=2, const uint32 aChunkSize=4\)](#)
- [void DestroyMemPool \(\)](#)
- [uint oscl_mem_aligned_size \(uint size\)](#)

7.159.1 Constructor & Destructor Documentation

7.159.1.1 OsclMemPoolAllocator::OsclMemPoolAllocator ([Oscl_DefAlloc * gen_alloc = NULL](#))

7.159.1.2 virtual OsclMemPoolAllocator::~OsclMemPoolAllocator () [virtual]

7.159.2 Member Function Documentation

7.159.2.1 [OsclAny* OsclMemPoolAllocator::CreateMemPool \(const uint32 aNumChunk = 2, const uint32 aChunkSize = 4\)](#)

7.159.2.2 void OsclMemPoolAllocator::DestroyMemPool ()

7.159.2.3 [uint OsclMemPoolAllocator::oscl_mem_aligned_size \(uint size\)](#)

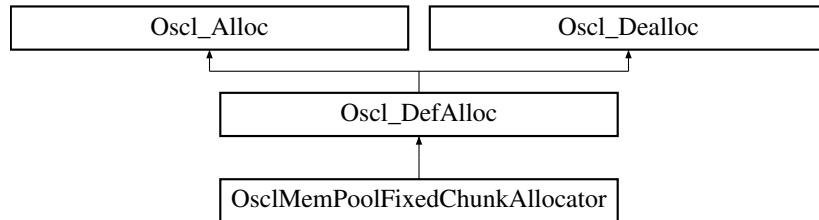
The documentation for this class was generated from the following file:

- [oscl_mempool_allocator.h](#)

7.160 OsclMemPoolFixedChunkAllocator Class Reference

```
#include <oscl_mem_mempool.h>
```

Inheritance diagram for OsclMemPoolFixedChunkAllocator::



Public Methods

- OSCL_IMPORT_REF OsclMemPoolFixedChunkAllocator (const uint32 numchunk=1, const uint32 chunksze=0, Oscl_DefAlloc *gen_alloc=NULL, const uint32 chunkalignment=0)
- virtual OSCL_IMPORT_REF void enablenullpointerreturn ()
- virtual OSCL_IMPORT_REF ~OsclMemPoolFixedChunkAllocator ()
- virtual OSCL_IMPORT_REF OsclAny * allocate (const uint32 n)
- virtual OSCL_IMPORT_REF void deallocate (OsclAny *p)
- virtual OSCL_IMPORT_REF void notifyfreechunkavailable (OsclMemPoolFixedChunkAllocatorObserver &obs, OsclAny *aContextData=NULL)
- virtual OSCL_IMPORT_REF void CancelFreeChunkAvailableCallback ()
- OSCL_IMPORT_REF void addRef ()
- OSCL_IMPORT_REF void removeRef ()

Protected Methods

- virtual OSCL_IMPORT_REF void createmempool ()
- virtual OSCL_IMPORT_REF void destroymempool ()

Protected Attributes

- uint32 iNumChunk
- uint32 iChunkSize
- uint32 iChunkSizeMemAligned
- uint32 iChunkAlignment
- Oscl_DefAlloc * iMemPoolAllocator
- OsclAny * iMemPool
- OsclAny * iMemPoolAligned
- Oscl_Vector< OsclAny *, OsclMemAllocator > iFreeMemChunkList
- bool iCheckNextAvailableFreeChunk
- OsclMemPoolFixedChunkAllocatorObserver * iObserver
- OsclAny * iNextAvailableContextData
- int32 iRefCount
- bool iEnableNullPtrReturn

7.160.1 Constructor & Destructor Documentation

7.160.1.1 OSCL_IMPORT_REF OsclMemPoolFixedChunkAllocator::OsclMemPoolFixedChunkAllocator (const uint32 numchunk = 1, const uint32 chunkszie = 0, Oscl_DefAlloc * gen_alloc = NULL, const uint32 chunkalignment = 0)

This API throws an exception when the memory allocation for pool fails If numchunk and chunkszie parameters are not set, memory pool of 1 chunk will be created in the first call to allocate. The chunk size will be set to the n passed in for [allocate\(\)](#). If numchunk parameter is set to 0, the memory pool will use 1 for numchunk. If chunkalignment is set to 0, memory pool will use default allocator alignment (8-byte) If chunkalignment is > 0, memory pool will align all buffers in the mempool to the specified alignment. Alignment should be a power of 2

Returns:

void

7.160.1.2 virtual OSCL_IMPORT_REF OsclMemPoolFixedChunkAllocator::~OsclMemPoolFixedChunkAllocator () [virtual]

The destructor for the memory pool

7.160.2 Member Function Documentation

7.160.2.1 OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::addRef ()

Increments the reference count for this memory pool allocator

Returns:

void

7.160.2.2 virtual OSCL_IMPORT_REF OsclAny* OsclMemPoolFixedChunkAllocator::allocate (const uint32 n) [virtual]

This API throws an exception when n is greater than the fixed chunk size or there are no free chunk available in the pool, if "enablenullpointerreturn" has not been called. If the memory pool hasn't been created yet, the pool will be created with chunk size equal to n so n must be greater than 0. Exception will be thrown if memory allocation for the memory pool fails.

Returns:

pointer to available chunk from memory pool

Implements [Oscl_DefAlloc](#).

7.160.2.3 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::CancelFreeChunkAvailableCallback () [virtual]

This API will cancel any past callback requests..

Returns:

void

7.160.2.4 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::createmempool() [protected, virtual]

**7.160.2.5 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::deallocate(
OsclAny *p)** [virtual]

This API throws an exception when the pointer p passed in is not part of the memory pool. Exception will be thrown if the memory pool is not set up yet.

Returns:

void

Implements [Oscl_DefAlloc](#).

7.160.2.6 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::destroymempool() [protected, virtual]

7.160.2.7 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::enablenullpointerreturn() [virtual]

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL_LEAVE\(OsclErrNoResources\)](#)" allocate API will return NULL.

Returns:

void

**7.160.2.8 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::notifyfreechunkavailable(
OsclMemPoolFixedChunkAllocatorObserver &obs, OsclAny *aContextData = NULL)** [virtual]

This API will set the flag to send a callback via specified observer object when the next memory chunk is deallocated by [deallocate\(\)](#) call..

Returns:

void

7.160.2.9 OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::removeRef()

Decrements the reference count for this memory pool allocator When the reference count goes to 0, this instance of the memory pool object is deleted

Returns:

void

7.160.3 Field Documentation

- 7.160.3.1 **bool OsclMemPoolFixedChunkAllocator::iCheckNextAvailableFreeChunk** [protected]
- 7.160.3.2 **uint32 OsclMemPoolFixedChunkAllocator::iChunkAlignment** [protected]
- 7.160.3.3 **uint32 OsclMemPoolFixedChunkAllocator::iChunkSize** [protected]
- 7.160.3.4 **uint32 OsclMemPoolFixedChunkAllocator::iChunkSizeMemAligned** [protected]
- 7.160.3.5 **bool OsclMemPoolFixedChunkAllocator::iEnableNullPtrReturn** [protected]
- 7.160.3.6 **Oscl_Vector<OsclAny*, OsclMemAllocator> OsclMemPoolFixedChunkAllocator::iFreeMemChunkList** [protected]
- 7.160.3.7 **OsclAny* OsclMemPoolFixedChunkAllocator::iMemPool** [protected]
- 7.160.3.8 **OsclAny* OsclMemPoolFixedChunkAllocator::iMemPoolAligned** [protected]
- 7.160.3.9 **Oscl_DefAlloc* OsclMemPoolFixedChunkAllocator::iMemPoolAllocator** [protected]
- 7.160.3.10 **OsclAny* OsclMemPoolFixedChunkAllocator::iNextAvailableContextData** [protected]
- 7.160.3.11 **uint32 OsclMemPoolFixedChunkAllocator::iNumChunk** [protected]
- 7.160.3.12 **OsclMemPoolFixedChunkAllocatorObserver* OsclMemPoolFixedChunkAllocator::iObserver** [protected]
- 7.160.3.13 **int32 OsclMemPoolFixedChunkAllocator::iRefCount** [protected]

The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

7.161 OsclMemPoolFixedChunkAllocatorObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

Public Methods

- virtual void [freechunkavailable \(OsclAny *aContextData\)=0](#)
- virtual [~OsclMemPoolFixedChunkAllocatorObserver \(\)](#)

7.161.1 Constructor & Destructor Documentation

7.161.1.1 virtual [OsclMemPoolFixedChunkAllocatorObserver::~OsclMemPoolFixedChunkAllocatorObserver \(\) \[inline, virtual\]](#)

7.161.2 Member Function Documentation

7.161.2.1 virtual void [OsclMemPoolFixedChunkAllocatorObserver::freechunkavailable \(OsclAny * aContextData\) \[pure virtual\]](#)

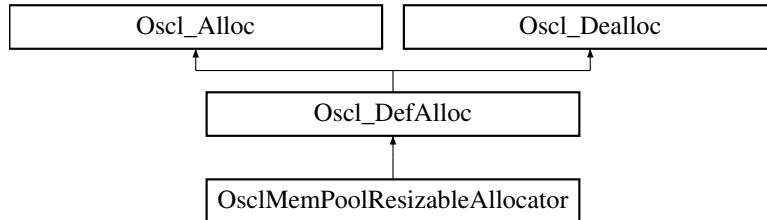
The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

7.162 OsclMemPoolResizableAllocator Class Reference

```
#include <oscl_mem_mempool.h>
```

Inheritance diagram for OsclMemPoolResizableAllocator::



Public Methods

- OSCL_IMPORT_REF [OsclMemPoolResizableAllocator](#) (uint32 aMemPoolBufferSize, uint32 aMemPoolBufferNumLimit=0, uint32 aExpectedNumBlocksPerBuffer=0, [Oscl_DefAlloc](#) *gen_alloc=NULL)
- virtual OSCL_IMPORT_REF void [enablenullpointerreturn](#) ()
- virtual OSCL_IMPORT_REF [OsclAny](#) * [allocate](#) (const uint32 aNumBytes)
- virtual OSCL_IMPORT_REF void [deallocate](#) ([OsclAny](#) *aPtr)
- virtual OSCL_IMPORT_REF bool [trim](#) ([OsclAny](#) *aPtr, uint32 aBytesToFree)
- OSCL_IMPORT_REF uint32 [getBufferSize](#) () const
- virtual OSCL_IMPORT_REF uint32 [getAllocatedSize](#) () const
- virtual OSCL_IMPORT_REF uint32 [getAvailableSize](#) () const
- virtual OSCL_IMPORT_REF uint32 [getLargestContiguousFreeBlockSize](#) () const
- virtual OSCL_IMPORT_REF bool [setMaxSzForNewMemPoolBuffer](#) (uint32 aMaxNewMemPoolBufferSz)
- virtual OSCL_IMPORT_REF void [notifyfreeblockavailable](#) ([OsclMemPoolResizableAllocatorObserver](#) &aObserver, uint32 aRequestedSize=0, [OsclAny](#) *aContextData=NULL)
- virtual OSCL_IMPORT_REF void [CancelFreeChunkAvailableCallback](#) ()
- virtual OSCL_IMPORT_REF void [notifyfreememoryavailable](#) ([OsclMemPoolResizableAllocatorMemoryObserver](#) &aObserver, uint32 aRequestedSize=0, [OsclAny](#) *aContextData=NULL)
- OSCL_IMPORT_REF void [CancelFreeMemoryAvailableCallback](#) ()
- OSCL_IMPORT_REF void [addRef](#) ()
- OSCL_IMPORT_REF void [removeRef](#) ()

Protected Methods

- virtual OSCL_IMPORT_REF ~[OsclMemPoolResizableAllocator](#) ()
- [MemPoolBufferInfo](#) * [addnewmempoolbuffer](#) (uint32 aBufferSize)
- void [destroyallmempoolbuffers](#) ()
- [MemPoolBlockInfo](#) * [findfreeblock](#) (uint32 aBlockSize)
- [OsclAny](#) * [allocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr, uint32 aNumBytes)
- void [deallocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr)
- bool [validateblock](#) ([OsclAny](#) *aBlockBufPtr)
- uint32 [getMemPoolBufferSize](#) ([MemPoolBufferInfo](#) *aBufferInfo) const
- uint32 [getMemPoolBufferAllocatedSize](#) ([MemPoolBufferInfo](#) *aBufferInfo) const
- uint32 [memoryPoolBufferMgmtOverhead](#) () const

Protected Attributes

- uint32 `iMemPoolBufferSize`
- uint32 `iMemPoolBufferNumLimit`
- uint32 `iExpectedNumBlocksPerBuffer`
- uint32 `iMaxNewMemPoolBufferSz`
- `Oscl_DefAlloc` * `iMemPoolBufferAllocator`
- `Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator >` `iMemPoolBufferList`
- uint32 `iBufferInfoAlignedSize`
- uint32 `iBlockInfoAlignedSize`
- bool `iCheckNextAvailable`
- uint32 `iRequestedNextAvailableSize`
- `OsclAny` * `iNextAvailableContextData`
- `OsclMemPoolResizableAllocatorObserver` * `iObserver`
- bool `iCheckFreeMemoryAvailable`
- uint32 `iRequestedAvailableFreeMemSize`
- `OsclAny` * `iFreeMemContextData`
- `OsclMemPoolResizableAllocatorMemoryObserver` * `iFreeMemPoolObserver`
- int32 `iRefCount`
- bool `iEnableNullPtrReturn`

7.162.1 Constructor & Destructor Documentation

7.162.1.1 OSCL_IMPORT_REF OsclMemPoolResizableAllocator::OsclMemPoolResizableAllocator (uint32 *aMemPoolBufferSize*, uint32 *aMemPoolBufferNumLimit* = 0, uint32 *aExpectedNumBlocksPerBuffer* = 0, `Oscl_DefAlloc` * *gen_alloc* = NULL)

Create the memory pool allocator with resizing functionality. The size of the memory pool buffer needs to be passed-in. The maximum number of memory pool buffers, expected number of blocks in a memory pool buffer, and outside allocator are optional. This API throws an exception when the memory allocation for the pool buffer fails. If memory pool buffer number limit parameter is not set, the assumption is that there is no limit and memory pool will grow as needed. If the expected number of blocks is not set or not known, the memory pool will use a default value to 10 to allocate extra memory for the block info header.

Returns:

`void`

7.162.1.2 virtual OSCL_IMPORT_REF OsclMemPoolResizableAllocator::~OsclMemPoolResizableAllocator () [protected, virtual]

The destructor for the memory pool. Should not be called directly. Use `removeRef()` instead.

7.162.2 Member Function Documentation

7.162.2.1 `MemPoolBufferInfo*` OsclMemPoolResizableAllocator::addnewmempoolbuffer (uint32 *aBufferSize*) [protected]

7.162.2.2 OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::addRef ()

Increments the reference count for this memory pool allocator

Returns:

void

**7.162.2.3 virtual OSCL_IMPORT_REF [OsclAny](#)* OsclMemPoolResizableAllocator::allocate
(const uint32 aNumBytes) [virtual]**

Allocates a block from the memory pool that is at least in size requested This API throws an exception if there isn't enough memory (if "enablenullpointerreturn" has not been called) for the requested amount in the pool or if the extra pool buffer cannot be allocated.

Returns:

Pointer to memory buffer from memory pool

Implements [Oscl_DefAlloc](#).

**7.162.2.4 [OsclAny](#)* OsclMemPoolResizableAllocator::allocateblock ([MemPoolBlockInfo](#) &
aBlockPtr, uint32 aNumBytes) [protected]****7.162.2.5 virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::CancelFree-
ChunkAvailableCallback () [virtual]**

This API will cancel any past callback requests..

Returns:

void

**7.162.2.6 OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::CancelFreeMemory-
AvailableCallback ()****7.162.2.7 virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::deallocate
([OsclAny](#) * aPtr) [virtual]**

Deallocates and returns a block back to the memory pool This API throws an exception if the pointer passed in is not part of the memory pool, aligned, or has corrupted block header.

Returns:

void

Implements [Oscl_DefAlloc](#).

**7.162.2.8 void OsclMemPoolResizableAllocator::deallocateblock ([MemPoolBlockInfo](#) &
aBlockPtr) [protected]****7.162.2.9 void OsclMemPoolResizableAllocator::destroyallmempoolbuffers () [protected]****7.162.2.10 virtual OSCL_IMPORT_REF void OsclMemPoolResizable-
Allocator::enablenullpointerreturn () [virtual]**

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL_LEAVE\(OsclErrNoResources\)](#)" allocate API will return NULL.

Returns:

void

7.162.2.11 **MemPoolBlockInfo*** OsclMemPoolResizableAllocator::findfreeblock (**uint32 aBlockSize**) [protected]

7.162.2.12 **virtual OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getAllocatedSize ()** [virtual]

Returns the number of bytes allocated from the buffer<including the overhead bytes that may be allocated by the allocator to keep track of the chunks allocated>

7.162.2.13 **virtual OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getAvailableSize ()** [virtual]

Returns the number of bytes available with the buffer

7.162.2.14 **OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getBufferSize ()**

Returns the size of the buffer <including the overhead bytes that may be allocated by the allocator>

7.162.2.15 **virtual OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getLargestContiguousFreeBlockSize ()** [virtual]

Returns the size of the largest available chunk in the memory.

7.162.2.16 **uint32 OsclMemPoolResizableAllocator::getMemPoolBufferAllocatedSize (MemPoolBufferInfo * aBufferInfo) const** [protected]

7.162.2.17 **uint32 OsclMemPoolResizableAllocator::getMemPoolBufferSize (MemPoolBufferInfo * aBufferInfo) const** [protected]

7.162.2.18 **uint32 OsclMemPoolResizableAllocator::memoryPoolBufferMgmtOverhead ()** [protected]

7.162.2.19 **virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::notifyfreeblockavailable (OsclMemPoolResizableAllocatorObserver & aObserver, uint32 aRequestedSize = 0, OsclAny * aContextData = NULL)** [virtual]

This API will set the flag to send a callback via specified observer object when the next memory block is deallocated by [deallocate\(\)](#) call. If the optional requested size parameter is set, the callback is sent when a free memory space of requested size becomes available. The optional context data is returned with the callback and can be used by the user to differentiate between different instances of memory pool objects. This memory pool only allows one notify to be queued. Another call to this function will just overwrite the previous call.

Returns:

void

7.162.2.20 `virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::notifyfreememoryavailable (OsclMemPoolResizableAllocatorMemoryObserver & aObserver, uint32 aRequestedSize = 0, OsclAny * aContextData = NULL)`
[virtual]

7.162.2.21 `OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::removeRef ()`

Decrements the reference count for this memory pool allocator When the reference count goes to 0, this instance of the memory pool object is deleted

Returns:

`void`

7.162.2.22 `virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::setMaxSzForNewMemPoolBuffer (uint32 aMaxNewMemPoolBufferSz)`
[virtual]

7.162.2.23 `virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::trim (OsclAny * aPtr, uint32 aBytesToFree) [virtual]`

Returns a tail segment of a previously allocated memory block back to the memory pool. The passed-in pointer to the memory buffer is still valid after the call completes but the buffer size is smaller by by specified amount that was freed. This function allows the user to allocate a larger size block initially when the amount needed is unknown and then return the unused portion of the block when the amount becomes known. This API throws an exception if the pointer passed in is not part of the memory pool or the size to return is bigger than the size of the passed-in block. Exception will be thrown if the memory pool is not set up yet.

Returns:

`bool` True if trim operation successful. False if the block wasn't trimmed

7.162.2.24 `bool OsclMemPoolResizableAllocator::validateblock (OsclAny * aBlockBufPtr)`
[protected]

7.162.3 Field Documentation

7.162.3.1 `uint32 OsclMemPoolResizableAllocator::iBlockInfoAlignedSize` [protected]

7.162.3.2 `uint32 OsclMemPoolResizableAllocator::iBufferInfoAlignedSize` [protected]

7.162.3.3 `bool OsclMemPoolResizableAllocator::iCheckFreeMemoryAvailable` [protected]

7.162.3.4 `bool OsclMemPoolResizableAllocator::iCheckNextAvailable` [protected]

7.162.3.5 `bool OsclMemPoolResizableAllocator::iEnableNullPtrReturn` [protected]

7.162.3.6 `uint32 OsclMemPoolResizableAllocator::iExpectedNumBlocksPerBuffer`
[protected]

7.162.3.7 `OsclAny* OsclMemPoolResizableAllocator::iFreeMemContextData` [protected]

7.162.3.8 `OsclMemPoolResizableAllocatorMemoryObserver* OsclMemPoolResizable-
Allocator::iFreeMemPoolObserver` [protected]

7.162.3.9 `uint32 OsclMemPoolResizableAllocator::iMaxNewMemPoolBufferSz` [protected]

7.162.3.10 `Oscl_DefAlloc* OsclMemPoolResizableAllocator::iMemPoolBufferAllocator`
[protected]

7.162.3.11 `Oscl_Vector<MemPoolBufferInfo*, OsclMemAllocator>`
`OsclMemPoolResizableAllocator::iMemPoolBufferList` [protected]

7.162.3.12 `uint32 OsclMemPoolResizableAllocator::iMemPoolBufferNumLimit` [protected]

7.162.3.13 `uint32 OsclMemPoolResizableAllocator::iMemPoolBufferSize` [protected]

7.162.3.14 `OsclAny* OsclMemPoolResizableAllocator::iNextAvailableContextData`
[protected]

7.162.3.15 `OsclMemPoolResizableAllocatorObserver* OsclMemPoolResizableAllocator::i-
Observer` [protected]

7.162.3.16 `int32 OsclMemPoolResizableAllocator::iRefCount` [protected]

7.162.3.17 `uint32 OsclMemPoolResizableAllocator::iRequestedAvailableFreeMemSize`
[protected]

7.162.3.18 `uint32 OsclMemPoolResizableAllocator::iRequestedNextAvailableSize`
[protected]

The documentation for this class was generated from the following file:

- `oscl_mem_mempool.h`

7.163 OsclMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference

```
#include <oscl_mem_mempool.h>
```

Data Fields

- uint32 iBlockPreFence
- MemPoolBlockInfo * iNextFreeBlock
- MemPoolBlockInfo * iPrevFreeBlock
- uint32 iBlockSize
- uint8 * iBlockBuffer
- MemPoolBufferInfo * iParentBuffer
- uint32 iBlockPostFence

7.163.1 Field Documentation

7.163.1.1 uint8* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockBuffer

7.163.1.2 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPostFence

7.163.1.3 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPreFence

7.163.1.4 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockSize

7.163.1.5 MemPoolBlockInfo* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iNextFree-Block

7.163.1.6 MemPoolBufferInfo* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iParent-Buffer

7.163.1.7 MemPoolBlockInfo* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iPrevFree-Block

The documentation for this struct was generated from the following file:

- `oscl_mem_mempool.h`

7.164 OsclMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference

```
#include <oscl_mem_mempool.h>
```

Data Fields

- uint32 iBufferPreFence
- [OsclAny](#) * iStartAddr
- [OsclAny](#) * iEndAddr
- uint32 iBufferSize
- uint32 iNumOutstanding
- [MemPoolBlockInfo](#) * iNextFreeBlock
- uint32 iAllocatedSz
- uint32 iBufferPostFence

7.164.1 Field Documentation

7.164.1.1 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iAllocatedSz

7.164.1.2 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPostFence

7.164.1.3 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPreFence

7.164.1.4 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferSize

7.164.1.5 [OsclAny](#)* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iEndAddr

7.164.1.6 [MemPoolBlockInfo](#)* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iNextFree-Block

7.164.1.7 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iNumOutstanding

7.164.1.8 [OsclAny](#)* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iStartAddr

The documentation for this struct was generated from the following file:

- [oscl_mem_mempool.h](#)

7.165 OsclMemPoolResizableAllocatorMemoryObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

Public Methods

- virtual void [freememoryavailable \(OsclAny *aContextData\)=0](#)
- virtual [~OsclMemPoolResizableAllocatorMemoryObserver \(\)](#)

7.165.1 Constructor & Destructor Documentation

7.165.1.1 virtual OsclMemPoolResizableAllocatorMemoryObserver::~OsclMemPoolResizableAllocatorMemoryObserver () [inline, virtual]

7.165.2 Member Function Documentation

7.165.2.1 virtual void OsclMemPoolResizableAllocatorMemoryObserver::freememoryavailable (OsclAny * aContextData) [pure virtual]

The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

7.166 OsclMemPoolResizableAllocatorObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

Public Methods

- virtual void [freeblockavailable \(OsclAny *aContextData\)=0](#)
- virtual [~OsclMemPoolResizableAllocatorObserver \(\)](#)

7.166.1 Constructor & Destructor Documentation

7.166.1.1 [virtual OsclMemPoolResizableAllocatorObserver::~OsclMemPoolResizableAllocatorObserver \(\) \[inline, virtual\]](#)

7.166.2 Member Function Documentation

7.166.2.1 [virtual void OsclMemPoolResizableAllocatorObserver::freeblockavailable \(OsclAny **aContextData*\) \[pure virtual\]](#)

The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

7.167 OsclMemStatsNode Class Reference

```
#include <oscl_mem_audit.h>
```

Public Methods

- [OsclMemStatsNode \(\)](#)
- [void reset \(\)](#)
- [~OsclMemStatsNode \(\)](#)
- [void * operator new \(oscl_memsize_t size\)](#)
- [void * operator new \(oscl_memsize_t size, OsclMemStatsNode *ptr\)](#)
- [void operator delete \(void *ptr\) throw \(\)](#)

Data Fields

- [MM_Stats_t * pMMStats](#)
- [MM_FailInsertParam * pMMFIParam](#)
- [char * tag](#)

7.167.1 Constructor & Destructor Documentation

7.167.1.1 OsclMemStatsNode::OsclMemStatsNode () [inline]

7.167.1.2 OsclMemStatsNode::~OsclMemStatsNode () [inline]

7.167.2 Member Function Documentation

7.167.2.1 void OsclMemStatsNode::operator delete (void *ptr) throw () [inline]

7.167.2.2 void* OsclMemStatsNode::operator new (oscl_memsize_t size, OsclMemStatsNode *ptr) [inline]

7.167.2.3 void* OsclMemStatsNode::operator new (oscl_memsize_t size) [inline]

7.167.2.4 void OsclMemStatsNode::reset () [inline]

7.167.3 Field Documentation

7.167.3.1 MM_FailInsertParam* OsclMemStatsNode::pMMFIParam

7.167.3.2 MM_Stats_t* OsclMemStatsNode::pMMStats

7.167.3.3 char* OsclMemStatsNode::tag

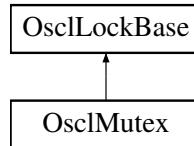
The documentation for this class was generated from the following file:

- [oscl_mem_audit.h](#)

7.168 OsclMutex Class Reference

```
#include <oscl_mutex.h>
```

Inheritance diagram for OsclMutex::



Public Methods

- OSCL_IMPORT_REF OsclMutex ()
- virtual OSCL_IMPORT_REF ~OsclMutex ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Create (void)
- OSCL_IMPORT_REF void Lock ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError TryLock ()
- OSCL_IMPORT_REF void Unlock ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Close (void)

7.168.1 Detailed Description

Class OsclMutex

7.168.2 Constructor & Destructor Documentation

7.168.2.1 OSCL_IMPORT_REF OsclMutex::OsclMutex ()

Class constructor

7.168.2.2 virtual OSCL_IMPORT_REF OsclMutex::~OsclMutex () [virtual]

Class destructor

7.168.3 Member Function Documentation

7.168.3.1 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclMutex::Close (void)

Closes the Mutex

Parameters:

It wont take any parameters

Returns:

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

7.168.3.2 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclMutex::Create (void)

Creates the Mutex

Parameters:

No input arguments

Returns:

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

7.168.3.3 OSCL_IMPORT_REF void OsclMutex::Lock () [virtual]

Locks the Mutex

Parameters:

It wont take any parameters

Returns:

Returns nothing

Implements [OsclLockBase](#).

7.168.3.4 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclMutex::TryLock ()

Try to lock the mutex,if the Mutex is already locked calling thread immediately returns with out blocking

Parameters:

It wont take any parameters

Returns:

Returns SUCCESS_ERROR if the mutex was acquired, MUTEX_LOCKED_ERROR if the mutex cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT_IMPLEMENTED.

7.168.3.5 OSCL_IMPORT_REF void OsclMutex::Unlock () [virtual]

Releases the Mutex

Parameters:

It wont take any parameters

Returns:

Returns nothing

Implements [OsclLockBase](#).

The documentation for this class was generated from the following file:

- [oscl_mutex.h](#)

7.169 OsclNameString< __len > Class Template Reference

```
#include <oscl_namestring.h>
```

Public Methods

- [OsclNameString \(\)](#)
- [OsclNameString \(const char a\[\]\)](#)
- [OsclNameString \(uint8 *a\)](#)
- void [Set \(uint8 *a\)](#)
- void [Set \(const char a\[\]\)](#)
- uint8 * [Str \(\) const](#)
- int32 [MaxLen \(\) const](#)

7.169.1 Detailed Description

`template<int __len> class OsclNameString< __len >`

Name string class appropriate for passing short constant ASCII strings around. All strings are automatically truncated and null-terminated.

7.169.2 Constructor & Destructor Documentation

7.169.2.1 template<int __len> OsclNameString< __len >::OsclNameString () [inline]

7.169.2.2 template<int __len> OsclNameString< __len >::OsclNameString (const char a[]) [inline]

7.169.2.3 template<int __len> OsclNameString< __len >::OsclNameString (uint8 * a) [inline]

7.169.3 Member Function Documentation

7.169.3.1 template<int __len> int32 OsclNameString< __len >::MaxLen () const [inline]

7.169.3.2 template<int __len> void OsclNameString< __len >::Set (const char a[]) [inline]

7.169.3.3 template<int __len> void OsclNameString< __len >::Set (uint8 * a) [inline]

Set the string to the input value. The string will be truncated to fit the storage class and automatically null-terminated.

Parameters:

a (input param): null-terminated character string.

7.169.3.4 template<int __len> uint8* OsclNameString< __len >::Str () const [inline]

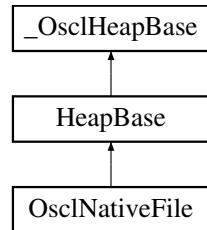
The documentation for this class was generated from the following file:

- [oscl_namestring.h](#)

7.170 OsclNativeFile Class Reference

```
#include <oscl_file_native.h>
```

Inheritance diagram for OsclNativeFile::



Public Methods

- [OsclNativeFile \(\)](#)
- [~OsclNativeFile \(\)](#)
- [int32 Open \(const OsclFileHandle &, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv\)](#)
- [int32 Open \(const oscl_wchar *filename, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv\)](#)
- [int32 Open \(const char *filename, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv\)](#)
- [uint32 Read \(OsclAny *buffer, uint32 size, uint32 numelements\)](#)
- [uint32 Write \(const OsclAny *buffer, uint32 size, uint32 numelements\)](#)
- [int32 Seek \(TOsclFileOffset offset, Oscl_File::seek_type origin\)](#)
- [TOsclFileOffset Tell \(\)](#)
- [int32 Flush \(\)](#)
- [int32 EndOfFile \(\)](#)
- [TOsclFileOffset Size \(\)](#)
- [int32 Close \(\)](#)
- [uint32 Mode \(\)](#)
- [int32 GetError \(\)](#)
- [int32 ReadAsync \(OsclAny *buffer, uint32 size, uint32 numelements, OsclAOStatus &status\)](#)
- [uint32 GetReadAsyncNumElements \(\)](#)
- [bool HasAsyncRead \(\)](#)
- [void ReadAsyncCancel \(\)](#)

7.170.1 Constructor & Destructor Documentation

7.170.1.1 OsclNativeFile::OsclNativeFile ()

7.170.1.2 OsclNativeFile::~OsclNativeFile ()

7.170.2 Member Function Documentation

7.170.2.1 int32 OsclNativeFile::Close ()

7.170.2.2 int32 OsclNativeFile::EndOfFile ()

7.170.2.3 int32 OsclNativeFile::Flush ()

7.170.2.4 int32 OsclNativeFile::GetError ()

7.170.2.5 uint32 OsclNativeFile::GetReadAsyncNumElements ()

Get the number of elements read in the last call to ReadAsync. @returns: number of elements read.

7.170.2.6 bool OsclNativeFile::HasAsyncRead ()

@returns: true if async read is supported natively.

7.170.2.7 uint32 OsclNativeFile::Mode () [inline]

7.170.2.8 int32 OsclNativeFile::Open (const char *filename, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv)

7.170.2.9 int32 OsclNativeFile::Open (const oscl_wchar *filename, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv)

7.170.2.10 int32 OsclNativeFile::Open (const OsclFileHandle &, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv)

7.170.2.11 uint32 OsclNativeFile::Read (OsclAny *buffer, uint32 size, uint32 numelements)

7.170.2.12 int32 OsclNativeFile::ReadAsync (OsclAny *buffer, uint32 size, uint32 numelements, OsclAOStatus &status)

Asynchronous read.

Parameters:

buffer: data buffer, must be at least size*numelements bytes

size: size of elements

numelements: number of elements to read

status: Request status for asynchronous completion @returns: 0 for success.

7.170.2.13 void OsclNativeFile::ReadAsyncCancel ()

Cancel any pending async read.

7.170.2.14 int32 OsclNativeFile::Seek ([TOsclFileOffset](#) *offset*, [Oscl_File::seek_type](#) *origin*)**7.170.2.15 [TOsclFileOffset](#) OsclNativeFile::Size ()****7.170.2.16 [TOsclFileOffset](#) OsclNativeFile::Tell ()****7.170.2.17 uint32 OsclNativeFile::Write (const [OsclAny](#) * *buffer*, uint32 *size*, uint32 *numelements*)**

The documentation for this class was generated from the following file:

- [oscl_file_native.h](#)

7.171 OsclNativeFileParams Class Reference

```
#include <oscl_file_types.h>
```

Public Methods

- [OsclNativeFileParams \(uint32 mode=0, uint32 bufsize=0, uint32 asyncsize=0\)](#)

Data Fields

- uint32 [iNativeAccessMode](#)
- uint32 [iNativeBufferSize](#)
- uint32 [iAsyncReadBufferSize](#)

7.171.1 Constructor & Destructor Documentation

7.171.1.1 OsclNativeFileParams::OsclNativeFileParams (uint32 mode = 0, uint32 bufsize = 0, uint32 asyncsize = 0) [inline]

7.171.2 Field Documentation

7.171.2.1 uint32 OsclNativeFileParams::iAsyncReadBufferSize

7.171.2.2 uint32 OsclNativeFileParams::iNativeAccessMode

7.171.2.3 uint32 OsclNativeFileParams::iNativeBufferSize

The documentation for this class was generated from the following file:

- [oscl_file_types.h](#)

7.172 OsclNetworkAddress Class Reference

```
#include <oscl_socket_types.h>
```

Public Methods

- [OsclNetworkAddress \(\)](#)
- [OsclNetworkAddress \(const char *addr, int p\)](#)
- [bool operator== \(const OsclNetworkAddress &rhs\) const](#)

Data Fields

- [OsclNameString< PVNETWORKADDRESS_LEN > ipAddr](#)
- [int port](#)

7.172.1 Constructor & Destructor Documentation

7.172.1.1 [OsclNetworkAddress::OsclNetworkAddress \(\) \[inline\]](#)

7.172.1.2 [OsclNetworkAddress::OsclNetworkAddress \(const char *addr, int p\) \[inline\]](#)

7.172.2 Member Function Documentation

7.172.2.1 [bool OsclNetworkAddress::operator== \(const OsclNetworkAddress & rhs\) const \[inline\]](#)

7.172.3 Field Documentation

7.172.3.1 [OsclNameString<PVNETWORKADDRESS_LEN> OsclNetworkAddress::ipAddr](#)

7.172.3.2 [int OsclNetworkAddress::port](#)

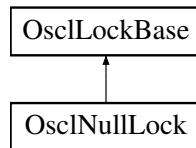
The documentation for this class was generated from the following file:

- [oscl_socket_types.h](#)

7.173 OsclNullLock Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclNullLock::



Public Methods

- virtual void [Lock \(\)](#)
- virtual void [Unlock \(\)](#)
- virtual [~OsclNullLock \(\)](#)

7.173.1 Constructor & Destructor Documentation

7.173.1.1 virtual OsclNullLock::~OsclNullLock () [inline, virtual]

7.173.2 Member Function Documentation

7.173.2.1 virtual void OsclNullLock::Lock () [inline, virtual]

Implements [OsclLockBase](#).

7.173.2.2 virtual void OsclNullLock::Unlock () [inline, virtual]

Implements [OsclLockBase](#).

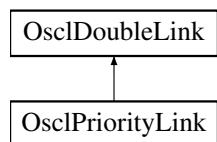
The documentation for this class was generated from the following file:

- [oscl_lock_base.h](#)

7.174 OsclPriorityLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclPriorityLink::



Data Fields

- int32 [iPriority](#)

7.174.1 Field Documentation

7.174.1.1 int32 OsclPriorityLink::iPriority

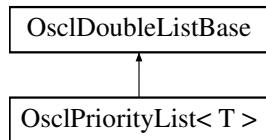
The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

7.175 OsclPriorityList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclPriorityList< T >::



Public Methods

- OSCL_INLINE OsclPriorityList ()
- OSCL_INLINE OsclPriorityList (int32 anOffset)
- OSCL_INLINE void Insert (T &aRef)
- OSCL_INLINE bool IsHead (const T *aPtr) const
- OSCL_INLINE bool IsTail (const T *aPtr) const
- OSCL_INLINE T * Head () const
- OSCL_INLINE T * Tail () const

```
template<class T> class OsclPriorityList< T >
```

7.175.1 Constructor & Destructor Documentation

7.175.1.1 template<class T> OSCL_INLINE OsclPriorityList< T >::OsclPriorityList ()

7.175.1.2 template<class T> OSCL_INLINE OsclPriorityList< T >::OsclPriorityList (int32 *anOffset*)

7.175.2 Member Function Documentation

7.175.2.1 template<class T> OSCL_INLINE T* OsclPriorityList< T >::Head ()

7.175.2.2 template<class T> OSCL_INLINE void OsclPriorityList< T >::Insert (T &*aRef*)

7.175.2.3 template<class T> OSCL_INLINE bool OsclPriorityList< T >::IsHead (const T **aPtr*) const

7.175.2.4 template<class T> OSCL_INLINE bool OsclPriorityList< T >::IsTail (const T **aPtr*) const

7.175.2.5 template<class T> OSCL_INLINE T* OsclPriorityList< T >::Tail ()

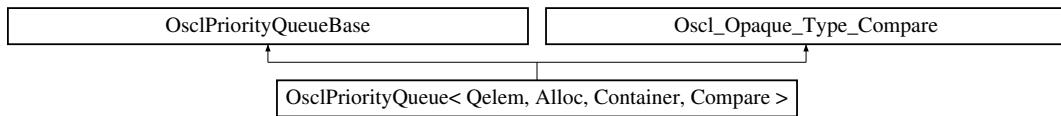
The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

7.176 OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference

```
#include <oscl_priqueue.h>
```

Inheritance diagram for OsclPriorityQueue< Qelem, Alloc, Container, Compare >::



Public Types

- `typedef Container::value_type value_type`
- `typedef Container container_type`
- `typedef Container::iterator iterator`
- `typedef Container::const_reference const_reference`

Public Methods

- `bool empty () const`
- `uint32 size () const`
- `void reserve (uint32 n)`
- `const_reference top () const`
- `const Container & vec ()`
- `void push (const value_type &input)`
- `void pop ()`
- `int remove (const value_type &input)`
- `OsclPriorityQueue ()`
- `virtual ~OsclPriorityQueue ()`

Protected Methods

- `void push_heap (iterator first, iterator last)`
- `void pop_heap (iterator first, iterator last)`
- `iterator find_heap (const value_type &input, iterator first, iterator last)`
- `int validate ()`
- `void swap (OsclAny *dest, const OsclAny *src)`
- `int compare_LT (OsclAny *a, OsclAny *b) const`
- `int compare_EQ (const OsclAny *a, const OsclAny *b) const`

Protected Attributes

- `Container c`
- `Compare comp`

Friends

- class [oscl_priqueue_test](#)

```
template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> class OsclPriorityQueue< Qelem, Alloc, Container, Compare >
```

7.176.1 Member Typedef Documentation

- 7.176.1.1 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container::const_reference OsclPriorityQueue< Qelem, Alloc, Container, Compare >::const_reference**
- 7.176.1.2 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::container_type**
- 7.176.1.3 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container::iterator OsclPriorityQueue< Qelem, Alloc, Container, Compare >::iterator**
- 7.176.1.4 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **typedef Container::value_type OsclPriorityQueue< Qelem, Alloc, Container, Compare >::value_type**

7.176.2 Constructor & Destructor Documentation

- 7.176.2.1 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **OsclPriorityQueue< Qelem, Alloc, Container, Compare >::OsclPriorityQueue () [inline]**
- 7.176.2.2 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **virtual OsclPriorityQueue< Qelem, Alloc, Container, Compare >::~OsclPriorityQueue () [inline, virtual]**

7.176.3 Member Function Documentation

- 7.176.3.1 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::compare_EQ (const OsclAny * a, const OsclAny * b) const [inline, protected, virtual]**

Return a==b.

Implements [Oscl_Opaque_Type_Compare](#).

- 7.176.3.2 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::compare_LT (OsclAny * a, OsclAny * b) const [inline, protected, virtual]**

Return a<b.

Implements [Oscl_Opaque_Type_Compare](#).

- 7.176.3.3 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> bool OsclPriorityQueue< Qelem, Alloc, Container, Compare >::empty () const [inline]`
- 7.176.3.4 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> iterator OsclPriorityQueue< Qelem, Alloc, Container, Compare >::find_heap (const value_type & input, iterator first, iterator last) [inline, protected]`
- 7.176.3.5 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::pop () [inline]`
- 7.176.3.6 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::pop_heap (iterator first, iterator last) [inline, protected]`
- 7.176.3.7 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::push (const value_type & input) [inline]`
- 7.176.3.8 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::push_heap (iterator first, iterator last) [inline, protected]`
- 7.176.3.9 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::remove (const value_type & input) [inline]`
- 7.176.3.10 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::reserve (uint32 n) [inline]`
- 7.176.3.11 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> uint32 OsclPriorityQueue< Qelem, Alloc, Container, Compare >::size () const [inline]`
- 7.176.3.12 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::swap (OsclAny * dest, const OsclAny * src) [inline, protected, virtual]`

Swap element at "a" with element at "b". Both pointers must be non-NULL.

Implements [Oscl_Opaque_Type_Compare](#).

- 7.176.3.13 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **const_reference** OsclPriorityQueue< Qelem, Alloc, Container, Compare >::top () const [inline]
- 7.176.3.14 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::validate () [inline, protected]
- 7.176.3.15 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **const Container&** OsclPriorityQueue< Qelem, Alloc, Container, Compare >::vec () [inline]

7.176.4 Friends And Related Function Documentation

- 7.176.4.1 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> friend class oscl_priqueue_test [friend]

7.176.5 Field Documentation

- 7.176.5.1 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::c [protected]
- 7.176.5.2 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Compare OsclPriorityQueue< Qelem, Alloc, Container, Compare >::comp [protected]

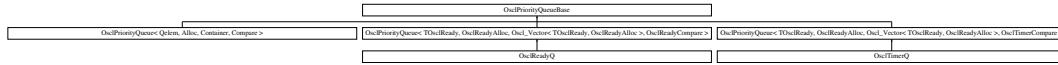
The documentation for this class was generated from the following file:

- [oscl_priqueue.h](#)

7.177 OsclPriorityQueueBase Class Reference

```
#include <oscl_priqueue.h>
```

Inheritance diagram for OsclPriorityQueueBase::



Protected Methods

- virtual ~OsclPriorityQueueBase ()
- OSCL_IMPORT_REF void [push_heap](#) (OsclAny *first, OsclAny *last)
- OSCL_IMPORT_REF void [pop_heap](#) (OsclAny *first, OsclAny *last)
- OSCL_IMPORT_REF OsclAny * [find_heap](#) (const OsclAny *input, OsclAny *first, OsclAny *last)
- OSCL_IMPORT_REF int [remove](#) (const OsclAny *input)
- void [construct](#) (Oscl_Opaque_Type_Compare *ot, Oscl_Vector_Base *vec)

7.177.1 Detailed Description

OsclPriorityQueueBase is a non-templated base class for [OsclPriorityQueue](#). The purpose of this base class is to avoid large inline routines in the [OsclPriorityQueue](#) implementation. This class is not intended for direct instantiation except by [OsclPriorityQueue](#).

7.177.2 Constructor & Destructor Documentation

7.177.2.1 virtual OsclPriorityQueueBase::~OsclPriorityQueueBase () [inline, protected, virtual]

7.177.3 Member Function Documentation

7.177.3.1 void OsclPriorityQueueBase::construct (Oscl_Opaque_Type_Compare * ot, Oscl_Vector_Base * vec) [inline, protected]

7.177.3.2 OSCL_IMPORT_REF OsclAny* OsclPriorityQueueBase::find_heap (const OsclAny * input, OsclAny *first, OsclAny * last) [protected]

7.177.3.3 OSCL_IMPORT_REF void OsclPriorityQueueBase::pop_heap (OsclAny *first, OsclAny * last) [protected]

7.177.3.4 OSCL_IMPORT_REF void OsclPriorityQueueBase::push_heap (OsclAny *first, OsclAny * last) [protected]

7.177.3.5 OSCL_IMPORT_REF int OsclPriorityQueueBase::remove (const OsclAny * input) [protected]

The documentation for this class was generated from the following file:

- [oscl_priqueue.h](#)

7.178 OsclProcStatus Class Reference

```
#include <oscl_procstatus.h>
```

Public Types

- enum `eOsclProcError` { `SUCCESS_ERROR` = 0, `OTHER_ERROR`, `TOO_MANY_THREADS_ERROR`, `BAD_THREADID_ADDR_ERROR`, `MAX_THRDS_REACHED_ERROR`, `INVALID_THREAD_ID_ERROR`, `NOT_ENOUGH_MEMORY_ERROR`, `OUTOFMEMORY_ERROR`, `NOT_ENOUGH_RESOURCES_ERROR`, `THREAD_1_INACTIVE_ERROR`, `ALREADY_SUSPENDED_ERROR`, `NOT_SUSPENDED_ERROR`, `INVALID_THREAD_ERROR`, `INVALID_PARAM_ERROR`, `NO_PERMISSION_ERROR`, `INVALID_PRIORITY_ERROR`, `PSHARED_NOT_ZERO_ERROR`, `EXCEED_MAX_COUNT_VARIABLE_ERROR`, `THREAD_BLOCK_ERROR`, `EXCEED_MAX_SEM_COUNT_ERROR`, `INVALID_HANDLE_ERROR`, `INVALID_OPERATION_ERROR`, `INVALID_FUNCTION_ERROR`, `INVALID_ACCESS_ERROR`, `INVALID_ARGUMENT_ERROR`, `SYSTEM_RESOURCES_UNAVAILABLE_ERROR`, `INVALID_POINTER_ERROR`, `RELOCK_MUTEX_ERROR`, `THREAD_NOT_OWN_MUTEX_ERROR`, `MUTEX_LOCKED_ERROR`, `WAIT_ABANDONED_ERROR`, `WAIT_TIMEOUT_ERROR`, `SEM_NOT_SIGNALED_ERROR`, `PSHARED_ATTRIBUTE_SETTING_ERROR`, `NOT_IMPLEMENTED` }

7.178.1 Detailed Description

Class OsclProcStatus

7.178.2 Member Enumeration Documentation

7.178.2.1 enum OsclProcStatus::eOsclProcError

List of enums which contain error codes

Enumeration values:

`SUCCESS_ERROR`
`OTHER_ERROR`
`TOO_MANY_THREADS_ERROR`
`BAD_THREADID_ADDR_ERROR`
`MAX_THRDS_REACHED_ERROR`
`INVALID_THREAD_ID_ERROR`
`NOT_ENOUGH_MEMORY_ERROR`
`OUTOFMEMORY_ERROR`
`NOT_ENOUGH_RESOURCES_ERROR`
`THREAD_1_INACTIVE_ERROR`
`ALREADY_SUSPENDED_ERROR`
`NOT_SUSPENDED_ERROR`
`INVALID_THREAD_ERROR`
`INVALID_PARAM_ERROR`
`NO_PERMISSION_ERROR`

INVALID_PRIORITY_ERROR
PSHARED_NOT_ZERO_ERROR
EXCEED_MAX_COUNT_VARIABLE_ERROR
THREAD_BLOCK_ERROR
EXCEED_MAX_SEM_COUNT_ERROR
INVALID_HANDLE_ERROR
INVALID_OPERATION_ERROR
INVALID_FUNCTION_ERROR
INVALID_ACCESS_ERROR
INVALID_ARGUMENT_ERROR
SYSTEM_RESOURCES_UNAVAILABLE_ERROR
INVALID_POINTER_ERROR
RELOCK_MUTEX_ERROR
THREAD_NOT_OWN_MUTEX_ERROR
MUTEX_LOCKED_ERROR
WAIT_ABANDONED_ERROR
WAIT_TIMEOUT_ERROR
SEM_NOT_SIGNALED_ERROR
PSHARED_ATTRIBUTE_SETTING_ERROR
NOT_IMPLEMENTED

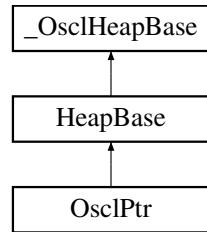
The documentation for this class was generated from the following file:

- [oscl_procstatus.h](#)

7.179 OsclPtr Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclPtr::



Public Methods

- [OsclPtr \(uint8 *ptr, int32 &len, int32 max\)](#)
- [OsclPtr \(const OsclPtr &d\)](#)
- [uint8 * Ptr \(\)](#)
- [void SetLength \(int32 l\)](#)
- [int32 Length \(\)](#)
- [void Zero \(\)](#)
- [void Set \(OsclPtr &v\)](#)
- [void Set \(uint8 *ptr, int32 len, int32 max\)](#)
- [void Append \(OsclPtrC &v\)](#)

7.179.1 Constructor & Destructor Documentation

7.179.1.1 OsclPtr::OsclPtr (uint8 *ptr, int32 &len, int32 max) [inline]

7.179.1.2 OsclPtr::OsclPtr (const OsclPtr &d) [inline]

7.179.2 Member Function Documentation

7.179.2.1 void OsclPtr::Append (OsclPtrC &v) [inline]

7.179.2.2 int32 OsclPtr::Length () [inline]

7.179.2.3 uint8* OsclPtr::Ptr () [inline]

7.179.2.4 void OsclPtr::Set (uint8 *ptr, int32 len, int32 max) [inline]

7.179.2.5 void OsclPtr::Set (OsclPtr &v) [inline]

7.179.2.6 void OsclPtr::SetLength (int32 l) [inline]

7.179.2.7 void OsclPtr::Zero () [inline]

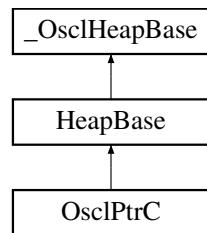
The documentation for this class was generated from the following file:

- [oscl_file_async_read.h](#)

7.180 OsclPtrC Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclPtrC::



Public Methods

- [OsclPtrC](#) (const uint8 *ptr, int32 len, int32 max)
- [OsclPtrC](#) (const OsclPtrC &d)
- const uint8 * [Ptr](#) ()
- void [SetLength](#) (int32 l)
- int32 [Length](#) ()
- void [Zero](#) ()
- void [Set](#) (OsclPtrC *v)
- void [Set](#) (uint8 *ptr, int32 len, int32 max)
- OsclPtrC [Right](#) (int32 size)
- OsclPtrC [Left](#) (int32 size)

7.180.1 Constructor & Destructor Documentation

7.180.1.1 `OsclPtrC::OsclPtrC (const uint8 *ptr, int32 len, int32 max)` [inline]

7.180.1.2 `OsclPtrC::OsclPtrC (const OsclPtrC & d)` [inline]

7.180.2 Member Function Documentation

7.180.2.1 `OsclPtrC OsclPtrC::Left (int32 size)` [inline]

7.180.2.2 `int32 OsclPtrC::Length ()` [inline]

7.180.2.3 `const uint8* OsclPtrC::Ptr ()` [inline]

7.180.2.4 `OsclPtrC OsclPtrC::Right (int32 size)` [inline]

7.180.2.5 `void OsclPtrC::Set (uint8 *ptr, int32 len, int32 max)` [inline]

7.180.2.6 `void OsclPtrC::Set (OsclPtrC *v)` [inline]

7.180.2.7 `void OsclPtrC::SetLength (int32 l)` [inline]

7.180.2.8 `void OsclPtrC::Zero ()` [inline]

The documentation for this class was generated from the following file:

- [oscl_file_async_read.h](#)

7.181 OsclRand Class Reference

```
#include <oscl_rand.h>
```

Public Methods

- OSCL_COND_IMPORT_REF void [Seed](#) (int32 seed)
- OSCL_COND_IMPORT_REF int32 [Rand](#) ()

7.181.1 Member Function Documentation

7.181.1.1 OSCL_COND_IMPORT_REF int32 OsclRand::Rand ()

7.181.1.2 OSCL_COND_IMPORT_REF void OsclRand::Seed (int32 *seed*)

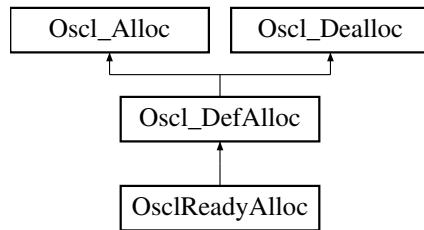
The documentation for this class was generated from the following file:

- [oscl_rand.h](#)

7.182 OsclReadyAlloc Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyAlloc::



Public Methods

- [OsclAny * allocate \(const uint32 size\)](#)
- [OsclAny * allocate_fl \(const uint32 size, const char *file_name, const int line_num\)](#)
- void [deallocate \(OsclAny *p\)](#)

7.182.1 Member Function Documentation

7.182.1.1 [OsclAny* OsclReadyAlloc::allocate \(const uint32 size\) \[virtual\]](#)

Implements [Oscl_DefAlloc](#).

7.182.1.2 [OsclAny* OsclReadyAlloc::allocate_fl \(const uint32 size, const char *file_name, const int line_num\) \[virtual\]](#)

Reimplemented from [Oscl_DefAlloc](#).

7.182.1.3 [void OsclReadyAlloc::deallocate \(OsclAny *p\) \[virtual\]](#)

Implements [Oscl_DefAlloc](#).

The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

7.183 OsclReadyCompare Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Static Public Methods

- int [compare \(TOsclReady &a, TOsclReady &b\)](#)

7.183.1 Member Function Documentation

7.183.1.1 int OsclReadyCompare::compare (TOsclReady &*a*, TOsclReady &*b*) [static]

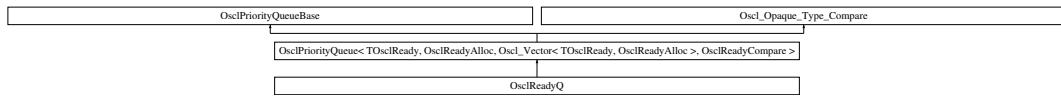
The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

7.184 OsclReadyQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyQ::



Public Methods

- void [Construct](#) (int)
- void [ThreadLogon](#) ()
- void [ThreadLogoff](#) ()
- void [Remove](#) (TOscIReady)
- bool [IsIn](#) (TOscIReady)
- uint32 [Depth](#) ()
- [TOscIReady PopTop](#) ()
- [TOscIReady Top](#) ()
- [TOscIReady WaitAndPopTop](#) ()
- [TOscIReady WaitAndPopTop](#) (uint32)
- int32 [PendComplete](#) (PVActiveBase *pvbase, int32 aReason)
- int32 [WaitForRequestComplete](#) (PVActiveBase *)
- void [RegisterForCallback](#) (OsclSchedulerObserver *aCallback, OsclAny *aCallbackContext)
- void [TimerCallback](#) (uint32 aDelayMicrosec)
- [OsclSchedulerObserver * Callback](#) ()

7.184.1 Member Function Documentation

7.184.1.1 **OsclSchedulerObserver*** OsclReadyQ::Callback () [inline]

7.184.1.2 void OsclReadyQ::Construct (int)

7.184.1.3 uint32 OsclReadyQ::Depth () [inline]

7.184.1.4 bool OsclReadyQ::IsIn (**TOsclReady**)

7.184.1.5 int32 OsclReadyQ::PendComplete (**PVActiveBase** **pvbase*, int32 *aReason*)

7.184.1.6 **TOsclReady** OsclReadyQ::PopTop ()

7.184.1.7 void OsclReadyQ::RegisterForCallback (**OsclSchedulerObserver** **aCallback*, **OsclAny** **aCallbackContext*)

7.184.1.8 void OsclReadyQ::Remove (**TOsclReady**)

7.184.1.9 void OsclReadyQ::ThreadLogoff ()

7.184.1.10 void OsclReadyQ::ThreadLogon ()

7.184.1.11 void OsclReadyQ::TimerCallback (uint32 *aDelayMicrosec*)

7.184.1.12 **TOsclReady** OsclReadyQ::Top ()

7.184.1.13 **TOsclReady** OsclReadyQ::WaitAndPopTop (uint32)

7.184.1.14 **TOsclReady** OsclReadyQ::WaitAndPopTop ()

7.184.1.15 int32 OsclReadyQ::WaitForRequestComplete (**PVActiveBase** *)

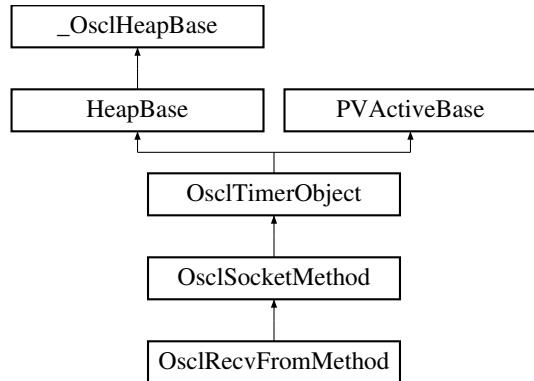
The documentation for this class was generated from the following file:

- `oscl_scheduler_readyq.h`

7.185 OsclRecvFromMethod Class Reference

```
#include <oscl_socket_recv_from.h>
```

Inheritance diagram for OsclRecvFromMethod:::



Public Methods

- [`~OsclRecvFromMethod \(\)`](#)
- [`TPVSocketEvent RecvFrom \(uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeout, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource\)`](#)
- [`uint8 * GetRecvData \(int32 *aLength\)`](#)
- [`OsclRecvFromRequest * RecvFromRequest \(\)`](#)

Static Public Methods

- [`OsclRecvFromMethod * NewL \(OsclIPSocketI &c\)`](#)

7.185.1 Constructor & Destructor Documentation

7.185.1.1 OsclRecvFromMethod::~OsclRecvFromMethod ()

7.185.2 Member Function Documentation

7.185.2.1 `uint8* OsclRecvFromMethod::GetRecvData (int32 * aLength)`

7.185.2.2 `OsclRecvFromMethod* OsclRecvFromMethod::NewL (OsclIPSocketI & c) [static]`

7.185.2.3 `TPVSocketEvent OsclRecvFromMethod::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeout, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource)`

7.185.2.4 `OsclRecvFromRequest* OsclRecvFromMethod::RecvFromRequest () [inline]`

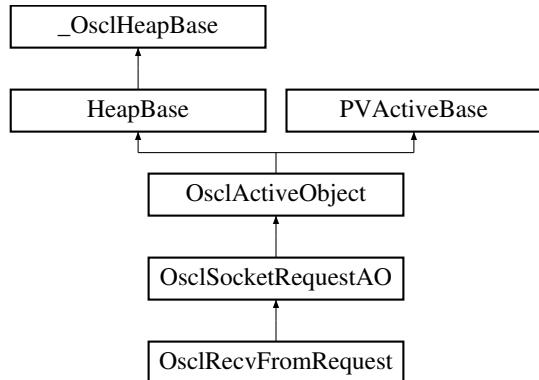
The documentation for this class was generated from the following file:

-
- [oscl_socket_recv_from.h](#)

7.186 OsclRecvFromRequest Class Reference

```
#include <oscl_socket_recv_from.h>
```

Inheritance diagram for OsclRecvFromRequest::



Public Methods

- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvFromRequest (OsclSocketMethod &c)`
- `void RecvFrom (uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource)`
- `void Success ()`

7.186.1 Detailed Description

This is the AO that interacts with the socket server

7.186.2 Constructor & Destructor Documentation

7.186.2.1 OsclRecvFromRequest::OsclRecvFromRequest (`OsclSocketMethod &c`) [inline]

7.186.3 Member Function Documentation

7.186.3.1 `uint8* OsclRecvFromRequest::GetRecvData (int32 * aLength)`

7.186.3.2 `void OsclRecvFromRequest::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource)`

7.186.3.3 `void OsclRecvFromRequest::Success () [virtual]`

Reimplemented from `OsclSocketRequestAO`.

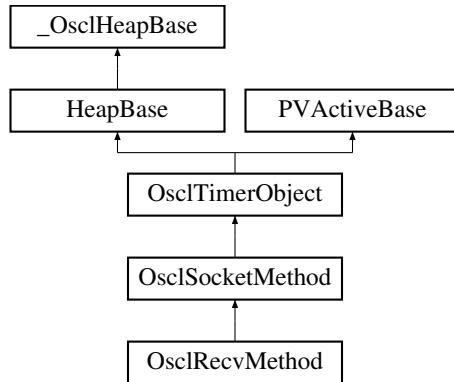
The documentation for this class was generated from the following file:

-
- [oscl_socket_recv_from.h](#)

7.187 OsclRecvMethod Class Reference

```
#include <oscl_socket_recv.h>
```

Inheritance diagram for OsclRecvMethod::



Public Methods

- [~OsclRecvMethod \(\)](#)
- [TPVSocketEvent Recv \(uint8 *&aPtr, uint32 aMaxLen, int32 aTimeout\)](#)
- [uint8 * GetRecvData \(int32 *aLength\)](#)
- [OsclRecvRequest * RecvRequest \(\)](#)

Static Public Methods

- [OsclRecvMethod * NewL \(OsclIPSocketI &c\)](#)

7.187.1 Constructor & Destructor Documentation

7.187.1.1 OsclRecvMethod::~OsclRecvMethod ()

7.187.2 Member Function Documentation

7.187.2.1 uint8* OsclRecvMethod::GetRecvData (int32 * aLength)

7.187.2.2 OsclRecvMethod* OsclRecvMethod::NewL (OsclIPSocketI & c) [static]

7.187.2.3 TPVSocketEvent OsclRecvMethod::Recv (uint8 *& aPtr, uint32 aMaxLen, int32 aTimeout)

7.187.2.4 OsclRecvRequest* OsclRecvMethod::RecvRequest () [inline]

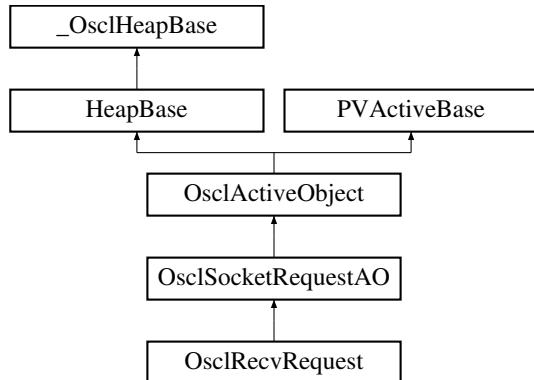
The documentation for this class was generated from the following file:

- [oscl_socket_recv.h](#)

7.188 OsclRecvRequest Class Reference

```
#include <oscl_socket_recv.h>
```

Inheritance diagram for OsclRecvRequest::



Public Methods

- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvRequest (OsclSocketMethod &c)`
- `void Recv (uint8 *&aPtr, uint32 aMaxLen)`
- `void Success ()`

7.188.1 Detailed Description

This is the AO that interacts with the socket server

7.188.2 Constructor & Destructor Documentation

7.188.2.1 OsclRecvRequest::OsclRecvRequest (`OsclSocketMethod &c`) [inline]

7.188.3 Member Function Documentation

7.188.3.1 `uint8* OsclRecvRequest::GetRecvData (int32 * aLength)`

7.188.3.2 `void OsclRecvRequest::Recv (uint8 *& aPtr, uint32 aMaxLen)`

7.188.3.3 `void OsclRecvRequest::Success () [virtual]`

Reimplemented from `OsclSocketRequestAO`.

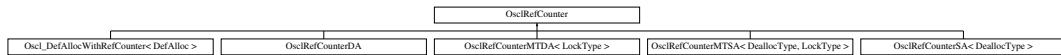
The documentation for this class was generated from the following file:

- `oscl_socket_recv.h`

7.189 OsclRefCounter Class Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounter:::



Public Methods

- virtual void `addRef ()=0`
- virtual void `removeRef ()=0`
- virtual uint32 `getCount ()=0`
- virtual `~OsclRefCounter ()`

7.189.1 Detailed Description

Interface class for OsclRefCounter implementations

7.189.2 Constructor & Destructor Documentation

7.189.2.1 virtual OsclRefCounter::~OsclRefCounter () [inline, virtual]

7.189.3 Member Function Documentation

7.189.3.1 virtual void OsclRefCounter::addRef () [pure virtual]

Add to the reference count

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_RefAllocWithRefCounter< DefAlloc >`.

7.189.3.2 virtual uint32 OsclRefCounter::getCount () [pure virtual]

Gets the current number of references

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_RefAllocWithRefCounter< DefAlloc >`.

7.189.3.3 virtual void OsclRefCounter::removeRef () [pure virtual]

Delete from reference count

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_RefAllocWithRefCounter< DefAlloc >`.

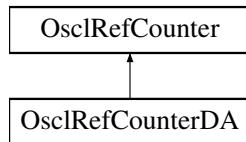
The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

7.190 OsclRefCounterDA Class Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterDA::



Public Methods

- [OsclRefCounterDA \(OsclAny *p, OsclDestructDealloc *dealloc\)](#)
- virtual [~OsclRefCounterDA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

7.190.1 Detailed Description

Implementation of an [OsclRefCounter](#) that uses a dynamically created deallocator.

7.190.2 Constructor & Destructor Documentation

7.190.2.1 OsclRefCounterDA::OsclRefCounterDA ([OsclAny *p](#), [OsclDestructDealloc *dealloc](#)) [inline]

Constructor Takes a pointer to the buffer to track, and a pointer to the deallocator object which will be used to delete the buffer.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to `delete()`.

Parameters:

p pointer to the buffer to track

dealloc pointer to the deallocator to use when deleting the buffer

7.190.2.2 virtual OsclRefCounterDA::~OsclRefCounterDA () [inline, virtual]

Destructor empty

7.190.3 Member Function Documentation

7.190.3.1 void OsclRefCounterDA::addRef () [inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

7.190.3.2 uint32 OsclRefCounterDA::getCount () [inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

7.190.3.3 void OsclRefCounterDA::removeRef () [inline, virtual]

Remove from the reference count

Implements [OsclRefCounter](#).

The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

7.191 OsclRefCounterMemFrag Class Reference

```
#include <oscl_refcounter_memfrag.h>
```

Public Methods

- [OsclRefCounterMemFrag \(OsclMemoryFragment &m, OsclRefCounter *r, uint32 in_capacity\)](#)
- [OsclRefCounterMemFrag \(const OsclRefCounterMemFrag &x\)](#)
- [OsclRefCounterMemFrag \(\)](#)
- [OsclRefCounterMemFrag & operator= \(const OsclRefCounterMemFrag &x\)](#)
- [~OsclRefCounterMemFrag \(\)](#)
- [OsclRefCounter * getRefCounter \(\)](#)
- [OsclMemoryFragment & getMemFrag \(\)](#)
- [OsclAny * getMemFragPtr \(\)](#)
- [uint32 getMemFragSize \(\)](#)
- [uint32 getCapacity \(\)](#)
- [uint32 getCount \(\)](#)

7.191.1 Detailed Description

Class to contain a memory fragment with it's associated reference counter.

7.191.2 Constructor & Destructor Documentation

7.191.2.1 OsclRefCounterMemFrag::OsclRefCounterMemFrag ([OsclMemoryFragment & m](#), [OsclRefCounter * r](#), [uint32 in_capacity](#)) [inline]

Constructor. A valid memory fragment and reference counter are required as input. The memory fragment structure will be copied locally.

Parameters:

- m* reference to memory fragment
- r* pointer to the reference counter associated with the memory fragment.

7.191.2.2 OsclRefCounterMemFrag::OsclRefCounterMemFrag ([const OsclRefCounterMemFrag & x](#)) [inline]

Copy constructor.

7.191.2.3 OsclRefCounterMemFrag::OsclRefCounterMemFrag () [inline]

Default constructor.

7.191.2.4 OsclRefCounterMemFrag::~OsclRefCounterMemFrag () [inline]

Destructor. Removes this object's reference from the reference counter. The reference counter will not be deleted. The reference counter is designed to self-delete when it's reference count reaches 0.

7.191.3 Member Function Documentation

7.191.3.1 **uint32 OsclRefCounterMemFrag::getCapacity () [inline]**

Returns the capacity of the memory fragment

Returns:

7.191.3.2 **uint32 OsclRefCounterMemFrag::getCount () [inline]**

Returns the reference counter's current count.

7.191.3.3 **OsclMemoryFragment& OsclRefCounterMemFrag::getMemFrag () [inline]**

Returns a reference to the contained memory fragment structure.

7.191.3.4 **OsclAny* OsclRefCounterMemFrag::getMemFragPtr () [inline]**

Returns a pointer to the memory fragment data.

7.191.3.5 **uint32 OsclRefCounterMemFrag::getMemFragSize () [inline]**

Returns the size of the memory fragment data which equals its filled size.

Returns:

7.191.3.6 **OsclRefCounter* OsclRefCounterMemFrag::getRefCounter () [inline]**

Returns a pointer to the contained reference counter object

7.191.3.7 **OsclRefCounterMemFrag& OsclRefCounterMemFrag::operator= (const OsclRefCounterMemFrag & x) [inline]**

Assignment Operator

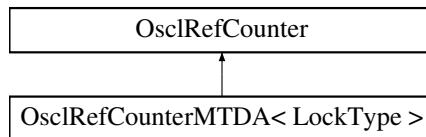
The documentation for this class was generated from the following file:

- [oscl_refcounter_memfrag.h](#)

7.192 OsclRefCounterMTDA< LockType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterMTDA< LockType >::



Public Methods

- [OsclRefCounterMTDA \(OsclAny *p, OsclDestructDealloc *dealloc\)](#)
- virtual [~OsclRefCounterMTDA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

7.192.1 Detailed Description

template<class LockType> class OsclRefCounterMTDA< LockType >

Implementation of [OsclRefCounterDA](#) for multi-threaded use. A templated lock class must be specified.

7.192.2 Constructor & Destructor Documentation

7.192.2.1 template<class LockType> OsclRefCounterMTDA< LockType >::OsclRefCounterMTDA (OsclAny *p, OsclDestructDealloc *dealloc) [inline]

Constructor Takes a pointer to the buffer to track, and a pointer to the deallocator object which will be used to delete the buffer.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to [delete\(\)](#).

Parameters:

p pointer to the buffer to track

dealloc pointer to the deallocator to use when deleting the buffer

7.192.2.2 template<class LockType> virtual OsclRefCounterMTDA< LockType >::~OsclRefCounterMTDA () [inline, virtual]

Destructor empty

7.192.3 Member Function Documentation

**7.192.3.1 template<class LockType> void OsclRefCounterMTDA< LockType >::addRef ()
[inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**7.192.3.2 template<class LockType> uint32 OsclRefCounterMTDA< LockType >::getCount ()
[inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**7.192.3.3 template<class LockType> void OsclRefCounterMTDA< LockType >::removeRef ()
[inline, virtual]**

Remove from the reference count

Implements [OsclRefCounter](#).

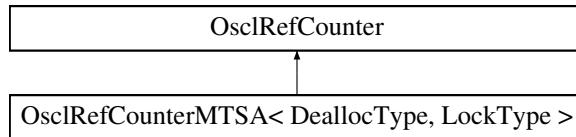
The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

7.193 OsclRefCounterMTSA< DeallocType, LockType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterMTSA< DeallocType, LockType >::



Public Methods

- [OsclRefCounterMTSA \(OsclAny *p\)](#)
- virtual [~OsclRefCounterMTSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

7.193.1 Detailed Description

```
template<class DeallocType, class LockType> class OsclRefCounterMTSA< DeallocType, LockType >
```

Implementation of [OsclRefCounterSA](#) for multi-threaded use. A templated lock class must be specified.

7.193.2 Constructor & Destructor Documentation

7.193.2.1 template<class DeallocType, class LockType> OsclRefCounterMTSA< DeallocType, LockType >::OsclRefCounterMTSA (OsclAny * p) [inline]

Constructor Takes a pointer to the buffer to track.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to `delete()`.

Parameters:

p pointer to the buffer to track

7.193.2.2 template<class DeallocType, class LockType> virtual OsclRefCounterMTSA< DeallocType, LockType >::~OsclRefCounterMTSA () [inline, virtual]

Destructor empty

7.193.3 Member Function Documentation

7.193.3.1 template<class DeallocType, class LockType> void OsclRefCounterMTSA< DeallocType, LockType >::addRef () [inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

7.193.3.2 template<class DeallocType, class LockType> uint32 OsclRefCounterMTSA< DeallocType, LockType >::getCount () [inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

7.193.3.3 template<class DeallocType, class LockType> void OsclRefCounterMTSA< DeallocType, LockType >::removeRef () [inline, virtual]

Remove from the reference count

Implements [OsclRefCounter](#).

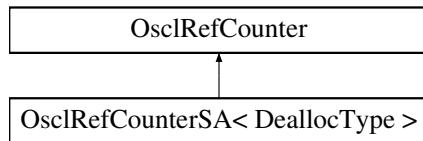
The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

7.194 OsclRefCounterSA< DeallocType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterSA< DeallocType >::



Public Methods

- [OsclRefCounterSA \(OsclAny *p\)](#)
- virtual [~OsclRefCounterSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

7.194.1 Detailed Description

template<class DeallocType> class OsclRefCounterSA< DeallocType >

Implementation of an [OsclRefCounter](#) that uses a statically created deallocator.

7.194.2 Constructor & Destructor Documentation

7.194.2.1 template<class DeallocType> OsclRefCounterSA< DeallocType >::OsclRefCounterSA (OsclAny *p) [inline]

Constructor Takes a pointer to the buffer to track.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to [delete\(\)](#).

Parameters:

p pointer to the buffer to track

7.194.2.2 template<class DeallocType> virtual OsclRefCounterSA< DeallocType >::~OsclRefCounterSA () [inline, virtual]

Destructor empty

7.194.3 Member Function Documentation

7.194.3.1 template<class DeallocType> void OsclRefCounterSA< DeallocType >::addRef () [inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

7.194.3.2 template<class DeallocType> uint32 OsclRefCounterSA< DeallocType >::getCount () [inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

7.194.3.3 template<class DeallocType> void OsclRefCounterSA< DeallocType >::removeRef () [inline, virtual]

Remove from the reference count

Implements [OsclRefCounter](#).

The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

7.195 OsclRegistryAccessClient Class Reference

```
#include <oscl_registry_access_client.h>
```

Public Methods

- OSCL_IMPORT_REF OsclRegistryAccessClient ()
- OSCL_IMPORT_REF ~OsclRegistryAccessClient ()
- OSCL_IMPORT_REF int32 Connect ()
- OSCL_IMPORT_REF OsclComponentFactory GetFactory (OSCL_String &aComponent)
- OSCL_IMPORT_REF void GetFactories (OSCL_String &aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &aVec)
- OSCL_IMPORT_REF void Close ()

7.195.1 Constructor & Destructor Documentation

7.195.1.1 OSCL_IMPORT_REF OsclRegistryAccessClient::OsclRegistryAccessClient ()

7.195.1.2 OSCL_IMPORT_REF OsclRegistryAccessClient::~OsclRegistryAccessClient ()

7.195.2 Member Function Documentation

7.195.2.1 OSCL_IMPORT_REF void OsclRegistryAccessClient::Close ()

Close and cleanup session.

7.195.2.2 OSCL_IMPORT_REF int32 OsclRegistryAccessClient::Connect ()

Create a session.

Returns:

OsclErrNone on success.

7.195.2.3 OSCL_IMPORT_REF void OsclRegistryAccessClient::GetFactories (OSCL_String & aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > & aVec)

Get all factories for a given registry type.

Parameters:

aRegistry: registry MIME type

aVec: output component factory + mimestring vector.

7.195.2.4 OSCL_IMPORT_REF OsclComponentFactory OsclRegistryAccessClient::GetFactory (OSCL_String & aComponent)

Lookup a factory by registry and component mime type.

Parameters:

aComponent: registry+component MIME type

Returns:

Factory. Factory will be NULL if not found.

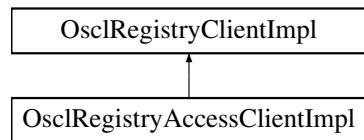
The documentation for this class was generated from the following file:

- [oscl_registry_access_client.h](#)

7.196 OsclRegistryAccessClientImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryAccessClientImpl::



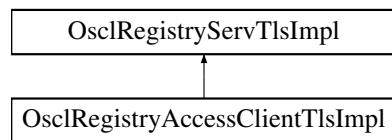
The documentation for this class was generated from the following file:

- [oscl_registry_client_impl.h](#)

7.197 OsclRegistryAccessClientTlsImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryAccessClientTlsImpl::



The documentation for this class was generated from the following file:

- [oscl_registry_client_impl.h](#)

7.198 OsclRegistryAccessElement Class Reference

```
#include <oscl_registry_types.h>
```

Data Fields

- [OsclComponentFactory](#) iFactory
- [OSCL_HeapString< OsclMemAllocator >](#) iMimeType

7.198.1 Detailed Description

A class used to access the registry data

7.198.2 Field Documentation

7.198.2.1 [OsclComponentFactory](#) OsclRegistryAccessElement::iFactory

7.198.2.2 [OSCL_HeapString<OsclMemAllocator>](#) OsclRegistryAccessElement::iMimeType

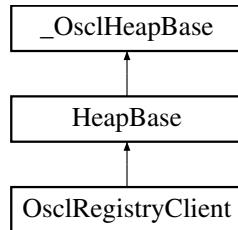
The documentation for this class was generated from the following file:

- [oscl_registry_types.h](#)

7.199 OsclRegistryClient Class Reference

```
#include <oscl_registry_client.h>
```

Inheritance diagram for OsclRegistryClient::



Public Methods

- OSCL_IMPORT_REF [OsclRegistryClient \(\)](#)
- OSCL_IMPORT_REF [~OsclRegistryClient \(\)](#)
- OSCL_IMPORT_REF int32 [Connect \(bool aPerThread=false\)](#)
- OSCL_IMPORT_REF int32 [Register \(OSCL_String &aComponentID, OsclComponentFactory aFactory\)](#)
- OSCL_IMPORT_REF int32 [UnRegister \(OSCL_String &aComponentID\)](#)
- OSCL_IMPORT_REF void [Close \(\)](#)

7.199.1 Constructor & Destructor Documentation

7.199.1.1 OSCL_IMPORT_REF OsclRegistryClient::OsclRegistryClient ()

7.199.1.2 OSCL_IMPORT_REF OsclRegistryClient::~OsclRegistryClient ()

7.199.2 Member Function Documentation

7.199.2.1 OSCL_IMPORT_REF void OsclRegistryClient::Close ()

Close and cleanup. All components registered in this session are automatically unregistered.

7.199.2.2 OSCL_IMPORT_REF int32 OsclRegistryClient::Connect (bool *aPerThread* = false)

Create a session.

Parameters:

aPerThread: Select per-thread registry instead of global registry.

Returns:

OsclErrNone on success.

**7.199.2.3 OSCL_IMPORT_REF int32 OsclRegistryClient::Register ([OSCL_String &](#)
aComponentID, [OsclComponentFactory](#) *aFactory*)**

Register a component factory by registry ID and component ID.

Parameters:

aComponentID: registry + component mime-string.

aFactory: factory function pointer.

aParam: component Create param.

Returns:

OsclErrNone on success.

**7.199.2.4 OSCL_IMPORT_REF int32 OsclRegistryClient::UnRegister ([OSCL_String &](#)
aComponentID)**

Unregister a previously registered component.

Returns:

OsclErrNone on success.

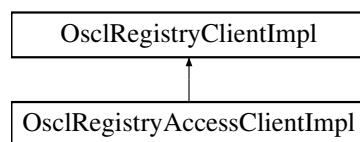
The documentation for this class was generated from the following file:

- [oscl_registry_client.h](#)

7.200 OsclRegistryClientImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryClientImpl:



Protected Methods

- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL_String &, OsclComponentFactory\)](#)
- int32 [UnRegister \(OSCL_String &\)](#)
- [OsclComponentFactory GetFactory \(OSCL_String &\)](#)
- void [GetFactories \(OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &\)](#)

Friends

- class [OsclRegistryClient](#)
- class [OsclRegistryAccessClient](#)

7.200.1 Member Function Documentation

7.200.1.1 **void OsclRegistryClientImpl::Close (void)** [inline, protected]

7.200.1.2 **int32 OsclRegistryClientImpl::Connect ()** [inline, protected]

7.200.1.3 **void OsclRegistryClientImpl::GetFactories (OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &)** [inline, protected]

7.200.1.4 **OsclComponentFactory OsclRegistryClientImpl::GetFactory (OSCL_String &)**
[inline, protected]

7.200.1.5 **int32 OsclRegistryClientImpl::Register (OSCL_String &, OsclComponentFactory)**
[inline, protected]

7.200.1.6 **int32 OsclRegistryClientImpl::UnRegister (OSCL_String &)** [inline,
protected]

7.200.2 Friends And Related Function Documentation

7.200.2.1 **friend class OsclRegistryAccessClient** [friend]

7.200.2.2 **friend class OsclRegistryClient** [friend]

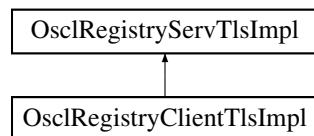
The documentation for this class was generated from the following file:

- [oscl_registry_client_impl.h](#)

7.201 OsclRegistryClientTlsImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryClientTlsImpl::



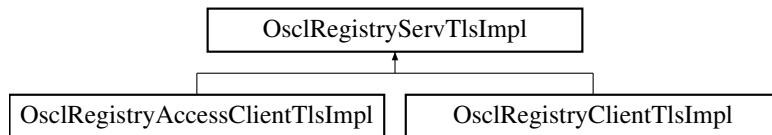
The documentation for this class was generated from the following file:

- [oscl_registry_client_impl.h](#)

7.202 OsclRegistryServTlsImpl Class Reference

```
#include <oscl_registry_serv_impl_tls.h>
```

Inheritance diagram for OsclRegistryServTlsImpl::



Protected Methods

- [OsclRegistryServTlsImpl \(\)](#)
- virtual [~OsclRegistryServTlsImpl \(\)](#)
- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL_String &aComponentID, OsclComponentFactory aFactory\)](#)
- int32 [UnRegister \(OSCL_String &aComponentID\)](#)
- [OsclComponentFactory GetFactory \(OSCL_String &aComponent\)](#)
- void [GetFactories \(OSCL_String &aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &aVec\)](#)

Friends

- class [OsclRegistryClient](#)
- class [OsclRegistryAccessClient](#)

7.202.1 Constructor & Destructor Documentation

7.202.1.1 `OsclRegistryServTlsImpl::OsclRegistryServTlsImpl ()` [protected]

7.202.1.2 `virtual OsclRegistryServTlsImpl::~OsclRegistryServTlsImpl ()` [protected, virtual]

7.202.2 Member Function Documentation

7.202.2.1 `void OsclRegistryServTlsImpl::Close ()` [protected]

7.202.2.2 `int32 OsclRegistryServTlsImpl::Connect ()` [protected]

7.202.2.3 `void OsclRegistryServTlsImpl::GetFactories (OSCL_String & aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > & aVec)` [protected]

7.202.2.4 `OsclComponentFactory OsclRegistryServTlsImpl::GetFactory (OSCL_String & aComponent)` [protected]

7.202.2.5 `int32 OsclRegistryServTlsImpl::Register (OSCL_String & aComponentID, OsclComponentFactory aFactory)` [protected]

7.202.2.6 `int32 OsclRegistryServTlsImpl::UnRegister (OSCL_String & aComponentID)` [protected]

7.202.3 Friends And Related Function Documentation

7.202.3.1 `friend class OsclRegistryAccessClient` [friend]

7.202.3.2 `friend class OsclRegistryClient` [friend]

The documentation for this class was generated from the following file:

- [oscl_registry_serv_impl_tls.h](#)

7.203 OsclScheduler Class Reference

```
#include <oscl_scheduler.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [Init](#) (const char *name, [Oscl_DefAlloc](#) *alloc=NULL, int nreserve=20)
- OSCL_IMPORT_REF void [Cleanup](#) ()

7.203.1 Detailed Description

Per-thread scheduler initialization and cleanup.

7.203.2 Member Function Documentation

7.203.2.1 OSCL_IMPORT_REF void OsclScheduler::Cleanup () [static]

This routine uninstalls and destroys Oscl scheduler for the calling thread.

7.203.2.2 OSCL_IMPORT_REF void OsclScheduler::Init (const char * *name*, [Oscl_DefAlloc](#) * *alloc* = NULL, int *nreserve* = 20) [static]

This routine creates and installs a scheduler in the calling thread.

Parameters:

- name*:** (input param) scheduler name.
- alloc*:** (input param) optional allocator to use for the internal implementation.
- nreserve*:** (input param) optional value for ready queue reserve size.

The documentation for this class was generated from the following file:

- [oscl_scheduler.h](#)

7.204 OsclSchedulerObserver Class Reference

```
#include <oscl_scheduler.h>
```

Public Methods

- virtual void [OsclSchedulerTimerCallback](#) ([OsclAny](#) *aContext, uint32 aDelayMsec)=0
- virtual void [OsclSchedulerReadyCallback](#) ([OsclAny](#) *aContext)=0
- virtual [~OsclSchedulerObserver](#) ()

7.204.1 Detailed Description

OsclSchedulerObserver is an observer class for use when running scheduler in non-blocking mode. The scheduler observer can register for callbacks so it will be notified when it is necessary to run scheduler again. Note: non-blocking mode and scheduler callbacks are not supported on Symbian.

7.204.2 Constructor & Destructor Documentation

7.204.2.1 virtual OsclSchedulerObserver::~OsclSchedulerObserver () [inline, virtual]

7.204.3 Member Function Documentation

**7.204.3.1 virtual void OsclSchedulerObserver::OsclSchedulerReadyCallback ([OsclAny](#) *
aContext)** [pure virtual]

OsclSchedulerReadyCallback is called when the ready queue is updated, meaning an AO is ready to run. Scheduler needs to be run ASAP. Calling context may be any thread, so be careful!

The current observer is cleared before making the callback, so the observer must call RegisterForCallback again if it wants further notifications.

**7.204.3.2 virtual void OsclSchedulerObserver::OsclSchedulerTimerCallback ([OsclAny](#) *
aContext, uint32 *aDelayMsec*)** [pure virtual]

OsclSchedulerTimerCallback is called when the front of the timer queue is updated. This means the minimum delay has changed and scheduler needs to be run again after aDelayMsec. Calling context is in-thread.

The current observer is cleared before making the callback, so the observer must call RegisterForCallback again if it wants further notifications.

The documentation for this class was generated from the following file:

- [oscl_scheduler.h](#)

7.205 OsclScopedLock< LockClass > Class Template Reference

The OsclScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope.

```
#include <oscl_lock_base.h>
```

Public Methods

- [OsclScopedLock \(LockClass &inLock\)](#)
Default constructor Initializes the pointer and takes ownership.
- [~OsclScopedLock \(\)](#)
Destructor.

7.205.1 Detailed Description

template<class LockClass> class OsclScopedLock< LockClass >

The OsclScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope.

The purpose of this class is to provide a way to prevent accidental resource leaks in a class or a method, due to "not remembering to unlock" variables which might lead to deadlock conditions.

7.205.2 Constructor & Destructor Documentation

7.205.2.1 **template<class LockClass> OsclScopedLock< LockClass >::OsclScopedLock (LockClass & inLock) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

7.205.2.2 **template<class LockClass> OsclScopedLock< LockClass >::~OsclScopedLock () [inline]**

Destructor.

The pointer is deleted in case this class still has ownership

The documentation for this class was generated from the following file:

- [oscl_lock_base.h](#)

7.206 OsclSelect Class Reference

```
#include <oscl_init.h>
```

Public Methods

- [OsclSelect \(\)](#)
- [OsclSelect \(Oscl_DefAlloc *erralloc, Oscl_DefAlloc *schedalloc, const char *name, int32 reserve=10, bool heapcheck=false, FILE *output=NULL\)](#)

Data Fields

- bool [iOsclBase](#)
- bool [iOsclMemory](#)
- bool [iOsclErrorTrap](#)
- bool [iOsclLogger](#)
- bool [iOsclScheduler](#)
- [Oscl_DefAlloc * iErrAlloc](#)
- [Oscl_DefAlloc * iSchedulerAlloc](#)
- const char * [iSchedulerName](#)
- int32 [iSchedulerReserve](#)
- bool [iHeapCheck](#)
- FILE * [iOutputFile](#)

7.206.1 Detailed Description

Oscl Module selection and Init/Cleanup options.

7.206.2 Constructor & Destructor Documentation

7.206.2.1 OsclSelect::OsclSelect () [inline]

7.206.2.2 OsclSelect::OsclSelect ([Oscl_DefAlloc](#) * *erralloc*, [Oscl_DefAlloc](#) * *schedalloc*, const char * *name*, int32 *reserve* = 10, bool *heapcheck* = false, FILE * *output* = NULL) [inline]

7.206.3 Field Documentation

7.206.3.1 [Oscl_DefAlloc](#)* OsclSelect::iErrAlloc

7.206.3.2 bool OsclSelect::iHeapCheck

7.206.3.3 bool OsclSelect::iOsclBase

7.206.3.4 bool OsclSelect::iOsclErrorTrap

7.206.3.5 bool OsclSelect::iOsclLogger

7.206.3.6 bool OsclSelect::iOsclMemory

7.206.3.7 bool OsclSelect::iOsclScheduler

7.206.3.8 FILE* OsclSelect::iOutputFile

7.206.3.9 [Oscl_DefAlloc](#)* OsclSelect::iSchedulerAlloc

7.206.3.10 const char* OsclSelect::iSchedulerName

7.206.3.11 int32 OsclSelect::iSchedulerReserve

The documentation for this class was generated from the following file:

- [oscl_init.h](#)

7.207 OsclSemaphore Class Reference

```
#include <oscl_semaphore.h>
```

Public Methods

- OSCL_IMPORT_REF OsclSemaphore ()
- OSCL_IMPORT_REF ~OsclSemaphore ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Create (uint32 initVal=0)
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Close ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Wait ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Wait (uint32 timeout_msec)
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError TryWait ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Signal ()

7.207.1 Detailed Description

Class Semaphore

7.207.2 Constructor & Destructor Documentation

7.207.2.1 OSCL_IMPORT_REF OsclSemaphore::OsclSemaphore ()

Class constructor

7.207.2.2 OSCL_IMPORT_REF OsclSemaphore::~OsclSemaphore ()

Class destructor

7.207.3 Member Function Documentation

7.207.3.1 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::Close ()

Closes the Semaphore

Parameters:

It wont take any parameters

Returns:

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

7.207.3.2 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::Create (uint32 initVal = 0)

Creates the Semaphore

Parameters:

Intialcount

Returns:

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

7.207.3.3 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::Signal ()

Signals that the thread is finished with the Semaphore

Parameters:

It wont take any parameters

Returns:

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

7.207.3.4 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::TryWait ()

Try to acquire semaphore ,if the semaphore is already acquired by another thread, calling thread immediately returns with out blocking

Parameters:

It wont take any parameters

Returns:

Returns SUCCESS_ERROR if the semaphore was acquired, SEM_LOCKED_ERROR if the semaphore cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT_IMPLEMENTED.

7.207.3.5 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::Wait (uint32 timeout_msec)

Makes the thread to wait on the Semaphore, with a timeout.

Parameters:

timeout in milliseconds.

Returns:

Returns SUCCESS_ERROR if the semaphore was aquired, WAIT_TIMEOUT_ERROR if the timeout expired without acquiring the semaphore, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT_IMPLEMENTED.

7.207.3.6 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::Wait ()

Makes the thread to wait on the Semaphore

Parameters:

It wont take any parameters

Returns:

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

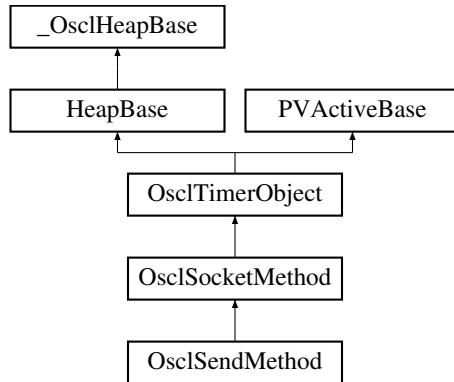
The documentation for this class was generated from the following file:

- [oscl_semaphore.h](#)

7.208 OsclSendMethod Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OsclSendMethod::



Public Methods

- [~OsclSendMethod \(\)](#)
- [TPVSocketEvent Send \(const uint8 *aPtr, uint32 aLen, int32 aTimeout\)](#)
- [uint8 * GetSendData \(int32 *aLength\)](#)
- [OsclSendRequest * SendRequest \(\)](#)

Static Public Methods

- [OsclSendMethod * NewL \(OsclIPSocketI &c\)](#)

7.208.1 Constructor & Destructor Documentation

7.208.1.1 OsclSendMethod::~OsclSendMethod ()

7.208.2 Member Function Documentation

7.208.2.1 uint8* OsclSendMethod::GetSendData (int32 * aLength)

7.208.2.2 OsclSendMethod* OsclSendMethod::NewL (OsclIPSocketI & c) [static]

7.208.2.3 TPVSocketEvent OsclSendMethod::Send (const uint8 *& aPtr, uint32 aLen, int32 aTimeout)

7.208.2.4 OsclSendRequest* OsclSendMethod::SendRequest () [inline]

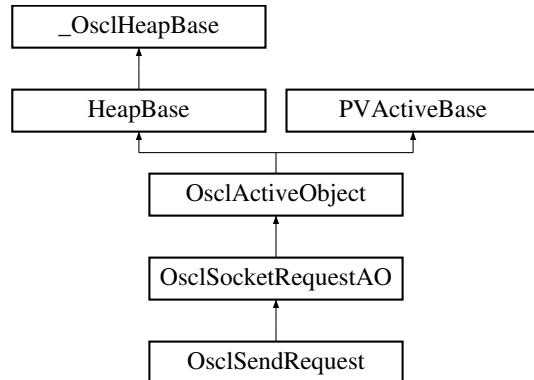
The documentation for this class was generated from the following file:

- [oscl_socket_send.h](#)

7.209 OsclSendRequest Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OsclSendRequest::



Public Methods

- [OsclSendRequest \(OsclSocketMethod &c\)](#)
- void [Send \(const uint8 *&aPtr, uint32 aLen\)](#)
- void [Success \(\)](#)
- uint8 * [GetSendData \(int32 *aLength\)](#)

7.209.1 Constructor & Destructor Documentation

7.209.1.1 OsclSendRequest::OsclSendRequest (OsclSocketMethod & c) [inline]

7.209.2 Member Function Documentation

7.209.2.1 uint8* OsclSendRequest::GetSendData (int32 * aLength)

7.209.2.2 void OsclSendRequest::Send (const uint8 *& aPtr, uint32 aLen)

7.209.2.3 void OsclSendRequest::Success () [virtual]

Reimplemented from [OsclSocketRequestAO](#).

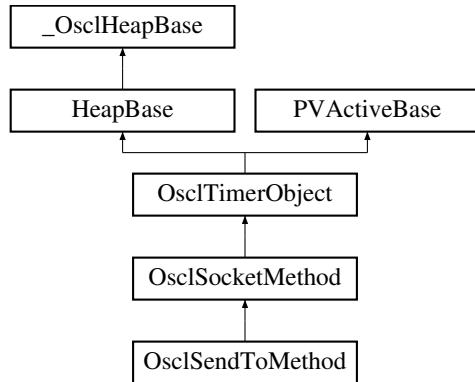
The documentation for this class was generated from the following file:

- [oscl_socket_send.h](#)

7.210 OsclSendToMethod Class Reference

```
#include <oscl_socket_send_to.h>
```

Inheritance diagram for OsclSendToMethod:::



Public Methods

- [~OsclSendToMethod \(\)](#)
- [TPVSocketEvent SendTo \(const uint8 *&aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [uint8 * GetSendData \(int32 *aLength\)](#)
- [OsclSendToRequest * SendToRequest \(\)](#)

Static Public Methods

- [OsclSendToMethod * NewL \(OsclIPSocketI &c\)](#)

7.210.1 Constructor & Destructor Documentation

7.210.1.1 OsclSendToMethod::~OsclSendToMethod ()

7.210.2 Member Function Documentation

7.210.2.1 uint8* OsclSendToMethod::GetSendData (int32 * aLength)

7.210.2.2 OsclSendToMethod* OsclSendToMethod::NewL (OsclIPSocketI & c) [static]

7.210.2.3 TPVSocketEvent OsclSendToMethod::SendTo (const uint8 *& aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeout)

7.210.2.4 OsclSendToRequest* OsclSendToMethod::SendToRequest () [inline]

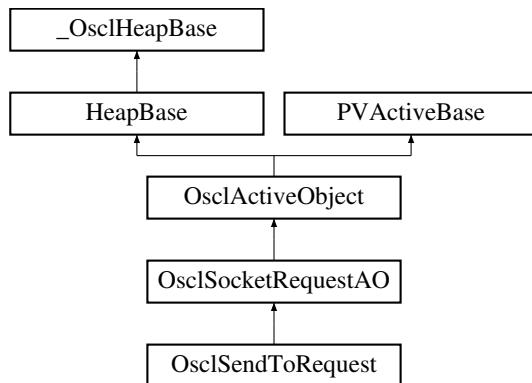
The documentation for this class was generated from the following file:

- [oscl_socket_send_to.h](#)

7.211 OsclSendToRequest Class Reference

```
#include <oscl_socket_send_to.h>
```

Inheritance diagram for OsclSendToRequest::



Public Methods

- [OsclSendToRequest \(OsclSocketMethod &c\)](#)
- void [SendTo \(const uint8 *&aPtr, uint32 aLen, OsclNetworkAddress &aAddress\)](#)
- void [Success \(\)](#)
- uint8 * [GetSendData \(int32 *aLength\)](#)

7.211.1 Detailed Description

This is the AO that interacts with the socket server

7.211.2 Constructor & Destructor Documentation

7.211.2.1 OsclSendToRequest::OsclSendToRequest (OsclSocketMethod & c) [inline]

7.211.3 Member Function Documentation

7.211.3.1 uint8* OsclSendToRequest::GetSendData (int32 * aLength)

7.211.3.2 void OsclSendToRequest::SendTo (const uint8 *& aPtr, uint32 aLen, OsclNetworkAddress & aAddress)

7.211.3.3 void OsclSendToRequest::Success () [virtual]

Reimplemented from [OsclSocketRequestAO](#).

The documentation for this class was generated from the following file:

- [oscl_socket_send_to.h](#)

7.212 OsclSharedPtr< TheClass > Class Template Reference

A parameterized smart pointer class.

```
#include <oscl_shared_ptr.h>
```

Public Methods

- **OsclSharedPtr ()**
Constructor.
- **OsclSharedPtr (TheClass *inClassPtr, OsclRefCounter *in_refcnt)**
Constructor.
- **OsclSharedPtr (const OsclSharedPtr &inSharedPtr)**
Copy constructor.
- **virtual ~OsclSharedPtr ()**
Destructor.
- **TheClass * operator → ()**
• **TheClass & operator * ()**
The indirection operator returns a reference to an object of the parameterized type.
- **operator TheClass * ()**
Casting operator.
- **TheClass * GetRep ()**
Use this function to get a pointer to the wrapped object.
- **OsclRefCounter * GetRefCounter ()**
Get the refcount pointer. This should primarily be used for conversion operations.
- **int get_count ()**
Get a count of how many references to the object exist.
- **void Bind (const OsclSharedPtr &inHandle)**
Use this function to bind an existing OsclSharedPtr to a already-wrapped object.
- **void Bind (TheClass *ptr, OsclRefCounter *in_refcnt)**
Use this function to bind an existing OsclSharedPtr to a new (unwrapped) object.
- **void Unbind ()**
Use this function of unbind an existing OsclSharedPtr.
- **OsclSharedPtr & operator= (const OsclSharedPtr &inSharedPtr)**
Assignment operator.
- **bool operator== (const OsclSharedPtr &b) const**
Test for equality to see if two PVHandles wrap the same object.

7.212.1 Detailed Description

template<class TheClass> class OsclSharedPtr< TheClass >

A parameterized smart pointer class.

7.212.2 Constructor & Destructor Documentation

7.212.2.1 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr () [inline]

Constructor.

7.212.2.2 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr (TheClass * *inClassPtr*, OsclRefCounter * *in_refcnt*) [inline]

Constructor.

Parameters:

inClassPtr A pointer to an instance of the parameterized type that the new OsclSharedPtr will wrap.

7.212.2.3 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr (const OsclSharedPtr< TheClass > & *inSharedPtr*) [inline]

Copy constructor.

7.212.2.4 template<class TheClass> virtual OsclSharedPtr< TheClass >::~OsclSharedPtr () [inline, virtual]

Destructor.

7.212.3 Member Function Documentation

7.212.3.1 template<class TheClass> int OsclSharedPtr< TheClass >::get_count () [inline]

Get a count of how many references to the object exist.

7.212.3.2 template<class TheClass> OsclRefCounter* OsclSharedPtr< TheClass >::GetRefCounter () [inline]

Get the refcount pointer. This should primarily be used for conversion operations.

7.212.3.3 template<class TheClass> TheClass* OsclSharedPtr< TheClass >::GetRep () [inline]

Use this function to get a pointer to the wrapped object.

7.212.3.4 template<class TheClass> TheClass& OsclSharedPtr< TheClass >::operator * () [inline]

The indirection operator returns a reference to an object of the parameterized type.

7.212.3.5 template<class TheClass> OsclSharedPtr< TheClass >::operator TheClass * () [inline]

Casting operator.

7.212.3.6 template<class TheClass> TheClass* OsclSharedPtr< TheClass >::operator -> () [inline]

The dereferencing operator returns a pointer to the parameterized type and can be used to access member elements of TheClass.

7.212.3.7 template<class TheClass> OsclSharedPtr& OsclSharedPtr< TheClass >::operator=(const OsclSharedPtr< TheClass > & *inSharedPtr*) [inline]

Assignment operator.

7.212.3.8 template<class TheClass> void OsclSharedPtr< TheClass >::Unbind () [inline]

Use this function of unbind an existing OsclSharedPtr.

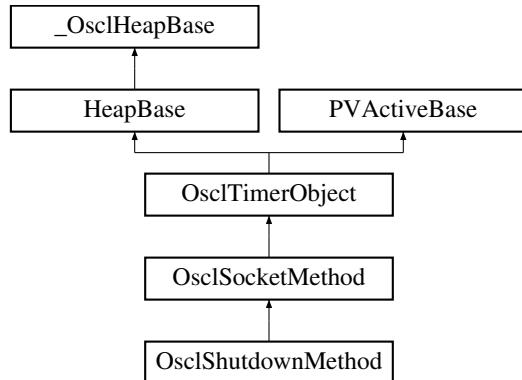
The documentation for this class was generated from the following file:

- [oscl_shared_ptr.h](#)

7.213 OsclShutdownMethod Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OsclShutdownMethod::



Public Methods

- [~OsclShutdownMethod \(\)](#)
- [TPVSocketEvent Shutdown \(TPVSocketShutdown aHow, int32 aTimeout\)](#)
- [OsclShutdownRequest * ShutdownRequest \(\)](#)

Static Public Methods

- [OsclShutdownMethod * NewL \(OsclIPSocketI &c\)](#)

7.213.1 Constructor & Destructor Documentation

7.213.1.1 OsclShutdownMethod::~OsclShutdownMethod ()

7.213.2 Member Function Documentation

7.213.2.1 OsclShutdownMethod* OsclShutdownMethod::NewL (OsclIPSocketI &c) [static]

7.213.2.2 TPVSocketEvent OsclShutdownMethod::Shutdown (TPVSocketShutdown aHow, int32 aTimeout)

7.213.2.3 OsclShutdownRequest* OsclShutdownMethod::ShutdownRequest () [inline]

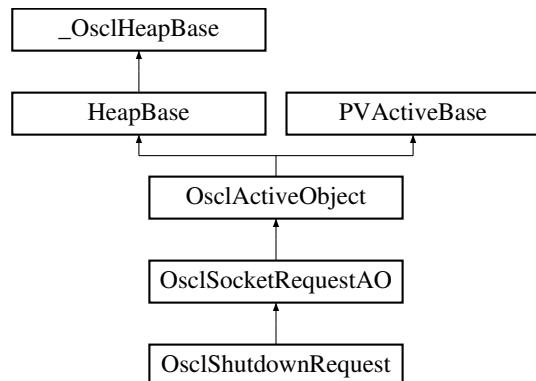
The documentation for this class was generated from the following file:

- [oscl_socket_shutdown.h](#)

7.214 OsclShutdownRequest Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OsclShutdownRequest::



Public Methods

- [OsclShutdownRequest \(OsclSocketMethod &c\)](#)
- [void Shutdown \(TPVSocketShutdown aHow\)](#)

7.214.1 Detailed Description

This is the AO that interacts with the socket server

7.214.2 Constructor & Destructor Documentation

7.214.2.1 OsclShutdownRequest::OsclShutdownRequest ([OsclSocketMethod & c](#)) [inline]

7.214.3 Member Function Documentation

7.214.3.1 void OsclShutdownRequest::Shutdown ([TPVSocketShutdown aHow](#))

The documentation for this class was generated from the following file:

- [oscl_socket_shutdown.h](#)

7.215 OsclSingleton< T, ID, Registry > Class Template Reference

```
#include <oscl_singleton.h>
```

Public Methods

- `OsclSingleton ()`
- `~OsclSingleton ()`
- `T & operator * () const`

The indirection operator () accesses a value indirectly, through a pointer.*

- `T * operator -> () const`

The indirection operator (->) accesses a value indirectly, through a pointer.

- `bool set ()`

set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

Protected Attributes

- `T * _Ptr`

```
template<class T, uint32 ID, class Registry = OsclSingletonRegistry> class OsclSingleton< T, ID, Registry >
```

7.215.1 Constructor & Destructor Documentation

7.215.1.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> OsclSingleton< T, ID, Registry >::OsclSingleton () [inline]

7.215.1.2 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> OsclSingleton< T, ID, Registry >::~OsclSingleton () [inline]

7.215.2 Member Function Documentation

7.215.2.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> T& OsclSingleton< T, ID, Registry >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclSingleton can be used like the regular pointer that it was initialized with.

7.215.2.2 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> T* OsclSingleton< T, ID, Registry >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclSingleton can be used like the regular pointer that it was initialized with.

7.215.2.3 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> bool OsclSingleton< T, ID, Registry >::set () [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

7.215.3 Field Documentation

7.215.3.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> T* OsclSingleton< T, ID, Registry >::_Ptr [protected]

The documentation for this class was generated from the following file:

- [oscl_singleton.h](#)

7.216 OsclSingletonRegistry Class Reference

```
#include <oscl_singleton.h>
```

Static Public Methods

- OSCL_IMPORT_REF [OsclAny](#) * getInstance (uint32 ID, int32 &error)
- OSCL_IMPORT_REF void registerInstance ([OsclAny](#) *ptr, uint32 ID, int32 &error)
- OSCL_IMPORT_REF [OsclAny](#) * lockAndGetInstance (uint32 ID, int32 &error)
- OSCL_IMPORT_REF void registerInstanceAndUnlock ([OsclAny](#) *ptr, uint32 ID, int32 &error)

Friends

- class [OsclBase](#)

7.216.1 Member Function Documentation

7.216.1.1 OSCL_IMPORT_REF [OsclAny](#)* OsclSingletonRegistry::getInstance (uint32 *ID*, int32 & *error*) [static]

7.216.1.2 OSCL_IMPORT_REF [OsclAny](#)* OsclSingletonRegistry::lockAndGetInstance (uint32 *ID*, int32 & *error*) [static]

7.216.1.3 OSCL_IMPORT_REF void OsclSingletonRegistry::registerInstance ([OsclAny](#) * *ptr*, uint32 *ID*, int32 & *error*) [static]

7.216.1.4 OSCL_IMPORT_REF void OsclSingletonRegistry::registerInstanceAndUnlock ([OsclAny](#) * *ptr*, uint32 *ID*, int32 & *error*) [static]

7.216.2 Friends And Related Function Documentation

7.216.2.1 friend class OsclBase [friend]

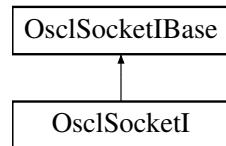
The documentation for this class was generated from the following file:

- [oscl_singleton.h](#)

7.217 OsclSocketI Class Reference

```
#include <oscl_socket_imp_pv.h>
```

Inheritance diagram for OsclSocketI::



Public Methods

- `~OsclSocketI ()`
- `int32 Open (OsclSocketServI &aServer, uint32 addrFamily, uint32 sockType, uint32 protocol)`
- `int32 Open (OsclSocketServI &aServer)`
- `int32 Bind (OsclNetworkAddress &anAddr)`
- `int32 Join (OsclNetworkAddress &anAddr)`
- `int32 Close ()`
- `int32 Listen (uint32 qSize)`
- `int32 SetRecvBufferSize (uint32 size)`
- `void Connect (ConnectParam &, OsclSocketRequestAO &)`
- `void Accept (AcceptParam &, OsclSocketRequestAO &)`
- `void Shutdown (ShutdownParam &, OsclSocketRequestAO &)`
- `void Send (SendParam &, OsclSocketRequestAO &)`
- `void SendSuccess (SendParam &)`
- `void SendTo (SendToParam &, OsclSocketRequestAO &)`
- `void SendToSuccess (SendToParam &)`
- `void Recv (RecvParam &, OsclSocketRequestAO &)`
- `void RecvSuccess (RecvParam &)`
- `void RecvFrom (RecvFromParam &, OsclSocketRequestAO &)`
- `void RecvFromSuccess (RecvFromParam &)`
- `TOsclSocket Socket ()`
- `void ProcessConnect (OsclSocketServRequestQElem *)`
- `void ProcessShutdown (OsclSocketServRequestQElem *)`
- `void ProcessAccept (OsclSocketServRequestQElem *)`
- `void ProcessSendTo (OsclSocketServRequestQElem *)`
- `void ProcessRecvFrom (OsclSocketServRequestQElem *)`
- `void ProcessSend (OsclSocketServRequestQElem *)`
- `void ProcessRecv (OsclSocketServRequestQElem *)`
- `PVLogger * Logger ()`

Static Public Methods

- `OsclSocketI * NewL (Oscl_DefAlloc &a)`
- `bool MakeAddr (OsclNetworkAddress &in, TOsclSockAddr &addr)`
- `void MakeAddr (TOsclSockAddr &in, OsclNetworkAddress &addr)`

Friends

- class [OsclAcceptRequest](#)
- class [OsclConnectRequest](#)
- class [OsclRecvRequest](#)
- class [OsclRecvFromRequest](#)
- class [OsclSendRequest](#)
- class [OsclSendToRequest](#)
- class [OsclShutdownRequest](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

7.217.1 Detailed Description

Socket implementation class

7.217.2 Constructor & Destructor Documentation

7.217.2.1 [OsclSocketI::~OsclSocketI \(\)](#)

7.217.3 Member Function Documentation

7.217.3.1 [void OsclSocketI::Accept \(\[AcceptParam\]\(#\) &, \[OsclSocketRequestAO\]\(#\) &\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

7.217.3.2 [int32 OsclSocketI::Bind \(\[OsclNetworkAddress\]\(#\) & *anAddr*\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

7.217.3.3 [int32 OsclSocketI::Close \(\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

7.217.3.4 [void OsclSocketI::Connect \(\[ConnectParam\]\(#\) &, \[OsclSocketRequestAO\]\(#\) &\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

7.217.3.5 [int32 OsclSocketI::Join \(\[OsclNetworkAddress\]\(#\) & *anAddr*\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

7.217.3.6 [int32 OsclSocketI::Listen \(uint32 *qSize*\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

7.217.3.7 **PVLogger*** OsclSocketI::Logger () [inline]

7.217.3.8 **void** OsclSocketI::MakeAddr (**TOsclSockAddr** & *in*, **OsclNetworkAddress** & *addr*)
[static]

7.217.3.9 **bool** OsclSocketI::MakeAddr (**OsclNetworkAddress** & *in*, **TOsclSockAddr** & *addr*)
[static]

7.217.3.10 **OsclSocketI*** OsclSocketI::NewL (**Oscl_DefAlloc** & *a*) [static]

7.217.3.11 **int32** OsclSocketI::Open (**OsclSocketServI** & *aServer*) [virtual]

Implements [OsclSocketIBase](#).

7.217.3.12 **int32** OsclSocketI::Open (**OsclSocketServI** & *aServer*, **uint32** *addrFamily*, **uint32**
sockType, **uint32** *protocol*) [virtual]

Implements [OsclSocketIBase](#).

7.217.3.13 **void** OsclSocketI::ProcessAccept (**OsclSocketServRequestQElem** *)

7.217.3.14 **void** OsclSocketI::ProcessConnect (**OsclSocketServRequestQElem** *)

7.217.3.15 **void** OsclSocketI::ProcessRecv (**OsclSocketServRequestQElem** *)

7.217.3.16 **void** OsclSocketI::ProcessRecvFrom (**OsclSocketServRequestQElem** *)

7.217.3.17 **void** OsclSocketI::ProcessSend (**OsclSocketServRequestQElem** *)

7.217.3.18 **void** OsclSocketI::ProcessSendTo (**OsclSocketServRequestQElem** *)

7.217.3.19 **void** OsclSocketI::ProcessShutdown (**OsclSocketServRequestQElem** *)

7.217.3.20 **void** OsclSocketI::Recv (**RecvParam** &, **OsclSocketRequestAO** &) [virtual]

Implements [OsclSocketIBase](#).

7.217.3.21 **void** OsclSocketI::RecvFrom (**RecvFromParam** &, **OsclSocketRequestAO** &)
[virtual]

Implements [OsclSocketIBase](#).

7.217.3.22 **void** OsclSocketI::RecvFromSuccess (**RecvFromParam** &) [virtual]

Implements [OsclSocketIBase](#).

7.217.3.23 **void** OsclSocketI::RecvSuccess (**RecvParam** &) [virtual]

Implements [OsclSocketIBase](#).

7.217.3.24 void OsclSocketI::Send ([SendParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

7.217.3.25 void OsclSocketI::SendSuccess ([SendParam](#) &) [virtual]

Implements [OsclSocketIBase](#).

7.217.3.26 void OsclSocketI::SendTo ([SendToParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

7.217.3.27 void OsclSocketI::SendToSuccess ([SendToParam](#) &) [virtual]

Implements [OsclSocketIBase](#).

7.217.3.28 int32 OsclSocketI::SetRecvBufferSize (uint32 *size*)

7.217.3.29 void OsclSocketI::Shutdown ([ShutdownParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

7.217.3.30 [TOsclSocket](#) OsclSocketI::Socket () [inline]

7.217.4 Friends And Related Function Documentation

7.217.4.1 friend class OsclAcceptRequest [friend]

7.217.4.2 friend class OsclConnectRequest [friend]

7.217.4.3 friend class OsclRecvFromRequest [friend]

7.217.4.4 friend class OsclRecvRequest [friend]

7.217.4.5 friend class OsclSendRequest [friend]

7.217.4.6 friend class OsclSendToRequest [friend]

7.217.4.7 friend class OsclShutdownRequest [friend]

7.217.4.8 friend class OsclTCPSocket [friend]

Reimplemented from [OsclSocketIBase](#).

7.217.4.9 friend class OsclUDPSocket [friend]

Reimplemented from [OsclSocketIBase](#).

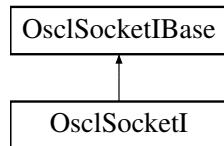
The documentation for this class was generated from the following file:

- [oscl_socket_imp_pv.h](#)

7.218 OsclSocketIBase Class Reference

```
#include <oscl_socket_imp_base.h>
```

Inheritance diagram for OsclSocketIBase::



Public Methods

- virtual ~OsclSocketIBase ()
- virtual int32 Open (OsclSocketServI &aServer, uint32 addrFamily, uint32 sockType, uint32 proto-col)=0
- virtual int32 Open (OsclSocketServI &aServer)=0
- virtual int32 Bind (OsclNetworkAddress &anAddr)=0
- virtual int32 Join (OsclNetworkAddress &anAddr)=0
- virtual int32 Close ()=0
- virtual int32 Listen (uint32 qSize)=0
- virtual void Connect (ConnectParam &, OsclSocketRequestAO &)=0
- virtual void Accept (AcceptParam &, OsclSocketRequestAO &)=0
- virtual void Shutdown (ShutdownParam &, OsclSocketRequestAO &)=0
- virtual void Send (SendParam &, OsclSocketRequestAO &)=0
- virtual void SendSuccess (SendParam &)=0
- virtual void SendTo (SendToParam &, OsclSocketRequestAO &)=0
- virtual void SendToSuccess (SendToParam &)=0
- virtual void Recv (RecvParam &, OsclSocketRequestAO &)=0
- virtual void RecvSuccess (RecvParam &)=0
- virtual void RecvFrom (RecvFromParam &, OsclSocketRequestAO &)=0
- virtual void RecvFromSuccess (RecvFromParam &)=0
- virtual void BindAsync (BindParam &, OsclSocketRequestAO &)
- virtual void ListenAsync (ListenParam &, OsclSocketRequestAO &)
- void CancelFxn (TPVSocketFxn)

Static Public Methods

- bool HasAsyncBind ()
- bool HasAsyncListen ()

Protected Methods

- OsclSocketIBase (Oscl_DefAlloc &a)
- virtual void CancelConnect ()=0
- virtual void CancelAccept ()=0
- virtual void CancelShutdown ()=0
- virtual void CancelSend ()=0

- virtual void [CancelSendTo \(\)=0](#)
- virtual void [CancelRecv \(\)=0](#)
- virtual void [CancelRecvFrom \(\)=0](#)
- virtual void [CancelBind \(\)](#)
- virtual void [CancelListen \(\)](#)
- virtual bool [IsOpen \(\)=0](#)

Static Protected Methods

- int [GetShutdown \(TPVSocketShutdown aOsclVal\)](#)

Protected Attributes

- [Oscl_DefAlloc & iAlloc](#)
- [OsclSocketServI * iSocketServ](#)

Friends

- class [OsclSocketRequest](#)
- class [OsclSocketMethod](#)
- class [OsclSocketRequestAO](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

7.218.1 Detailed Description

Socket implementation base class

7.218.2 Constructor & Destructor Documentation

7.218.2.1 virtual OsclSocketIBase::~OsclSocketIBase () [virtual]

7.218.2.2 OsclSocketIBase::OsclSocketIBase ([Oscl_DefAlloc & a](#)) [protected]

7.218.3 Member Function Documentation

7.218.3.1 virtual void OsclSocketIBase::Accept ([AcceptParam &](#), [OsclSocketRequestAO &](#)) [pure virtual]

Implemented in [OsclSocketI](#).

7.218.3.2 virtual int32 OsclSocketIBase::Bind ([OsclNetworkAddress & anAddr](#)) [pure virtual]

Implemented in [OsclSocketI](#).

- 7.218.3.3 **virtual void OsclSocketIBase::BindAsync (BindParam &, OsclSocketRequestAO &)**
[inline, virtual]
- 7.218.3.4 **virtual void OsclSocketIBase::CancelAccept ()** [protected, pure virtual]
- 7.218.3.5 **virtual void OsclSocketIBase::CancelBind ()** [inline, protected, virtual]
- 7.218.3.6 **virtual void OsclSocketIBase::CancelConnect ()** [protected, pure virtual]
- 7.218.3.7 **void OsclSocketIBase::CancelFxn (TPVSocketFxn)**
- 7.218.3.8 **virtual void OsclSocketIBase::CancelListen ()** [inline, protected, virtual]
- 7.218.3.9 **virtual void OsclSocketIBase::CancelRecv ()** [protected, pure virtual]
- 7.218.3.10 **virtual void OsclSocketIBase::CancelRecvFrom ()** [protected, pure virtual]
- 7.218.3.11 **virtual void OsclSocketIBase::CancelSend ()** [protected, pure virtual]
- 7.218.3.12 **virtual void OsclSocketIBase::CancelSendTo ()** [protected, pure virtual]
- 7.218.3.13 **virtual void OsclSocketIBase::CancelShutdown ()** [protected, pure virtual]
- 7.218.3.14 **virtual int32 OsclSocketIBase::Close ()** [pure virtual]

Implemented in [OsclSocketI](#).

- 7.218.3.15 **virtual void OsclSocketIBase::Connect (ConnectParam &, OsclSocketRequestAO &)**
[pure virtual]

Implemented in [OsclSocketI](#).

- 7.218.3.16 **int OsclSocketIBase::GetShutdown (TPVSocketShutdown aOsclVal)** [static,
protected]
- 7.218.3.17 **bool OsclSocketIBase::HasAsyncBind ()** [static]
- 7.218.3.18 **bool OsclSocketIBase::HasAsyncListen ()** [static]
- 7.218.3.19 **virtual bool OsclSocketIBase::IsOpen ()** [protected, pure virtual]
- 7.218.3.20 **virtual int32 OsclSocketIBase::Join (OsclNetworkAddress & anAddr)** [pure
virtual]

Implemented in [OsclSocketI](#).

- 7.218.3.21 **virtual int32 OsclSocketIBase::Listen (uint32 qSize)** [pure virtual]

Implemented in [OsclSocketI](#).

7.218.3.22 `virtual void OsclSocketIBase::ListenAsync (ListenParam &, OsclSocketRequestAO &)`
[`inline`, `virtual`]

7.218.3.23 `virtual int32 OsclSocketIBase::Open (OsclSocketServI & aServer)` [pure
`virtual`]

Implemented in [OsclSocketI](#).

7.218.3.24 `virtual int32 OsclSocketIBase::Open (OsclSocketServI & aServer, uint32 addrFamily,`
`uint32 sockType, uint32 protocol)` [pure `virtual`]

Implemented in [OsclSocketI](#).

7.218.3.25 `virtual void OsclSocketIBase::Recv (RecvParam &, OsclSocketRequestAO &)` [pure
`virtual`]

Implemented in [OsclSocketI](#).

7.218.3.26 `virtual void OsclSocketIBase::RecvFrom (RecvFromParam &, OsclSocketRequestAO &)` [pure
`virtual`]

Implemented in [OsclSocketI](#).

7.218.3.27 `virtual void OsclSocketIBase::RecvFromSuccess (RecvFromParam &)` [pure
`virtual`]

Implemented in [OsclSocketI](#).

7.218.3.28 `virtual void OsclSocketIBase::RecvSuccess (RecvParam &)` [pure `virtual`]

Implemented in [OsclSocketI](#).

7.218.3.29 `virtual void OsclSocketIBase::Send (SendParam &, OsclSocketRequestAO &)` [pure
`virtual`]

Implemented in [OsclSocketI](#).

7.218.3.30 `virtual void OsclSocketIBase::SendSuccess (SendParam &)` [pure `virtual`]

Implemented in [OsclSocketI](#).

7.218.3.31 `virtual void OsclSocketIBase::SendTo (SendToParam &, OsclSocketRequestAO &)`
[pure `virtual`]

Implemented in [OsclSocketI](#).

7.218.3.32 virtual void OsclSocketIBase::SendToSuccess ([SendToParam](#) &) [pure virtual]

Implemented in [OsclSocketI](#).

7.218.3.33 virtual void OsclSocketIBase::Shutdown ([ShutdownParam](#) &, [OsclSocketRequestAO](#) &) [pure virtual]

Implemented in [OsclSocketI](#).

7.218.4 Friends And Related Function Documentation

7.218.4.1 friend class OsclSocketMethod [friend]

7.218.4.2 friend class OsclSocketRequest [friend]

7.218.4.3 friend class OsclSocketRequestAO [friend]

7.218.4.4 friend class OsclTCPSocket [friend]

Reimplemented in [OsclSocketI](#).

7.218.4.5 friend class OsclUDPSocket [friend]

Reimplemented in [OsclSocketI](#).

7.218.5 Field Documentation

7.218.5.1 [Oscl_DefAlloc](#)& OsclSocketIBase::iAlloc [protected]

7.218.5.2 [OsclSocketServI](#)* OsclSocketIBase::iSocketServ [protected]

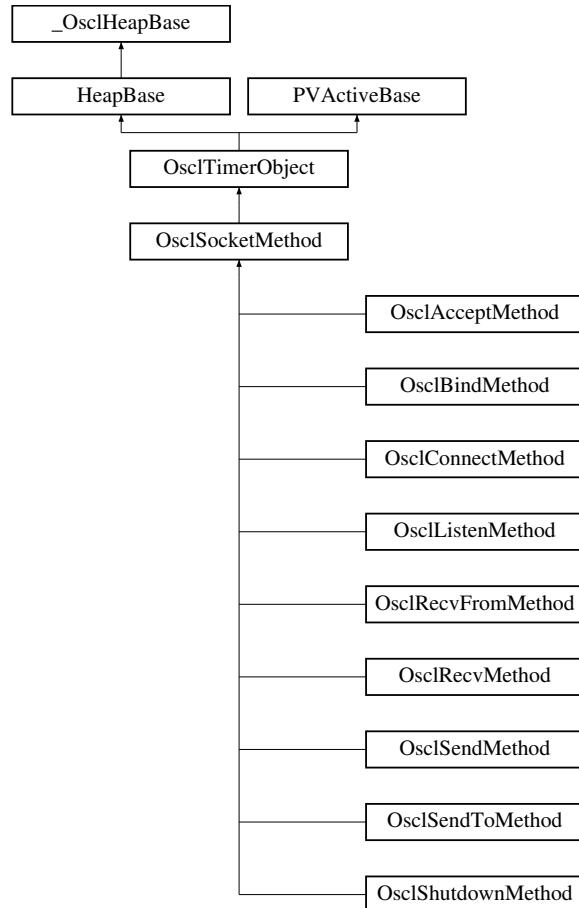
The documentation for this class was generated from the following file:

- [oscl_socket_imp_base.h](#)

7.219 OsclSocketMethod Class Reference

```
#include <oscl_socket_method.h>
```

Inheritance diagram for OsclSocketMethod::



Public Methods

- [OsclSocketMethod \(OsclIPSocketI &aContainer, const char *name, TPVSocketFxn fxn\)](#)
- virtual [~OsclSocketMethod \(\)](#)
- void [Abort \(\)](#)
- void [AbortAll \(\)](#)
- void [CancelMethod \(\)](#)
- [Oscl_DefAlloc & Alloc \(\)](#)

Data Fields

- [OsclIPSocketI & iContainer](#)
- [TPVSocketFxn iSocketFxn](#)

Protected Methods

- void [ConstructL \(OsclSocketRequestAO *aAO\)](#)
- bool [StartMethod \(int32 aTimeoutMsec\)](#)
- void [MethodDone \(\)](#)
- void [Run \(\)](#)

Protected Attributes

- [OsclSocketRequestAO * iSocketRequestAO](#)

7.219.1 Detailed Description

OsclSocketMethod is the base class for all socket methods. Two AOs are required for each socket operation— one to provide a timeout, and one to detect request completion. The OsclSocketMethod class implements the timeout and contains the request completion AO.

7.219.2 Constructor & Destructor Documentation

7.219.2.1 OsclSocketMethod::OsclSocketMethod ([OsclIPSocketI & aContainer](#), [const char * name](#), [TPVSocketFxn ffn](#)) [inline]

7.219.2.2 virtual OsclSocketMethod::~OsclSocketMethod () [inline, virtual]

7.219.3 Member Function Documentation

7.219.3.1 void OsclSocketMethod::Abort () [inline]

7.219.3.2 void OsclSocketMethod::AbortAll () [inline]

7.219.3.3 Oscl_DefAlloc& OsclSocketMethod::Alloc () [inline]

7.219.3.4 void OsclSocketMethod::CancelMethod () [inline]

7.219.3.5 void OsclSocketMethod::ConstructL (OsclSocketRequestAO * aAO) [inline, protected]

7.219.3.6 void OsclSocketMethod::MethodDone () [inline, protected]

7.219.3.7 void OsclSocketMethod::Run () [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request

2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls [ExecError\(\)](#) to handle the leave.

Note that once the active scheduler's [Start\(\)](#) function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

7.219.3.8 bool OsclSocketMethod::StartMethod (int32 *aTimeoutMsec*) [protected]

7.219.4 Field Documentation

7.219.4.1 OsclIPSocketI& OsclSocketMethod::iContainer

7.219.4.2 TPVSocketFxn OsclSocketMethod::iSocketFxn

7.219.4.3 OsclSocketRequestAO* OsclSocketMethod::iSocketRequestAO [protected]

The documentation for this class was generated from the following file:

- [oscl_socket_method.h](#)

7.220 OsclSocketObserver Class Reference

```
#include <oscl_socket_types.h>
```

Public Methods

- virtual OSCL_IMPORT_REF void [HandleSocketEvent](#) (int32 *aId*, [TPVSocketFxn](#) *aFxn*, [TPVSocketEvent](#) *aEvent*, int32 *aError*)=0
- virtual ~[OsclSocketObserver](#) ()

7.220.1 Detailed Description

Socket event observer. The client implements this to get asynchronous command completion.

7.220.2 Constructor & Destructor Documentation

7.220.2.1 virtual [OsclSocketObserver::~OsclSocketObserver](#) () [inline, virtual]

7.220.3 Member Function Documentation

7.220.3.1 virtual OSCL_IMPORT_REF void [OsclSocketObserver::HandleSocketEvent](#) (int32 *aId*, [TPVSocketFxn](#) *aFxn*, [TPVSocketEvent](#) *aEvent*, int32 *aError*) [pure virtual]

Socket Event callback.

Parameters:

aId: The ID that was supplied when the socket was created.

aFxn: Type of socket function call.

aEvent: Function completion event. Will be EPVSocketSuccess, EPVSocketTimeout, or EPVSocketFailure.

aError: When the event is EPVSocketFailure, this may contain a platform-specific error code, or zero if none is available.

The documentation for this class was generated from the following file:

- [oscl_socket_types.h](#)

7.221 OsclSocketRequest Class Reference

```
#include <oscl_socket_request.h>
```

Public Methods

- [OsclSocketRequest \(\)](#)
- [TPVSocketFxn Fxn \(\)](#)
- void [CancelRequest \(\)](#)
- void [Activate \(SocketRequestParam *iParam, OsclSocketRequestAO &a\)](#)
- void [Complete \(OsclSocketServRequestQElem *, int32 aStatus, int32 aSockErr=0\)](#)

Data Fields

- [OsclSocketRequestAO * iSocketRequestAO](#)
- [SocketRequestParam * iParam](#)
- [OsclSocketI * iSocketI](#)

7.221.1 Detailed Description

This class defines the request interface to the PV socket server.

7.221.2 Constructor & Destructor Documentation

7.221.2.1 [OsclSocketRequest::OsclSocketRequest \(\) \[inline\]](#)

7.221.3 Member Function Documentation

7.221.3.1 [void OsclSocketRequest::Activate \(SocketRequestParam * iParam, OsclSocketRequestAO & a\)](#)

7.221.3.2 [void OsclSocketRequest::CancelRequest \(\)](#)

7.221.3.3 [void OsclSocketRequest::Complete \(OsclSocketServRequestQElem *, int32 aStatus, int32 aSockErr = 0\)](#)

7.221.3.4 [TPVSocketFxn OsclSocketRequest::Fxn \(\) \[inline\]](#)

7.221.4 Field Documentation

7.221.4.1 [SocketRequestParam* OsclSocketRequest::iParam](#)

7.221.4.2 [OsclSocketI* OsclSocketRequest::iSocketI](#)

7.221.4.3 [OsclSocketRequestAO* OsclSocketRequest::iSocketRequestAO](#)

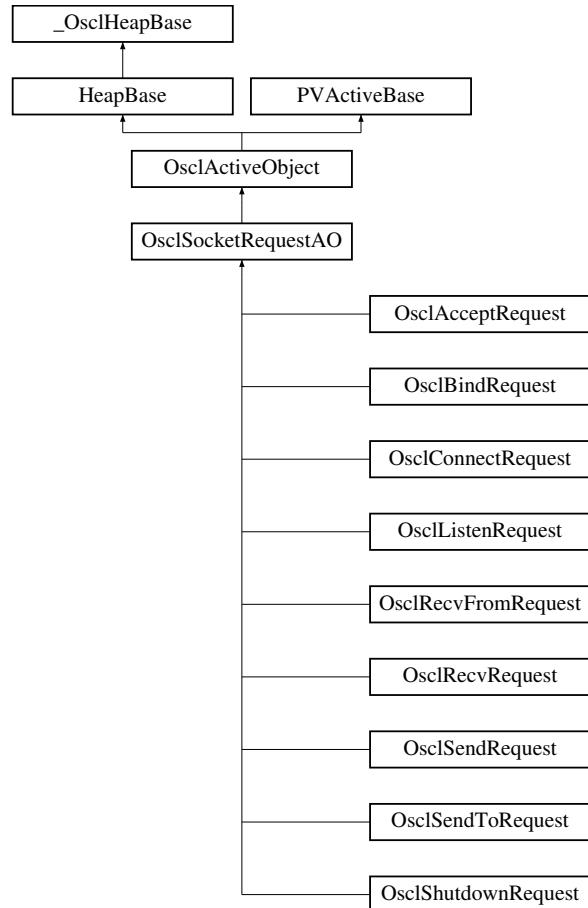
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.222 OsclSocketRequestAO Class Reference

```
#include <oscl_socket_method.h>
```

Inheritance diagram for OsclSocketRequestAO::



Public Methods

- void [ConstructL \(\)](#)

Protected Methods

- [OsclSocketRequestAO \(OsclSocketMethod &aContainer, const char *name\)](#)
- virtual [~OsclSocketRequestAO \(\)](#)
- [OsclAny * NewRequest \(const uint32 size\)](#)
- void [CleanupParam \(bool deallocate=false\)](#)
- void [Abort \(\)](#)
- void [RequestDone \(\)](#)
- int [GetSocketError \(\)](#)
- void [DoCancel \(\)](#)
- void [Run \(\)](#)

- virtual void [Success \(\)](#)
- [OsclSocketI * SocketI \(\)](#)
- [OsclSocketObserver * SocketObserver \(\)](#)
- uint32 [Id \(\)](#)
- [Oscl_DefAlloc & Alloc \(\)](#)

Protected Attributes

- [OsclSocketMethod & iContainer](#)
- int32 [iSocketError](#)
- [SocketRequestParam * iParam](#)
- uint32 [iParamSize](#)

Friends

- class [OsclSocketI](#)
- class [OsclSocketMethod](#)
- class [OsclSocketRequest](#)

7.222.1 Detailed Description

This is the base class for all the AOs that interact with the socket server. This object is contained within an [OsclSocketMethod](#) object

7.222.2 Constructor & Destructor Documentation

7.222.2.1 OsclSocketRequestAO::OsclSocketRequestAO ([OsclSocketMethod & aContainer](#), const char * *name*) [inline, protected]

7.222.2.2 virtual OsclSocketRequestAO::~OsclSocketRequestAO () [inline, protected, virtual]

7.222.3 Member Function Documentation

7.222.3.1 void OsclSocketRequestAO::Abort () [inline, protected]

7.222.3.2 [Oscl_DefAlloc& OsclSocketRequestAO::Alloc \(\) \[inline, protected\]](#)

7.222.3.3 void OsclSocketRequestAO::CleanupParam (bool *deallocate* = false) [protected]

7.222.3.4 void OsclSocketRequestAO::ConstructL () [inline]

7.222.3.5 void OsclSocketRequestAO::DoCancel () [inline, protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Reimplemented from [OsclActiveObject](#).

7.222.3.6 **int OsclSocketRequestAO::GetSocketError ()** [inline, protected]

7.222.3.7 **uint32 OsclSocketRequestAO::Id ()** [inline, protected]

7.222.3.8 **OsclAny* OsclSocketRequestAO::NewRequest (const uint32 size)** [protected]

7.222.3.9 **void OsclSocketRequestAO::RequestDone ()** [inline, protected]

7.222.3.10 **void OsclSocketRequestAO::Run ()** [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's Run() function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

Run() runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's Run() or RunError() functions.

Implements [PVActiveBase](#).

7.222.3.11 **OsclSocketI* OsclSocketRequestAO::SocketI ()** [inline, protected]

7.222.3.12 **OsclSocketObserver* OsclSocketRequestAO::SocketObserver ()** [inline, protected]

7.222.3.13 **virtual void OsclSocketRequestAO::Success ()** [inline, protected, virtual]

Reimplemented in [OsclRecvRequest](#), [OsclRecvFromRequest](#), [OsclSendRequest](#), and [OsclSendToRequest](#).

7.222.4 Friends And Related Function Documentation

7.222.4.1 **friend class OsclSocketI** [friend]

7.222.4.2 **friend class OsclSocketMethod** [friend]

7.222.4.3 **friend class OsclSocketRequest** [friend]

7.222.5 Field Documentation

7.222.5.1 **OsclSocketMethod& OsclSocketRequestAO::iContainer** [protected]

7.222.5.2 **SocketRequestParam* OsclSocketRequestAO::iParam** [protected]

7.222.5.3 **uint32 OsclSocketRequestAO::iParamSize** [protected]

7.222.5.4 **int32 OsclSocketRequestAO::iSocketError** [protected]

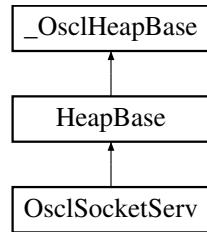
The documentation for this class was generated from the following file:

- [oscl_socket_method.h](#)

7.223 OsclSocketServ Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclSocketServ::



Public Methods

- OSCL_IMPORT_REF ~OsclSocketServ ()
- OSCL_IMPORT_REF int32 Connect (uint32 aMessageSlots=8)
- OSCL_IMPORT_REF void Close (bool aCleanup=true)

Static Public Methods

- OSCL_IMPORT_REF OsclSocketServ * NewL (Oscl_DefAlloc &alloc)

Friends

- class OsclTCPSocket
- class OsclUDPSocket
- class OsclDNS

7.223.1 Constructor & Destructor Documentation

7.223.1.1 OSCL_IMPORT_REF OsclSocketServ::~OsclSocketServ ()

Destructor. The server object must be deleted using the same allocator used in the NewL call.

7.223.2 Member Function Documentation

7.223.2.1 OSCL_IMPORT_REF void OsclSocketServ::Close (bool aCleanup = true)

Close socket server. This is a synchronous method.

Parameters:

aCleanup: cleanup the socket system? the socket cleanup should only be done when all parts of the application are done using sockets.

7.223.2.2 OSCL_IMPORT_REF int32 OsclSocketServ::Connect (uint32 *aMessageSlots* = 8)

Connect to socket server. This is a synchronous method.

Parameters:

Number of message slots.

Returns:

Returns OsclErrNone for success, or a platform-specific code.

**7.223.2.3 OSCL_IMPORT_REF OsclSocketServ* OsclSocketServ::NewL (Oscl_DefAlloc & *alloc*)
[static]**

Create a socket server. May leave if failure.

Parameters:

alloc: Memory allocator.

Returns:

Returns pointer to socket server

7.223.3 Friends And Related Function Documentation**7.223.3.1 friend class OsclDNS [friend]****7.223.3.2 friend class OsclTCPSocket [friend]****7.223.3.3 friend class OsclUDPSocket [friend]**

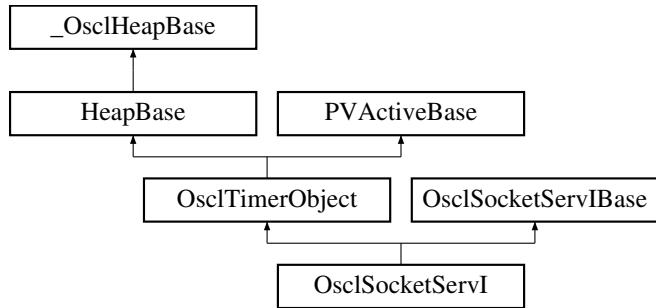
The documentation for this class was generated from the following file:

- [oscl_socket.h](#)

7.224 OsclSocketServI Class Reference

```
#include <oscl_socket_serv_imp_pv.h>
```

Inheritance diagram for OsclSocketServI::



Public Methods

- int32 [Connect](#) (uint32 aMessageSlots)
- void [Close](#) (bool)
- bool [IsServerThread](#) ()

Static Public Methods

- OsclSocketServI * [NewL](#) (Oscl_DefAlloc &a)

Friends

- class [OsclSocketServRequestList](#)
- class [LoopbackSocket](#)
- class [OsclTCPSocketI](#)
- class [OsclUDPSocketI](#)
- class [OsclSocketI](#)
- class [OsclDNSI](#)
- class [OsclSocketRequest](#)
- class [OsclSocketServ](#)

7.224.1 Detailed Description

PV socket server implementation

7.224.2 Member Function Documentation

7.224.2.1 void OsclSocketServI::Close (bool) [virtual]

Implements [OsclSocketServIBase](#).

7.224.2.2 **int32 OsclSocketServI::Connect (uint32 *aMessageSlots*)** [virtual]

Implements [OsclSocketServIBase](#).

7.224.2.3 **bool OsclSocketServI::IsServerThread ()**

7.224.2.4 **OsclSocketServI* OsclSocketServI::NewL (Oscl_DefAlloc & *a*)** [static]

7.224.3 Friends And Related Function Documentation

7.224.3.1 **friend class LoopbackSocket** [friend]

7.224.3.2 **friend class OsclDNSI** [friend]

7.224.3.3 **friend class OsclSocketI** [friend]

7.224.3.4 **friend class OsclSocketRequest** [friend]

7.224.3.5 **friend class OsclSocketServ** [friend]

7.224.3.6 **friend class OsclSocketServRequestList** [friend]

7.224.3.7 **friend class OsclTCPSocketI** [friend]

7.224.3.8 **friend class OsclUDPSocketI** [friend]

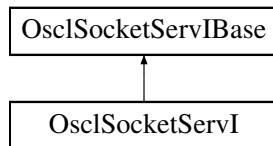
The documentation for this class was generated from the following file:

- [oscl_socket_serv_imp_pv.h](#)

7.225 OsclSocketServIBase Class Reference

```
#include <oscl_socket_serv_imp_base.h>
```

Inheritance diagram for OsclSocketServIBase::



Public Methods

- virtual ~OsclSocketServIBase ()
- virtual int32 [Connect](#) (uint32 aMessageSlots)=0
- virtual void [Close](#) (bool)=0

Data Fields

- [PVLogger * iLogger](#)

Protected Types

- enum [TSocketServState](#) { [ESocketServ_Idle](#), [ESocketServ_Connected](#), [ESocketServ_Error](#) }

Protected Methods

- [OsclSocketServIBase \(Oscl_DefAlloc &a\)](#)
- [TSocketServState State \(\) const](#)
- [bool IsServConnected \(\) const](#)

Protected Attributes

- [Oscl_DefAlloc & iAlloc](#)
- [TSocketServState iServState](#)
- [int iServError](#)

7.225.1 Detailed Description

Base class common to all implementations

7.225.2 Member Enumeration Documentation

7.225.2.1 enum OsclSocketServIBase::TSocketServState [protected]

Enumeration values:

[ESocketServ_Idle](#)

ESocketServ_Connected

ESocketServ_Error

7.225.3 Constructor & Destructor Documentation

7.225.3.1 virtual OsclSocketServIBase::~OsclSocketServIBase () [inline, virtual]

7.225.3.2 OsclSocketServIBase::OsclSocketServIBase ([Oscl_DefAlloc](#) & *a*) [inline, protected]

7.225.4 Member Function Documentation

7.225.4.1 virtual void OsclSocketServIBase::Close (bool) [pure virtual]

Implemented in [OsclSocketServI](#).

7.225.4.2 virtual int32 OsclSocketServIBase::Connect (uint32 *aMessageSlots*) [pure virtual]

Implemented in [OsclSocketServI](#).

7.225.4.3 bool OsclSocketServIBase::IsServConnected () const [inline, protected]

7.225.4.4 [TSocketServState](#) OsclSocketServIBase::State () const [inline, protected]

7.225.5 Field Documentation

7.225.5.1 [Oscl_DefAlloc](#)& OsclSocketServIBase::iAlloc [protected]

7.225.5.2 [PVLogger](#)* OsclSocketServIBase::iLogger

7.225.5.3 int OsclSocketServIBase::iServerError [protected]

7.225.5.4 [TSocketServState](#) OsclSocketServIBase::iServState [protected]

The documentation for this class was generated from the following file:

- [oscl_socket_serv_imp_base.h](#)

7.226 OsclSocketServRequestList Class Reference

```
#include <oscl_socket_serv_imp_reqlist.h>
```

Public Methods

- [OsclSocketServRequestList \(\)](#)
- void [Add \(OsclSocketRequest *\)](#)
- void [StartCancel \(OsclSocketRequest *\)](#)
- void [Open \(OsclSocketServI *s\)](#)
- void [Close \(\)](#)
- void [Wakeup \(\)](#)
- void [WaitOnRequests \(\)](#)
- void [Remove \(OsclSocketServRequestQElem *aElem\)](#)

Friends

- class [OsclSocketServI](#)

7.226.1 Detailed Description

PV socket server request queue

7.226.2 Constructor & Destructor Documentation

7.226.2.1 OsclSocketServRequestList::OsclSocketServRequestList ()

7.226.3 Member Function Documentation

7.226.3.1 void OsclSocketServRequestList::Add ([OsclSocketRequest *](#))

7.226.3.2 void OsclSocketServRequestList::Close ()

7.226.3.3 void OsclSocketServRequestList::Open ([OsclSocketServI * s](#))

7.226.3.4 void OsclSocketServRequestList::Remove ([OsclSocketServRequestQElem * aElem](#)) [inline]

7.226.3.5 void OsclSocketServRequestList::StartCancel ([OsclSocketRequest *](#))

7.226.3.6 void OsclSocketServRequestList::WaitOnRequests ()

7.226.3.7 void OsclSocketServRequestList::Wakeup ()

7.226.4 Friends And Related Function Documentation

7.226.4.1 friend class OsclSocketServI [friend]

The documentation for this class was generated from the following file:

- [oscl_socket_serv_imp_reqlist.h](#)

7.227 OsclSocketServRequestQElem Class Reference

```
#include <oscl_socket_serv_imp_reqlist.h>
```

Public Methods

- [OsclSocketServRequestQElem \(OsclSocketRequest *r\)](#)

Data Fields

- [OsclSocketRequest * iSocketRequest](#)
- [uint8 iSelect](#)
- [bool iCancel](#)

7.227.1 Constructor & Destructor Documentation

7.227.1.1 [OsclSocketServRequestQElem::OsclSocketServRequestQElem \(OsclSocketRequest * r\)](#)
[inline]

7.227.2 Field Documentation

7.227.2.1 [bool OsclSocketServRequestQElem::iCancel](#)

7.227.2.2 [uint8 OsclSocketServRequestQElem::iSelect](#)

7.227.2.3 [OsclSocketRequest* OsclSocketServRequestQElem::iSocketRequest](#)

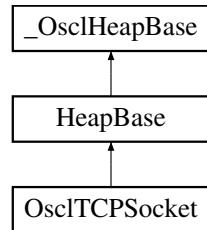
The documentation for this class was generated from the following file:

- [oscl_socket_serv_imp_reqlist.h](#)

7.228 OsclTCPSocket Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclTCPSocket::



Public Methods

- OSCL_IMPORT_REF ~OsclTCPSocket ()
- OSCL_IMPORT_REF int32 Close ()
- OSCL_IMPORT_REF int32 Bind (OsclNetworkAddress &aAddress)
- OSCL_IMPORT_REF TPVSocketEvent BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL_IMPORT_REF void CancelBind ()
- OSCL_IMPORT_REF int32 Listen (int32 aQueueSize)
- OSCL_IMPORT_REF TPVSocketEvent ListenAsync (int32 aQueueSize, int32 aTimeoutMsec=(-1))
- OSCL_IMPORT_REF void CancelListen ()
- OSCL_IMPORT_REF OsclTCPSocket * GetAcceptedSocketL (uint32 aId)
- OSCL_IMPORT_REF uint8 * GetRecvData (int32 *aLength)
- OSCL_IMPORT_REF uint8 * GetSendData (int32 *aLength)
- OSCL_IMPORT_REF TPVSocketEvent Connect (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelConnect ()
- OSCL_IMPORT_REF TPVSocketEvent Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelShutdown ()
- OSCL_IMPORT_REF TPVSocketEvent Accept (int32 aTimeout=-1)
- OSCL_IMPORT_REF void CancelAccept ()
- OSCL_IMPORT_REF TPVSocketEvent Send (const uint8 *aPtr, uint32 aLen, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelSend ()
- OSCL_IMPORT_REF TPVSocketEvent Recv (uint8 *aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelRecv ()

Static Public Methods

- OSCL_IMPORT_REF OsclTCPSocket * NewL (Oscl_DefAlloc &alloc, OsclSocketServ &aServ, OsclSocketObserver *aObserver, uint32 aId)

7.228.1 Detailed Description

The TCP Socket class

7.228.2 Constructor & Destructor Documentation

7.228.2.1 OSCL_IMPORT_REF OsclTCPSocket::~OsclTCPSocket ()

Destructor. The object must be deleted using the same allocator used in the NewL call.

7.228.3 Member Function Documentation

7.228.3.1 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::Accept (int32 *aTimeout* = -1)

Accept incoming connections. This is an asynchronous method.

Parameters:

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

7.228.3.2 OSCL_IMPORT_REF int32 OsclTCPSocket::Bind (OsclNetworkAddress & *aAddress*)

Bind a TCP socket to an address. This is a synchronous method.

Parameters:

aAddress: Bind address.

Returns:

Returns OsclErrNone for success, or a platform-specific error code.

7.228.3.3 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::BindAsync (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = (-1))

Bind a TCP socket to an address. This is an asynchronous method.

Parameters:

aAddress: Bind address.

aTimeoutMsec: Optional timeout. Use a negative value for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

7.228.3.4 OSCL_IMPORT_REF void OsclTCPSocket::CancelAccept ()

Cancel Accept

This method will cancel any pending Accept operation on the current socket, causing the Accept to complete with error EPVSocketCancel. If there is no pending Accept operation, this method will have no effect.

7.228.3.5 OSCL_IMPORT_REF void OsclTCPSocket::CancelBind ()

Cancel Bind

This method will cancel any pending BindAsync operation on the current socket, causing the BindAsync to complete with error EPVSocketCancel. If there is no pending BindAsync operation, this method will have no effect.

7.228.3.6 OSCL_IMPORT_REF void OsclTCPSocket::CancelConnect ()

Cancel Connect

This method will cancel any pending Connect operation on the current socket, causing the Connect to complete with error EPVSocketCancel. If there is no pending Connect operation, this method will have no effect.

7.228.3.7 OSCL_IMPORT_REF void OsclTCPSocket::CancelListen ()

Cancel Async Listen

This method will cancel any pending ListenAsync operation on the current socket, causing the Listen to complete with error EPVSocketCancel. If there is no pending Listen operation, this method will have no effect.

7.228.3.8 OSCL_IMPORT_REF void OsclTCPSocket::CancelRecv ()

Cancel Recv

This method will cancel any pending Recv operation on the current socket, causing the Recv to complete with error EPVSocketCancel. If there is no pending Recv operation, this method will have no effect.

7.228.3.9 OSCL_IMPORT_REF void OsclTCPSocket::CancelSend ()

Cancel Send

This method will cancel any pending Send operation on the current socket, causing the Send to complete with error EPVSocketCancel. If there is no pending Send operation, this method will have no effect.

7.228.3.10 OSCL_IMPORT_REF void OsclTCPSocket::CancelShutdown ()

Cancel Shutdown

This method will cancel any pending Shutdown operation on the current socket, causing the Shutdown to complete with error EPVSocketCancel. If there is no pending Shutdown operation, this method will have no effect.

7.228.3.11 OSCL_IMPORT_REF int32 OsclTCPSocket::Close ()

Close a TCP socket. This is a synchronous method.

Once it is closed a socket cannot be re-opened. Sockets are automatically closed when they are deleted. This method may be used to see any error code returned from the platform's socket close call.

Returns:

Returns OsclErrNone for success, or a platform-specific error code.

**7.228.3.12 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::Connect
(OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1)**

Connect to an address. This is an asynchronous method.

Parameters:

aAddress: a network address.

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

7.228.3.13 OSCL_IMPORT_REF OsclTCPSocket* OsclTCPSocket::GetAcceptedSocketL (uint32 aId)

Retrieve the accept socket after a successful Accept operation. This is a synchronous method.

Parameters:

aId: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

Returns:

Returns pointer to socket, or NULL if error. Note: The caller is responsible for deleting any accepted socket that it retrieves.

7.228.3.14 OSCL_IMPORT_REF uint8* OsclTCPSocket::GetRecvData (int32 * aLength)

Retrieve the received data after a successful Recv operation. This is a synchronous method.

Parameters:

aLength: (output) number of bytes of data received.

Returns:

Returns pointer to received data, or NULL if none.

7.228.3.15 OSCL_IMPORT_REF uint8* OsclTCPSocket::GetSendData (int32 * aLength)

Retrieve the sent data after a successful Send operation. This is a synchronous method.

Parameters:

aLength: (output) number of bytes of data sent.

Returns:

Returns pointer to sent data, or NULL if none.

7.228.3.16 OSCL_IMPORT_REF int32 OsclTCPSocket::Listen (int32 aQueueSize)

Listen. This is a synchronous method.

Parameters:

aQueueSize: Queue size.

Returns:

Returns OsclErrNone for success, or a platform-specific error code.

7.228.3.17 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::ListenAsync (int32 aQueueSize, int32 aTimeoutMsec = (-1))

ListenAsync This is an asynchronous method.

Parameters:

aQueueSize: Queue size.

aTimeoutMsec: Optional timeout. Use a negative value for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

7.228.3.18 OSCL_IMPORT_REF OsclTCPSocket* OsclTCPSocket::NewL (Oscl_DefAlloc & alloc, OsclSocketServ & aServ, OsclSocketObserver * aObserver, uint32 aId) [static]

Create a TCP Socket. May leave if failure.

Parameters:

alloc: Memory allocator.

aServ: Socket server. Must be connected.

aObserver: Socket observer.

aId: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

Returns:

Returns pointer to socket.

7.228.3.19 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::Recv (uint8 * aPtr, uint32 aMaxLen, int32 aTimeoutMsec = -1)

Receive Data. This is an asynchronous method.

Parameters:

aPtr: Buffer for received data.

aMaxLen: Length of buffer.

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

7.228.3.20 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::Send (const uint8 * aPtr, uint32 aLen, int32 aTimeoutMsec = -1)

Send Data. This is an asynchronous method.

Parameters:

aPtr: Data to send.

aLen: Length of data to send.

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

7.228.3.21 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec = -1)

Shutdown a socket. This is an asynchronous method.

Parameters:

aHow: type of shutdown

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

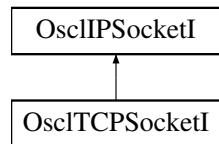
The documentation for this class was generated from the following file:

- [oscl_socket.h](#)

7.229 OsclTCPSocketI Class Reference

```
#include <oscl_tcp_socket.h>
```

Inheritance diagram for OsclTCPSocketI::



Public Methods

- virtual ~OsclTCPSocketI ()
- int32 [Close \(\)](#)
- int32 [Listen \(int aQueueSize\)](#)
- OsclTCPSocketI * [GetAcceptedSocketL \(uint32 aId\)](#)
- uint8 * [GetRecvData \(int32 *aLength\)](#)
- uint8 * [GetSendData \(int32 *aLength\)](#)
- [TPVSocketEvent BindAsync \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelBind \(\)](#)
- [TPVSocketEvent ListenAsync \(uint32 qsize, int32 aTimeoutMsec=-1\)](#)
- void [CancelListen \(\)](#)
- [TPVSocketEvent Connect \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelConnect \(\)](#)
- [TPVSocketEvent Shutdown \(TPVSocketShutdown aHow, int32 aTimeoutMsec=-1\)](#)
- void [CancelShutdown \(\)](#)
- [TPVSocketEvent Accept \(int32 aTimeout=-1\)](#)
- void [CancelAccept \(\)](#)
- [TPVSocketEvent Send \(const uint8 *&aPtr, uint32 aLen, int32 aTimeoutMsec=-1\)](#)
- void [CancelSend \(\)](#)
- [TPVSocketEvent Recv \(uint8 *&aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1\)](#)
- void [CancelRecv \(\)](#)

Static Public Methods

- OsclTCPSocketI * [NewL \(Oscl_DefAlloc &a, OsclSocketServI *aServ, OsclSocketObserver *aObserver, uint32 aId\)](#)

7.229.1 Detailed Description

Internal implementation class for [OsclTCPSocket](#)

7.229.2 Constructor & Destructor Documentation

7.229.2.1 **virtual OsclTCPSocketI::~OsclTCPSocketI () [virtual]**

7.229.3 Member Function Documentation

7.229.3.1 **TPVSocketEvent OsclTCPSocketI::Accept (int32 *aTimeout* = -1) [inline]**

7.229.3.2 **TPVSocketEvent OsclTCPSocketI::BindAsync (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1) [inline]**

7.229.3.3 **void OsclTCPSocketI::CancelAccept () [inline]**

7.229.3.4 **void OsclTCPSocketI::CancelBind () [inline]**

7.229.3.5 **void OsclTCPSocketI::CancelConnect () [inline]**

7.229.3.6 **void OsclTCPSocketI::CancelListen () [inline]**

7.229.3.7 **void OsclTCPSocketI::CancelRecv () [inline]**

7.229.3.8 **void OsclTCPSocketI::CancelSend () [inline]**

7.229.3.9 **void OsclTCPSocketI::CancelShutdown () [inline]**

7.229.3.10 **int32 OsclTCPSocketI::Close () [virtual]**

Implements [OsclIPSocketI](#).

7.229.3.11 **TPVSocketEvent OsclTCPSocketI::Connect (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1) [inline]**

7.229.3.12 **OsclTCPSocketI* OsclTCPSocketI::GetAcceptedSocketL (uint32 *aId*)**

7.229.3.13 **uint8 * OsclTCPSocketI::GetRecvData (int32 * *aLength*) [inline, virtual]**

Implements [OsclIPSocketI](#).

7.229.3.14 **uint8 * OsclTCPSocketI::GetSendData (int32 * *aLength*) [inline, virtual]**

Implements [OsclIPSocketI](#).

- 7.229.3.15 **int32 OsclTCPSocketI::Listen (int *aQueueSize*)** [inline]
- 7.229.3.16 **TPVSocketEvent OsclTCPSocketI::ListenAsync (uint32 *qsize*, int32 *aTimeoutMsec* = -1)** [inline]
- 7.229.3.17 **OsclTCPSocketI* OsclTCPSocketI::NewL (Oscl_DefAlloc & *a*, OsclSocketServI * *aServ*, OsclSocketObserver * *aObserver*, uint32 *aId*)** [static]
- 7.229.3.18 **TPVSocketEvent OsclTCPSocketI::Recv (uint8 *& *aPtr*, uint32 *aMaxLen*, int32 *aTimeoutMsec* = -1)** [inline]
- 7.229.3.19 **TPVSocketEvent OsclTCPSocketI::Send (const uint8 *& *aPtr*, uint32 *aLen*, int32 *aTimeoutMsec* = -1)** [inline]
- 7.229.3.20 **TPVSocketEvent OsclTCPSocketI::Shutdown (TPVSocketShutdown *aHow*, int32 *aTimeoutMsec* = -1)** [inline]

The documentation for this class was generated from the following file:

- [oscl_tcp_socket.h](#)

7.230 OsclThread Class Reference

```
#include <oscl_thread.h>
```

Public Methods

- OSCL_IMPORT_REF OsclThread ()
- OSCL_IMPORT_REF ~OsclThread ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Create (TOsclThreadFuncPtr func, int32 stack_size, TOsclThreadFuncArg argument, OsclThread_State state=Start_on_creation, bool oIsJoinable=false)
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError GetPriority (OsclThreadPriority &refThreadPriority)
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError SetPriority (OsclThreadPriority ePriority)
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Suspend ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Resume ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Terminate (OsclAny *exitcode)

Static Public Methods

- OSCL_IMPORT_REF void Exit (OsclAny *exitcode)
- OSCL_IMPORT_REF void EnableKill ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError GetId (TOsclThreadId &refThreadId)
- OSCL_IMPORT_REF bool CompareId (TOsclThreadId &t1, TOsclThreadId &t2)
- OSCL_IMPORT_REF void SleepMillisec (const int32 msec)

7.230.1 Detailed Description

Thread Class. A subset of Thread APIs. It implements platform independent APIs for thread creation, exiting, suspend, resume, priority and termination. With the use of proper defines it implements the basic thread features. It provides an opaque layer through which user doesn't need to worry about OS specific data.

7.230.2 Constructor & Destructor Documentation

7.230.2.1 OSCL_IMPORT_REF OsclThread::OsclThread ()

Class constructor

7.230.2.2 OSCL_IMPORT_REF OsclThread::~OsclThread ()

Class destructor

7.230.3 Member Function Documentation

7.230.3.1 OSCL_IMPORT_REF bool OsclThread::CompareId (TOsclThreadId & t1, TOsclThreadId & t2) [static]

Static routine to compare whether two thread ID's are equal.

Parameters:

t1, t2: thread ID passed by the application

Returns:

true if equal.

**7.230.3.2 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::Create
(TOsclThreadFuncPtr *func*, int32 *stack_size*, TOsclThreadFuncArg *argument*,
OsclThread_State *state* = Start_on_creation, bool *oIsJoinable* = false)**

This routine will create a thread. The thread may be launched immediately or may be created in a suspended state and launched with a Resume call.

Parameters:

func = Name of the thread Function *stack_size* = Size of the thread stack. If zero, then the platform-specific default stack size will be used. *argument* = Argument to be passed to thread function *state* = Enumeration which specifies the state of the thread on creation with values Running and Suspend. Note: the Suspend option may not be available on all platforms. If it is not supported, the Create call will return INVALID_PARAM_ERROR. *oIsJoinable* = A boolean, which when set to true, creates a Joinable thread. The default value for this is false, which creates a Detached thread. Note 1: When a joinable thread is created, it is imperative to call thread Terminate. Otherwise, it would cause a memory leak. Note 2: This is currently available only for platforms that have support for pthreads.

Returns:

eOsclProcError

7.230.3.3 OSCL_IMPORT_REF void OsclThread::EnableKill () [static]

EnableKill is a static function which can be called by the thread routine in order to enable thread termination without waiting for cancellation points. EnableKill only applies to pthread implementations. For other implementations this function will do nothing.

Returns:

None

7.230.3.4 OSCL_IMPORT_REF void OsclThread::Exit (OsclAny * *exitcode*) [static]

Exit is a static function which is used to end the current thread. When called it just ends the execution of the current thread.

Parameters:

exitcode = Exitcode of the thread. This can be used by other threads to know the exit status of this thread.

Returns:

None

**7.230.3.5 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::GetId
(**ToScIThreadId** & *refThreadId*) [static]**

Static routine to retrieve ID of calling thread.

Parameters:

Thread ID passed by the application

Returns:

Error code

**7.230.3.6 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::GetPriority
(**OsclThreadPriority** & *refThreadPriority*)**

GetThreadPriority gets the priority of the thread. It takes reference of the input argument and assigns priority to it from one of the already defined priorities.

Parameters:

int16& *refThreadPriority* : Output Priority value

Returns:

Error code

7.230.3.7 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::Resume ()

ResumeThread resumes the suspended thread and brings it into execution.

Parameters:

None

Returns:

Error code Note: this function may not be supported on all platforms, and may return NOT_ - IMPLEMENTED.

**7.230.3.8 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::SetPriority
(**OsclThreadPriority** *ePriority*)**

SetThreadPriority sets the priority of the thread. It takes priority as the input argument and assigns it to the thread referred.

Parameters:

ePriorityLevel : Input Priority value

Returns:

Error code Note: this function may not be supported on all platforms, and may return NOT_ - IMPLEMENTED.

7.230.3.9 OSCL_IMPORT_REF void OsclThread::SleepMillisec (const int32 msec) [static]

Suspend current thread execution for specified time.

Parameters:

msec, t2: sleep time in milliseconds.

7.230.3.10 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::Suspend ()

This API suspends the thread being referred. The thread can later be brought into execution by calling OSCL_ResumeThread() on it.

Parameters:

None

Returns:

Error code Note: this function may not be supported on all platforms, and may return NOT_IMPLEMENTED.

**7.230.3.11 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::Terminate
(OsclAny * exitcode)**

Terminate a thread other than the calling thread.

Note: for pthread implementations, the Terminate call will block until the thread has terminated. By default, threads will not terminate until a cancellation point is reached. The EnableKill method may be used to override this default behavior and allow immediate termination.

Parameters:

exitcode = Exitcode of the thread.

Returns:

Error code

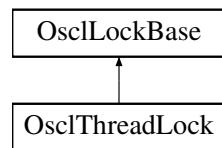
The documentation for this class was generated from the following file:

- [oscl_thread.h](#)

7.231 OsclThreadLock Class Reference

```
#include <oscl_mutex.h>
```

Inheritance diagram for OsclThreadLock::



Public Methods

- OSCL_IMPORT_REF OsclThreadLock ()
- virtual OSCL_IMPORT_REF ~OsclThreadLock ()
- OSCL_IMPORT_REF void Lock ()
- OSCL_IMPORT_REF void Unlock ()

7.231.1 Detailed Description

An implementation of [OsclLockBase](#) using a mutex

7.231.2 Constructor & Destructor Documentation

7.231.2.1 OSCL_IMPORT_REF OsclThreadLock::OsclThreadLock ()

7.231.2.2 virtual OSCL_IMPORT_REF OsclThreadLock::~OsclThreadLock () [virtual]

7.231.3 Member Function Documentation

7.231.3.1 OSCL_IMPORT_REF void OsclThreadLock::Lock () [virtual]

Implements [OsclLockBase](#).

7.231.3.2 OSCL_IMPORT_REF void OsclThreadLock::Unlock () [virtual]

Implements [OsclLockBase](#).

The documentation for this class was generated from the following file:

- [oscl_mutex.h](#)

7.232 OsclTickCount Class Reference

```
#include <oscl_tickcount.h>
```

Static Public Methods

- uint32 [TickCount \(\)](#)
- uint32 [TickCountFrequency \(\)](#)
- uint32 [TickCountPeriod \(\)](#)
- uint32 [TicksToMsec \(uint32 ticks\)](#)
- uint32 [MsecToTicks \(uint32 msec\)](#)

7.232.1 Detailed Description

OsclTickCount class is used to retrieve the system tick count and the tick counter's frequency. The maximum tick count value is equivalent to the maximum uint32 value.

7.232.2 Member Function Documentation

7.232.2.1 uint32 OsclTickCount::MsecToTicks (uint32 *msec*) [static]

This function converts milliseconds to ticks

Returns:

ticks

7.232.2.2 uint32 OsclTickCount::TickCount () [static]

This function returns the current system tick count

Returns:

returns the tick count

7.232.2.3 uint32 OsclTickCount::TickCountFrequency () [static]

This function returns the tick frequency in ticks per second

Returns:

ticks per second

7.232.2.4 uint32 OsclTickCount::TickCountPeriod () [static]

This function returns the tick period in microseconds per tick

Returns:

microseconds per tick

7.232.2.5 uint32 OsclTickCount::TicksToMsec (uint32 *ticks*) [static]

This function converts ticks to milliseconds

Returns:

milliseconds

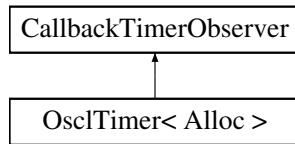
The documentation for this class was generated from the following file:

- [oscl_tickcount.h](#)

7.233 OsclTimer< Alloc > Class Template Reference

```
#include <oscl_timer.h>
```

Inheritance diagram for OsclTimer< Alloc >::



Public Types

- `typedef CallbackTimer< Alloc > callback_timer_type`

Public Methods

- `OsclTimer (const char *name, uint32 frequency=1, int32 priority=OsclActiveObject::EPriorityNominal)`
- `virtual ~OsclTimer ()`
- `void SetObserver (OsclTimerObserver *obs)`
- `void SetFrequency (uint32 frequency)`
- `void SetExactFrequency (uint32 frequency)`
- `void Request (int32 timerID, int32 timeoutInfo, int32 cycles, OsclTimerObserver *obs=0, bool recurring=0)`
- `void Cancel (int32 timerID, int32 timeoutInfo=-1)`
- `void Clear ()`

Protected Methods

- `void TimerBaseElapsed ()`

Friends

- `class CallbackTimer< Alloc >`

template<class Alloc> class OsclTimer< Alloc >

7.233.1 Member Typedef Documentation

7.233.1.1 template<class Alloc> typedef CallbackTimer<Alloc> OsclTimer< Alloc >::callback_timer_type

7.233.2 Constructor & Destructor Documentation

7.233.2.1 template<class Alloc> OsclTimer< Alloc >::OsclTimer (const char * *name*, uint32 *frequency* = 1, int32 *priority* = OsclActiveObject::EPriorityNominal)

Constructor

Parameters:

frequency The frequency of the timer in cycles/second. A value of 1 means the timer will cycle in 1 second intervals.

7.233.2.2 template<class Alloc> OsclTimer< Alloc >::~OsclTimer () [virtual]

7.233.3 Member Function Documentation

7.233.3.1 template<class Alloc> void OsclTimer< Alloc >::Cancel (int32 *timerID*, int32 *timeoutInfo* = -1)

Cancel a timer

Parameters:

timerID used to identify the timer to cancel.

timeoutInfo if not set to -1, this value will be used as additional matching criteria to cancel a timer.

7.233.3.2 template<class Alloc> void OsclTimer< Alloc >::Clear ()

Cancel all pending timers.

7.233.3.3 template<class Alloc> void OsclTimer< Alloc >::Request (int32 *timerID*, int32 *timeoutInfo*, int32 *cycles*, OsclTimerObserver * *obs* = 0, bool *recurring* = 0)

Request a timer

Parameters:

timerID used to identify the timer for cancellation. This value will be returned as part of the timeout event.

timeoutInfo for user info. Returned to the observer on a timeout event

cycles the number of cycles to wait before a timeout event. If the timer frequency is 1 and the cycles are set to 2, then the timeout event will occur in 2 seconds.

obs a local observer object to be called on a timeout event. This observer overides the global observer if set.

7.233.3.4 template<class Alloc> void OsclTimer< Alloc >::SetExactFrequency (uint32 *frequency*)

Set the exact frequency of the timer in microsecond.

Parameters:

frequency A value of 1 means the timer will cycle in one microsecond intervals, 1000 means millisecond intervals, etc.

7.233.3.5 template<class Alloc> void OsclTimer< Alloc >::SetFrequency (uint32 *frequency*)

Set the frequency of the timer in cycles/second.

Parameters:

frequency A value of 1 means the timer will cycle in one second intervals, 1000 means millisecond intervals, etc.

7.233.3.6 template<class Alloc> void OsclTimer< Alloc >::SetObserver ([OsclTimerObserver](#) * *obs*) [inline]

Set the global observer. Each timer can request a local observer, which if set overrides the global observer.

Parameters:

obs observer object.

7.233.3.7 template<class Alloc> void OsclTimer< Alloc >::TimerBaseElapsed () [protected, virtual]

Implements [CallbackTimerObserver](#).

7.233.4 Friends And Related Function Documentation

7.233.4.1 template<class Alloc> friend class [CallbackTimer](#)< Alloc > [friend]

The documentation for this class was generated from the following file:

- [oscl_timer.h](#)

7.234 OsclTimerCompare Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Static Public Methods

- int [compare \(TOsclReady &a, TOsclReady &b\)](#)

7.234.1 Member Function Documentation

7.234.1.1 int OsclTimerCompare::compare (TOsclReady &*a*, TOsclReady &*b*) [static]

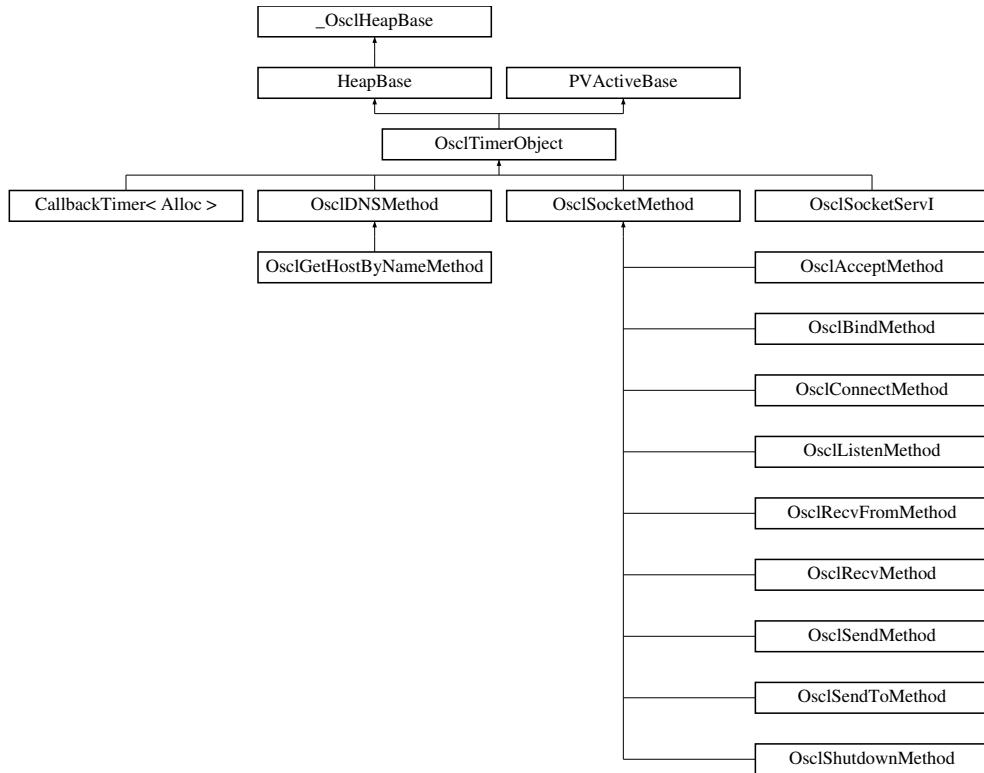
The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

7.235 OsclTimerObject Class Reference

```
#include <oscl_scheduler_ao.h>
```

Inheritance diagram for OsclTimerObject::



Public Methods

- OSCL_IMPORT_REF [OsclTimerObject](#) (int32 aPriority, const char name[])
- virtual OSCL_IMPORT_REF [~OsclTimerObject](#) ()
- OSCL_IMPORT_REF void [AddToScheduler](#) ()
- OSCL_IMPORT_REF void [RemoveFromScheduler](#) ()
- OSCL_IMPORT_REF void [After](#) (int32 aDelayMicrosec)
- OSCL_IMPORT_REF void [RunIfNotReady](#) (uint32 aDelayMicrosec=0)
- OSCL_IMPORT_REF void [SetBusy](#) ()
- OSCL_IMPORT_REF bool [IsBusy](#) () const
- OSCL_IMPORT_REF void [Cancel](#) ()
- OSCL_IMPORT_REF int32 [Priority](#) () const
- OSCL_IMPORT_REF int32 [Status](#) () const
- OSCL_IMPORT_REF void [SetStatus](#) (int32)
- OSCL_IMPORT_REF [OsclAOStatus](#) & [StatusRef](#) ()

Protected Methods

- virtual OSCL_IMPORT_REF void [DoCancel](#) ()
- virtual OSCL_IMPORT_REF int32 [RunError](#) (int32 aError)

7.235.1 Detailed Description

User base class for execution objects. OsclTimerObject defines an exec object with a timer.

7.235.2 Constructor & Destructor Documentation

7.235.2.1 OSCL_IMPORT_REF OsclTimerObject::OsclTimerObject (int32 *aPriority*, const char *name*[])

Constructor.

Parameters:

aPriority (input param): scheduling priority

name (input param): optional name for this AO.

7.235.2.2 virtual OSCL_IMPORT_REF OsclTimerObject::~OsclTimerObject () [virtual]

Destructor.

7.235.3 Member Function Documentation

7.235.3.1 OSCL_IMPORT_REF void OsclTimerObject::AddToScheduler ()

Add this AO to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

7.235.3.2 OSCL_IMPORT_REF void OsclTimerObject::After (int32 *aDelayMicrosec*)

'After' sets the request ready, with request status OSCL_REQUEST_STATUS_PENDING, and starts a timer. When the timer expires, the request will complete with status OSCL_REQUEST_ERR_NONE. Must be called from the same thread in which the active object is scheduled. Will leave if the request is already readied, the object is not added to any scheduler, or the calling thread does not match the scheduling thread.

Parameters:

anInterval: timeout interval in microseconds.

7.235.3.3 OSCL_IMPORT_REF void OsclTimerObject::Cancel ()

Cancel any active request. If the request is pending, this will call the DoCancel routine, wait for the request to cancel, then set the request idle. The AO will not run. If the request is not active, it does nothing. Request must be canceled from the same thread in which it is scheduled.

Reimplemented from [PVActiveBase](#).

7.235.3.4 virtual OSCL_IMPORT_REF void OsclTimerObject::DoCancel () [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will cancel the timer. If any additional action is needed, the derived class may override this. If the derived class does override this, it should explicitly call [OsclTimerObject::DoCancel](#) in its own DoCancel routine.

Implements [PVActiveBase](#).

7.235.3.5 OSCL_IMPORT_REF bool OsclTimerObject::IsBusy ()

Return true if this AO is active, false otherwise.

7.235.3.6 OSCL_IMPORT_REF int32 OsclTimerObject::Priority ()

Return scheduling priority of this exec object.

7.235.3.7 OSCL_IMPORT_REF void OsclTimerObject::RemoveFromScheduler ()

Remove this AO from its scheduler. Will leave if the calling thread context does not match the scheduling thread. Cancels any pending request before removing.

Reimplemented from [PVActiveBase](#).

7.235.3.8 virtual OSCL_IMPORT_REF int32 OsclTimerObject::RunError (int32 *aError*) [protected, virtual]

Run Leave handler. This gets called by scheduler when the Run routine leaves. The default implementation simply returns the leave code. If the derived class wants to handle errors from Run, it may override this. The ExecError should return OsclErrNone if it handles the error, otherwise it should return the input error code.

Parameters:

aError: the leave code generated by the Run.

Implements [PVActiveBase](#).

7.235.3.9 OSCL_IMPORT_REF void OsclTimerObject::RunIfNotReady (uint32 *aDelayMicrosec* = 0)

Complete the request after a time interval. RunIfNotReady is identical to [After\(\)](#) except that it first checks the request status, and if it is already readied, it does nothing.

Parameters:

aDelayMicrosec (input param): delay in microseconds.

7.235.3.10 OSCL_IMPORT_REF void OsclTimerObject::SetBusy ()

Set request ready for this AO. Will leave if the request is already readied, or the exec object is not added to any scheduler, or the calling thread context does not match the scheduler thread.

7.235.3.11 OSCL_IMPORT_REF void OsclTimerObject::SetStatus (int32)

7.235.3.12 OSCL_IMPORT_REF int32 OsclTimerObject::Status ()

Request status access

7.235.3.13 OSCL_IMPORT_REF OsclAOStatus& OsclTimerObject::StatusRef ()

The documentation for this class was generated from the following file:

- [oscl_scheduler_ao.h](#)

7.236 OsclTimerObserver Class Reference

```
#include <oscl_timer.h>
```

Public Methods

- virtual void [TimeoutOccurred](#) (int32 timerID, int32 timeoutInfo)=0
- virtual [~OsclTimerObserver](#) ()

7.236.1 Detailed Description

The observer class to receive timeout callbacks

7.236.2 Constructor & Destructor Documentation

7.236.2.1 virtual OsclTimerObserver::[~OsclTimerObserver](#) () [inline, virtual]

7.236.3 Member Function Documentation

7.236.3.1 virtual void OsclTimerObserver::[TimeoutOccurred](#) (int32 *timerID*, int32 *timeoutInfo*) [pure virtual]

This function will be called when the timer associated with this observer is executed

Parameters:

timerID The ID given at timer request.

timeoutInfo Any extra info given at timer request.

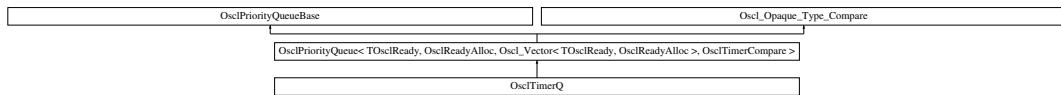
The documentation for this class was generated from the following file:

- [oscl_timer.h](#)

7.237 OsclTimerQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclTimerQ::



Public Methods

- void [Construct](#) (int)
- void [Add](#) ([TOsclReady](#))
- void [Remove](#) ([TOsclReady](#))
- [TOsclReady](#) [PopTop](#) ()
- [TOsclReady](#) [Top](#) ()
- void [Pop](#) ([TOsclReady](#))
- bool [IsIn](#) ([TOsclReady](#))

7.237.1 Member Function Documentation

7.237.1.1 void OsclTimerQ::Add ([TOsclReady](#))

7.237.1.2 void OsclTimerQ::Construct (int)

7.237.1.3 bool OsclTimerQ::IsIn ([TOsclReady](#))

7.237.1.4 void OsclTimerQ::Pop ([TOsclReady](#))

7.237.1.5 [TOsclReady](#) OsclTimerQ::PopTop ()

7.237.1.6 void OsclTimerQ::Remove ([TOsclReady](#))

7.237.1.7 [TOsclReady](#) OsclTimerQ::Top ()

The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

7.238 OsclTLS< T, ID, Registry > Class Template Reference

```
#include <oscl_tls.h>
```

Public Methods

- `OsclTLS ()`
- `~OsclTLS ()`
- `T & operator * () const`
The indirection operator () accesses a value indirectly, through a pointer.*
- `T * operator -> () const`
The indirection operator (->) accesses a value indirectly, through a pointer.
- `bool set ()`
set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

Protected Attributes

- `T * _Ptr`

`template<class T, uint32 ID, class Registry = OsclTLSRegistry> class OsclTLS< T, ID, Registry >`

7.238.1 Constructor & Destructor Documentation

- 7.238.1.1** `template<class T, uint32 ID, class Registry = OsclTLSRegistry> OsclTLS< T, ID, Registry >::OsclTLS () [inline]`
- 7.238.1.2** `template<class T, uint32 ID, class Registry = OsclTLSRegistry> OsclTLS< T, ID, Registry >::~OsclTLS () [inline]`

7.238.2 Member Function Documentation

- 7.238.2.1** `template<class T, uint32 ID, class Registry = OsclTLSRegistry> T& OsclTLS< T, ID, Registry >::operator * () const [inline]`

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclTLS can be used like the regular pointer that it was initialized with.

- 7.238.2.2** `template<class T, uint32 ID, class Registry = OsclTLSRegistry> T* OsclTLS< T, ID, Registry >::operator -> () const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclTLS can be used like the regular pointer that it was initialized with.

7.238.2.3 template<class T, uint32 ID, class Registry = OsclTLSRegistry> bool OsclTLS< T, ID, Registry >::set () [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

7.238.3 Field Documentation

7.238.3.1 template<class T, uint32 ID, class Registry = OsclTLSRegistry> T* OsclTLS< T, ID, Registry >::_Ptr [protected]

The documentation for this class was generated from the following file:

- [oscl_tls.h](#)

7.239 OsclTLSEEx< T, ID, Registry > Class Template Reference

```
#include <oscl_error.h>
```

Public Methods

- `OsclTLSEEx ()`
- `~OsclTLSEEx ()`
- `T & operator * () const`

The indirection operator () accesses a value indirectly, through a pointer.*

- `T * operator -> () const`

The indirection operator (->) accesses a value indirectly, through a pointer.

- `bool set ()`

set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

Protected Attributes

- `T * _Ptr`

```
template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> class OsclTLSEEx< T, ID, Registry >
```

7.239.1 Constructor & Destructor Documentation

7.239.1.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> OsclTLSEEx< T, ID, Registry >::OsclTLSEEx () [inline]

7.239.1.2 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> OsclTLSEEx< T, ID, Registry >::~OsclTLSEEx () [inline]

7.239.2 Member Function Documentation

7.239.2.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T& OsclTLSEEx< T, ID, Registry >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the `OsclTLS` can be used like the regular pointer that it was initialized with.

7.239.2.2 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T* OsclTLSEEx< T, ID, Registry >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the `OsclTLS` can be used like the regular pointer that it was initialized with.

7.239.2.3 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> bool OsclTLSE< T, ID, Registry >::set () [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

7.239.3 Field Documentation

7.239.3.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T* OsclTLSE< T, ID, Registry >::_Ptr [protected]

The documentation for this class was generated from the following file:

- [oscl_error.h](#)

7.240 OsclTLSRegistry Class Reference

```
#include <oscl_tls.h>
```

Static Public Methods

- OSCL_IMPORT_REF [OsclAny](#) * getInstance (uint32 ID, int32 &error)
- OSCL_IMPORT_REF void [registerInstance](#) ([OsclAny](#) *ptr, uint32 ID, int32 &error)

Friends

- class [OsclBase](#)

7.240.1 Member Function Documentation

7.240.1.1 OSCL_IMPORT_REF [OsclAny](#)* OsclTLSRegistry::getInstance (uint32 *ID*, int32 & *error*) [static]

7.240.1.2 OSCL_IMPORT_REF void OsclTLSRegistry::registerInstance ([OsclAny](#) * *ptr*, uint32 *ID*, int32 & *error*) [static]

7.240.2 Friends And Related Function Documentation

7.240.2.1 friend class [OsclBase](#) [friend]

The documentation for this class was generated from the following file:

- [oscl_tls.h](#)

7.241 OsclTLSRegistryEx Class Reference

```
#include <oscl_error.h>
```

Static Public Methods

- [OsclAny * getInstance \(uint32 ID\)](#)
- [void registerInstance \(OsclAny *ptr, uint32 ID\)](#)

7.241.1 Member Function Documentation

7.241.1.1 [OsclAny* OsclTLSRegistryEx::getInstance \(uint32 ID\)](#) [inline, static]

7.241.1.2 [void OsclTLSRegistryEx::registerInstance \(OsclAny *ptr, uint32 ID\)](#) [inline, static]

The documentation for this class was generated from the following file:

- [oscl_error.h](#)

7.242 OsclTrapItem Class Reference

```
#include <oscl_heapbase.h>
```

Public Methods

- OSCL_INLINE [OsclTrapItem \(OsclTrapOperation anOperation\)](#)
- OSCL_INLINE [OsclTrapItem \(OsclTrapOperation anOperation, OsclAny *aPtr\)](#)

Friends

- class [OsclTrapStackItem](#)
- class [OsclTrapStack](#)

7.242.1 Constructor & Destructor Documentation

7.242.1.1 OSCL_INLINE OsclTrapItem::OsclTrapItem ([OsclTrapOperation anOperation](#))

7.242.1.2 OSCL_INLINE OsclTrapItem::OsclTrapItem ([OsclTrapOperation anOperation,](#)
[OsclAny * aPtr](#))

7.242.2 Friends And Related Function Documentation

7.242.2.1 friend class [OsclTrapStack](#) [friend]

7.242.2.2 friend class [OsclTrapStackItem](#) [friend]

The documentation for this class was generated from the following file:

- [oscl_heapbase.h](#)

7.243 OsclTrapStack Class Reference

```
#include <oscl_error_trapcleanup.h>
```

Friends

- class [OsclError](#)
- class [OsclErrorTrap](#)
- class [OsclErrorTrapImp](#)

7.243.1 Detailed Description

A common type for cleanup stack and trap mark stack. for internal use only.

7.243.2 Friends And Related Function Documentation

7.243.2.1 friend class OsclError [friend]

7.243.2.2 friend class OsclErrorTrap [friend]

7.243.2.3 friend class OsclErrorTrapImp [friend]

The documentation for this class was generated from the following file:

- [oscl_error_trapcleanup.h](#)

7.244 OsclTrapStackItem Class Reference

```
#include <oscl_error_trapcleanup.h>
```

Public Methods

- [OsclTrapStackItem \(\)](#)
- [OsclTrapStackItem \(_OsclHeapBase *aCBase\)](#)
- [OsclTrapStackItem \(OsclAny *aTAny\)](#)
- [OsclTrapStackItem \(OsclTrapItem aItem\)](#)

Data Fields

- [_OsclHeapBase * iCBase](#)
- [OsclAny * iTAny](#)
- [OsclTrapOperation iTrapOperation](#)
- [OsclTrapStackItem * iNext](#)

7.244.1 Detailed Description

Internal cleanup stack item type.

7.244.2 Constructor & Destructor Documentation

7.244.2.1 OsclTrapStackItem::OsclTrapStackItem () [inline]

7.244.2.2 OsclTrapStackItem::OsclTrapStackItem (_OsclHeapBase * aCBase) [inline]

7.244.2.3 OsclTrapStackItem::OsclTrapStackItem (OsclAny * aTAny) [inline]

7.244.2.4 OsclTrapStackItem::OsclTrapStackItem (OsclTrapItem aItem) [inline]

7.244.3 Field Documentation

7.244.3.1 _OsclHeapBase* OsclTrapStackItem::iCBase

7.244.3.2 OsclTrapStackItem* OsclTrapStackItem::iNext

7.244.3.3 OsclAny* OsclTrapStackItem::iTAny

7.244.3.4 OsclTrapOperation OsclTrapStackItem::iTrapOperation

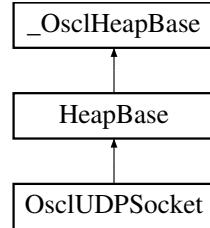
The documentation for this class was generated from the following file:

- [oscl_error_trapcleanup.h](#)

7.245 OsclUDPSocket Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclUDPSocket::



Public Methods

- OSCL_IMPORT_REF ~OsclUDPSocket ()
- OSCL_IMPORT_REF int32 Close ()
- OSCL_IMPORT_REF int32 Bind (OsclNetworkAddress &aAddress)
- OSCL_IMPORT_REF int32 Join (OsclNetworkAddress &aAddress)
- OSCL_IMPORT_REF TPVSocketEvent BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL_IMPORT_REF void CancelBind ()
- OSCL_IMPORT_REF uint8 * GetRecvData (int32 *aLength)
- OSCL_IMPORT_REF uint8 * GetSendData (int32 *aLength)
- OSCL_IMPORT_REF TPVSocketEvent SendTo (const uint8 *aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelSendTo ()
- OSCL_IMPORT_REF TPVSocketEvent RecvFrom (uint8 *aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiRecvLimit=0, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen=NULL, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource=NULL)
- OSCL_IMPORT_REF void CancelRecvFrom ()
- OSCL_IMPORT_REF int32 SetRecvBufferSize (uint32 size)

Static Public Methods

- OSCL_IMPORT_REF OsclUDPSocket * NewL (Oscl_DefAlloc &alloc, OsclSocketServ &aServ, OsclSocketObserver *aObserver, uint32 aId)

7.245.1 Detailed Description

The UDP Socket class

7.245.2 Constructor & Destructor Documentation

7.245.2.1 OSCL_IMPORT_REF OsclUDPSocket::~OsclUDPSocket ()

Destructor. The object must be deleted using the same allocator used in the NewL call.

7.245.3 Member Function Documentation

7.245.3.1 OSCL_IMPORT_REF int32 OsclUDPSocket::Bind ([OsclNetworkAddress](#) & *aAddress*)

Bind a UDP socket to an address. This is a synchronous method.

Parameters:

aAddress: Bind address.

Returns:

Returns OsclErrNone for success, or a platform-specific error code.

7.245.3.2 OSCL_IMPORT_REF [TPVSocketEvent](#) OsclUDPSocket::BindAsync ([OsclNetworkAddress](#) & *aAddress*, int32 *aTimeoutMsec* = (-1))

Bind a UDP socket to an address. This is an asynchronous method.

Parameters:

aAddress: Bind address.

aTimeoutMsec: Optional timeout. Use a negative value for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

7.245.3.3 OSCL_IMPORT_REF void OsclUDPSocket::CancelBind ()

Cancel Bind

This method will cancel any pending BindAsync operation on the current socket, causing the BindAsync to complete with error EPVSocketCancel. If there is no pending BindAsync operation, this method will have no effect.

7.245.3.4 OSCL_IMPORT_REF void OsclUDPSocket::CancelRecvFrom ()

Cancel RecvFrom

This method will cancel any pending RecvFrom operation on the current socket, causing the RecvFrom to complete with error EPVSocketCancel. If there is no pending RecvFrom operation, this method will have no effect.

7.245.3.5 OSCL_IMPORT_REF void OsclUDPSocket::CancelSendTo ()

Cancel SendTo

This method will cancel any pending SendTo operation on the current socket, causing the SendTo to complete with error EPVSocketCancel. If there is no pending SendTo operation, this method will have no effect.

7.245.3.6 OSCL_IMPORT_REF int32 OsclUDPSocket::Close ()

Close a UDP socket. This is a synchronous method.

Once it is closed a socket cannot be re-opened. Sockets are automatically closed when they are deleted. This method may be used to see any error code returned from the platform's socket close call.

Returns:

Returns OsclErrNone for success, or a platform-specific error code.

7.245.3.7 OSCL_IMPORT_REF uint8* OsclUDPSocket::GetRecvData (int32 * aLength)

Retrieve the received data after a successful RecvFrom operation. This is a synchronous method.

Parameters:

aLength: (output) number of bytes of data received.

Returns:

Returns pointer to received data, or NULL if none.

7.245.3.8 OSCL_IMPORT_REF uint8* OsclUDPSocket::GetSendData (int32 * aLength)

Retrieve the sent data after a successful SendTo operation. This is a synchronous method.

Parameters:

aLength: (output) number of bytes of data sent.

Returns:

Returns pointer to sent data, or NULL if none.

7.245.3.9 OSCL_IMPORT_REF int32 OsclUDPSocket::Join ([OsclNetworkAddress](#) & aAddress)

Bind a UDP socket to an address and Join the multicast group. This is a synchronous method.

Parameters:

aAddress: Bind address.

Returns:

Returns OsclErrNone for success, or a platform-specific error code. May throw an OsclErrNotSupported Exception

7.245.3.10 OSCL_IMPORT_REF OsclUDPSocket* OsclUDPSocket::NewL ([Oscl_DefAlloc](#) & alloc, [OsclSocketServ](#) & aServ, [OsclSocketObserver](#) * aObserver, uint32 aId) [static]

Create a UDP Socket. May leave if failure.

Parameters:

alloc: Memory allocator.

aServ: Socket server. Must be connected.

aObserver: Socket observer.

aId: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

Returns:

Returns pointer to socket.

7.245.3.11 OSCL_IMPORT_REF TPVSocketEvent OsclUDPSocket::RecvFrom (uint8 * aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiRecvLimit = 0, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen = NULL, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource = NULL)

Receive Data. This is an asynchronous method.

Parameters:

aPtr: Buffer to receive incoming data

aMaxLen: Length of buffer.

aAddress: (output) Source address.

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

aMultiRecvLimit (optional input): Configures multiple packet receive mode. As long as there are packets queued at the socket and at least aMultiRecvLimit bytes are available in the buffer, recvfrom operations will continue. A value of zero disabled multiple packet mode. The individual packet lengths can be retrieved in the aPacketLen parameter; and the individual packet source addresses can be retrieved in the aPacketSource parameter.

aPacketLen: (optional output) a vector of packet lengths, in case multiple packets were received.

aPacketSource: (optional output) a vector of source addresses, in case multiple packets were received.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

7.245.3.12 OSCL_IMPORT_REF TPVSocketEvent OsclUDPSocket::SendTo (const uint8 * aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1)

Send Data. This is an asynchronous method.

Parameters:

aPtr: Data to send.

aLen: Length of data to send.

aAddress: Destination address.

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

7.245.3.13 OSCL_IMPORT_REF int32 OsclUDPSocket::SetRecvBufferSize (uint32 *size*)

Set the buffer size of the socket This is a synchronous method.

Parameters:

size: buffer size

Returns:

Returns OsclErrNone for success, or a platform-specific error code. May throw an OsclErrNotSupported Exception.

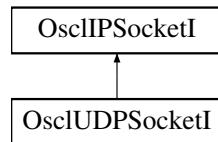
The documentation for this class was generated from the following file:

- [oscl_socket.h](#)

7.246 OsclUDPSocketI Class Reference

```
#include <oscl_udp_socket.h>
```

Inheritance diagram for OsclUDPSocketI::



Public Methods

- virtual ~OsclUDPSocketI ()
- int32 [Close \(\)](#)
- uint8 * [GetRecvData \(int32 *aLength\)](#)
- uint8 * [GetSendData \(int32 *aLength\)](#)
- [TPVSocketEvent BindAsync \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelBind \(\)](#)
- [TPVSocketEvent SendTo \(const uint8 *&aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelSendTo \(\)](#)
- [TPVSocketEvent RecvFrom \(uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiMaxLen=0, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen=NULL, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource=NULL\)](#)
- void [CancelRecvFrom \(\)](#)

Static Public Methods

- OsclUDPSocketI * [NewL \(Oscl_DefAlloc &a, OsclSocketServI *aServ, OsclSocketObserver *aObserver, uint32 aId\)](#)

7.246.1 Detailed Description

Internal implementation class for [OsclUDPSocket](#)

7.246.2 Constructor & Destructor Documentation

7.246.2.1 `virtual OsclUDPSocketI::~OsclUDPSocketI () [virtual]`

7.246.3 Member Function Documentation

7.246.3.1 `TPVSocketEvent OsclUDPSocketI::BindAsync (OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]`

7.246.3.2 `void OsclUDPSocketI::CancelBind () [inline]`

7.246.3.3 `void OsclUDPSocketI::CancelRecvFrom () [inline]`

7.246.3.4 `void OsclUDPSocketI::CancelSendTo () [inline]`

7.246.3.5 `int32 OsclUDPSocketI::Close () [virtual]`

Implements [OsclIPSocketI](#).

7.246.3.6 `uint8 * OsclUDPSocketI::GetRecvData (int32 * aLength) [inline, virtual]`

Implements [OsclIPSocketI](#).

7.246.3.7 `uint8 * OsclUDPSocketI::GetSendData (int32 * aLength) [inline, virtual]`

Implements [OsclIPSocketI](#).

7.246.3.8 `OsclUDPSocketI* OsclUDPSocketI::NewL (Oscl_DefAlloc & a, OsclSocketServI * aServ, OsclSocketObserver * aObserver, uint32 aId) [static]`

7.246.3.9 `TPVSocketEvent OsclUDPSocketI::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiMaxLen = 0, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen = NULL, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource = NULL) [inline]`

7.246.3.10 `TPVSocketEvent OsclUDPSocketI::SendTo (const uint8 *& aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]`

The documentation for this class was generated from the following file:

- [oscl_udp_socket.h](#)

7.247 OsclUuid Struct Reference

```
#include <oscl_uuid.h>
```

Public Methods

- [OsclUuid \(\)](#)
- [OsclUuid \(uint32 l, uint16 w1, uint16 w2, uint8 b1, uint8 b2, uint8 b3, uint8 b4, uint8 b5, uint8 b6, uint8 b7, uint8 b8\)](#)
- [OsclUuid \(const char *aUuidString\)](#)
- [OsclUuid \(const OsclUuid &uuid\)](#)
- [OsclUuid & operator= \(const OsclUuid &src\)](#)
- [bool operator== \(const OsclUuid &src\) const](#)
- [bool operator!= \(const OsclUuid &src\) const](#)

Data Fields

- uint32 [data1](#)
- uint16 [data2](#)
- uint16 [data3](#)
- uint8 [data4](#) [BYTES_IN_UUID_ARRAY]

7.247.1 Detailed Description

OSCL UUID structure used for unique identification of modules and interfaces.

7.247.2 Constructor & Destructor Documentation

7.247.2.1 OsclUuid::OsclUuid () [inline]

7.247.2.2 OsclUuid::OsclUuid (uint32 *l*, uint16 *w1*, uint16 *w2*, uint8 *b1*, uint8 *b2*, uint8 *b3*, uint8 *b4*, uint8 *b5*, uint8 *b6*, uint8 *b7*, uint8 *b8*) [inline]

7.247.2.3 OsclUuid::OsclUuid (const char * *aUuidString*) [inline]

7.247.2.4 OsclUuid::OsclUuid (const OsclUuid & *uuid*) [inline]

7.247.3 Member Function Documentation

7.247.3.1 bool OsclUuid::operator!= (const OsclUuid & *src*) const [inline]

7.247.3.2 OsclUuid& OsclUuid::operator= (const OsclUuid & *src*) [inline]

7.247.3.3 bool OsclUuid::operator== (const OsclUuid & *src*) const [inline]

7.247.4 Field Documentation

7.247.4.1 uint32 OsclUuid::data1

7.247.4.2 uint16 OsclUuid::data2

7.247.4.3 uint16 OsclUuid::data3

7.247.4.4 uint8 OsclUuid::data4[BYTES_IN_UUID_ARRAY]

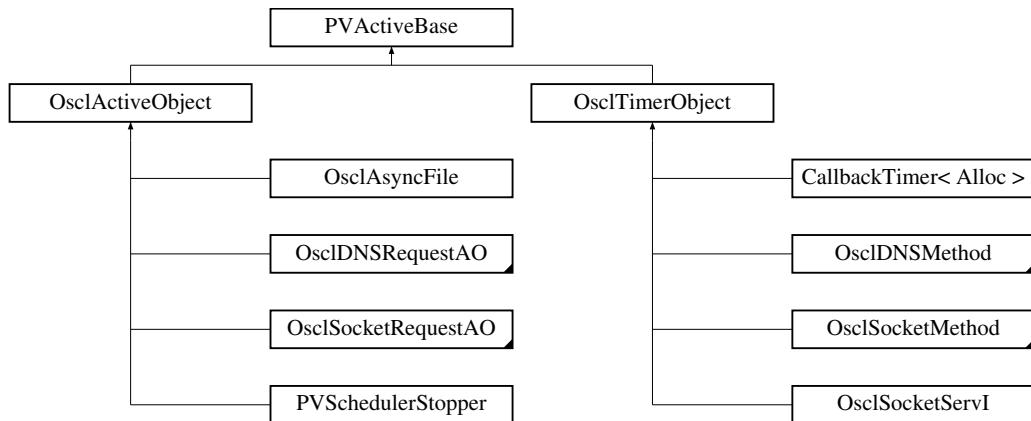
The documentation for this struct was generated from the following file:

- [oscl_uuid.h](#)

7.248 PVActiveBase Class Reference

```
#include <oscl_scheduler_aobase.h>
```

Inheritance diagram for PVActiveBase::



Public Methods

- [PVActiveBase](#) (const char name[], int32 pri)
- virtual [~PVActiveBase](#) ()
- bool [IsInAnyQ](#) ()
- virtual int32 [RunError](#) (int32 aError)=0
- virtual void [Run](#) ()=0
- virtual void [DoCancel](#) ()=0
- void [AddToScheduler](#) ()
- void [RemoveFromScheduler](#) ()
- void [Destroy](#) ()
- void [Activate](#) ()
- OSCL_IMPORT_REF bool [IsAdded](#) () const
- void [Cancel](#) ()

Data Fields

- uint32 [iAddedNum](#)
- [OsclNameString< PVEXECNAMELEN >](#) [iName](#)
- [PVThreadContext](#) [iThreadContext](#)
- [PVActiveStats](#) * [iPVActiveStats](#)
- [TReadyQueLink](#) [iPVReadyQLink](#)
- bool [iBusy](#)
- [OsclAOStatus](#) [iStatus](#)

Friends

- class [PVActiveStats](#)
- class [OsclSchedulerCommonBase](#)

- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [OsclReadyQ](#)
- class [OsclReadyCompare](#)
- class [OsclReadySetPosition](#)
- class [OsclExecScheduler](#)

7.248.1 Detailed Description

PV Scheduler internal AO base class. Both [OsclActiveObject](#) and [OsclTimerObject](#) derive from this class. For Symbian, this just container has the desired additions to the basic CTimer or OsclActiveObj functionality. For non-Symbian, this class contains the entire AO implementation.

7.248.2 Constructor & Destructor Documentation

7.248.2.1 PVActiveBase::PVActiveBase (const char *name*[], int32 *pri*)

7.248.2.2 virtual PVActiveBase::~PVActiveBase () [virtual]

7.248.3 Member Function Documentation

7.248.3.1 void PVActiveBase::Activate ()

7.248.3.2 void PVActiveBase::AddToScheduler ()

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

7.248.3.3 void PVActiveBase::Cancel ()

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

7.248.3.4 void PVActiveBase::Destroy ()

7.248.3.5 virtual void PVActiveBase::DoCancel () [pure virtual]

Implements cancellation of an outstanding request.

This function is called as part of the active object's [Cancel\(\)](#).

It must call the appropriate cancel function offered by the active object's asynchronous service provider. The asynchronous service provider's cancel is expected to act immediately.

[DoCancel\(\)](#) must not wait for event completion; this is handled by [Cancel\(\)](#).

Implemented in [OsclDNSRequestAO](#), [OsclSocketRequestAO](#), [OsclActiveObject](#), and [OsclTimerObject](#).

7.248.3.6 OSCL_IMPORT_REF bool PVActiveBase::IsAdded ()

7.248.3.7 bool PVActiveBase::IsInAnyQ () [inline]

7.248.3.8 void PVActiveBase::RemoveFromScheduler ()

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

7.248.3.9 virtual void PVActiveBase::Run () [pure virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implemented in [OsclDNSMethod](#), [OsclDNSRequestAO](#), [OsclSocketMethod](#), [OsclSocketRequestAO](#), and [CallbackTimer< Alloc >](#).

7.248.3.10 virtual int32 PVActiveBase::RunError (int32 *aError*) [pure virtual]

Virtual routine that gets called if the active object's Run routine leaves.

Parameters:

***aError*:** the leave code generated by the Run.

Returns:

:returns `OsclErrNone` if the error was handled, or returns the input *aError* value if not handled.

Implemented in [OsclActiveObject](#), and [OsclTimerObject](#).

7.248.4 Friends And Related Function Documentation

7.248.4.1 friend class OsclActiveObject [friend]

7.248.4.2 friend class OsclExecScheduler [friend]

7.248.4.3 friend class OsclReadyCompare [friend]

7.248.4.4 friend class OsclReadyQ [friend]

7.248.4.5 friend class OsclReadySetPosition [friend]

7.248.4.6 friend class OsclSchedulerCommonBase [friend]

7.248.4.7 friend class OsclTimerObject [friend]

7.248.4.8 friend class PVActiveStats [friend]

7.248.5 Field Documentation

7.248.5.1 uint32 PVActiveBase::iAddedNum

7.248.5.2 bool PVActiveBase::iBusy

7.248.5.3 OsclNameString<PVEXECNAMELEN> PVActiveBase::iName

7.248.5.4 PVActiveStats* PVActiveBase::iPVActiveStats

7.248.5.5 TReadyQueLink PVActiveBase::iPVReadyQLink

7.248.5.6 OsclAOStatus PVActiveBase::iStatus

The request status associated with an asynchronous request.

This is passed as a parameter to all asynchronous service providers.

The active scheduler uses this to check whether the active object's request has completed.

The function can use the completion code to judge the success or otherwise of the request.

Request status contains one of the values OSCL_REQUEST_ERR_NONE: request completed with no error, or request is not active. OSCL_REQUEST_PENDING: request is active & pending OSCL_REQUEST_ERR_CANCEL: request was canceled before completion. or any user-defined value.

7.248.5.7 PVThreadContext PVActiveBase::iThreadContext

The documentation for this class was generated from the following file:

- [oscl_scheduler_aobase.h](#)

7.249 PVActiveStats Class Reference

```
#include <oscl_scheduler_aobase.h>
```

Friends

- class [PVActiveBase](#)
- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [OsclReadyQ](#)

7.249.1 Detailed Description

PV AO statistics

7.249.2 Friends And Related Function Documentation

7.249.2.1 friend class OsclActiveObject [friend]

7.249.2.2 friend class OsclExecScheduler [friend]

7.249.2.3 friend class OsclExecSchedulerCommonBase [friend]

7.249.2.4 friend class OsclReadyQ [friend]

7.249.2.5 friend class OsclTimerObject [friend]

7.249.2.6 friend class PVActiveBase [friend]

The documentation for this class was generated from the following file:

- [oscl_scheduler_aobase.h](#)

7.250 PVLogger Class Reference

```
#include <pvlogger.h>
```

Public Types

- `typedef int32 log_level_type`
- `typedef int32 message_id_type`
- `typedef int32 filter_status_type`
- `typedef _OsclBasicAllocator alloc_type`

Public Methods

- `void SetLogLevel (log_level_type level)`
- `OSCL_IMPORT_REF void SetLogLevelAndPropagate (log_level_type level)`
- `log_level_type GetLogLevel ()`
- `void DisableAppenderInheritance ()`
- `void AddAppender (OsclSharedPtr< PVLoggerAppender > &appender)`
- `void RemoveAppender (OsclSharedPtr< PVLoggerAppender > &appender)`
- `void AddFilter (OsclSharedPtr< PVLoggerFilter > &filter)`
- `uint32 GetNumAppenders ()`
- `OSCL_IMPORT_REF bool IsActive (log_level_type level)`
- `OSCL_IMPORT_REF void LogMsgStringV (message_id_type msgID, const char *fmt, va_list arguments)`
- `OSCL_IMPORT_REF void LogMsgBuffersV (message_id_type msgID, int32 numPairs, va_list arguments)`
- `OSCL_IMPORT_REF void LogMsgString (message_id_type msgID, const char *fmt,...)`
- `OSCL_IMPORT_REF void LogMsgBuffers (message_id_type msgID, int32 numPairs,...)`
- `OSCL_IMPORT_REF PVLogger (const char *inputTag, log_level_type level, bool oAppenderInheritance)`
- `virtual ~PVLogger ()`

Static Public Methods

- `OSCL_IMPORT_REF void Init ()`
- `OSCL_IMPORT_REF void Cleanup ()`
- `OSCL_IMPORT_REF PVLogger * GetLoggerObject (const char *inputTag)`

Protected Methods

- `void SetParent (PVLogger *parentLogger)`
- `PVLogger * GetParent ()`

Friends

- class `PVLoggerRegistry`

7.250.1 Member Typedef Documentation

7.250.1.1 `typedef _OsclBasicAllocator PVLogger::alloc_type`

7.250.1.2 `typedef int32 PVLogger::filter_status_type`

7.250.1.3 `typedef int32 PVLogger::log_level_type`

7.250.1.4 `typedef int32 PVLogger::message_id_type`

7.250.2 Constructor & Destructor Documentation

7.250.2.1 `OSCL_IMPORT_REF PVLogger::PVLogger (const char * inputTag, log_level_type level, bool oAppenderInheritance)`

Logger Constructor

Parameters:

tag Logger tag, unique to a logging control point

level Active Log level of the logger

oAppenderInheritance

Returns:

NONE

7.250.2.2 `virtual PVLogger::~PVLogger () [inline, virtual]`

7.250.3 Member Function Documentation

7.250.3.1 `void PVLogger::AddAppender (OsclSharedPtr< PVLoggerAppender > & appender) [inline]`

This method adds an appender to the logging control point. Each logger maintains a list of appenders. Any msg to a logger if deemed active is logged to all the appenders.

Parameters:

appender pointer to the appender to add

Returns:

NONE

Exceptions:

leaves if out of memory

7.250.3.2 `void PVLogger::AddFilter (OsclSharedPtr< PVLoggerFilter > & filter) [inline]`

This method adds a message filter to the logging control point. Each logger maintains a list of filters. Any msg to a logger if deemed active is passed through the msg filters prior to logging.

Parameters:

msgFilter pointer to the filter to add

Returns:

NONE

Exceptions:

leaves if out of memory

7.250.3.3 OSCL_IMPORT_REF void PVLogger::Cleanup () [static]

Frees the PVLogger singleton used by the current thread. This must be called before thread exit. No messages can be logged after cleanup.

Returns:**7.250.3.4 void PVLogger::DisableAppenderInheritance () [inline]**

This method disables appender inheritance for the logging control point

7.250.3.5 OSCL_IMPORT_REF PVLogger* PVLogger::GetLoggerObject (const char * *inputTag*) [static]

This is a factory method to create a log control point, with a certain input tag. There is a central registry of all the loggers, with their corresponding tags, called PV Logger Registry. In case the logger with the specified tag exists in the global registry, it is returned, else a new one is created and a pointer to the same is returned.

Parameters:

inputTag logger tag, viz. "x.y.z"

level log level associated with the logging control point (All messages with log levels less than equal to the log level of the control point would be logged)

oAppenderInheritance

Returns:

PVLogger* Pointer to the logging control point

Exceptions:

leaves if out of memory

7.250.3.6 log_level_type PVLogger::GetLogLevel () [inline]

This method returns the log level of a control point. This could either have been set explicitly by the user (at the time of creation or later) or could have been inherited from one of its ancestors.

Returns:

log level associated with the logging control point

7.250.3.7 uint32 PVLogger::GetNumAppenders () [inline]

This method returns the number of appenders attached to the logging control point.

7.250.3.8 PVLogger* PVLogger::GetParent () [inline, protected]**7.250.3.9 OSCL_IMPORT_REF void PVLogger::Init () [static]**

PVLogger needs to be initialized once per thread. This creates the PVLogger singleton that is used throughout the duration of the thread. Initialization must occur before the first message is logged.

Exceptions:

leaves if out of memory

7.250.3.10 OSCL_IMPORT_REF bool PVLogger::IsActive ([log_level_type level](#))

This method determines if a msg passed to the logging control point is active or not. Only messages that are deemed active are logged. Messages are considered not active if any of the following criteria are met:

- All logging is disabled at this logging control point
- If all the log levels, leading upto the root log point are uninitialized
- If the log level of the incoming message is LESS THAN that of the active log level of the logging control point.

Returns:

BOOL

7.250.3.11 OSCL_IMPORT_REF void PVLogger::LogMsgBuffers ([message_id_type msgID](#), int32 *numPairs*, ...)

This method logs opaque data buffers to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

Parameters:

msgID Message ID, that is unique to a message

numPairs Number of (ptr_len, ptr) pairs

arguments Variable list of arguments

Returns:

NONE

7.250.3.12 OSCL_IMPORT_REF void PVLogger::LogMsgBuffersV ([message_id_type msgID](#), int32 *numPairs*, va_list *arguments*)

This method logs opaque data buffers to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

Parameters:

msgID Message ID, that is unique to a message

numPairs Number of (ptr_len, ptr) pairs

arguments Variable list of arguments

Returns:

NONE

7.250.3.13 OSCL_IMPORT_REF void PVLogger::LogMsgString (*message_id_type msgID, const char *fmt, ...)*

This method logs formatted text msg to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

Parameters:

msgID Message ID, that is unique to a message

fmt format string, similar to one taken by printf

arguments Variable list of arguments

Returns:

NONE

7.250.3.14 OSCL_IMPORT_REF void PVLogger::LogMsgStringV (*message_id_type msgID, const char *fmt, va_list arguments)*

This method logs formatted text msg to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

Parameters:

msgID Message ID, that is unique to a message

fmt format string, similar to one taken by printf

arguments Variable list of arguments

Returns:

NONE

7.250.3.15 void PVLogger::RemoveAppender (*OsclSharedPtr< PVLoggerAppender > & appender*) [inline]

This method removes an appender from the logging control point. Each logger maintains a list of appenders. Any msg to a logger if deemed active is logged to all the appenders.

Parameters:

appender pointer to the appender to delete

Returns:

NONE

7.250.3.16 void PVLogger::SetLogLevel (*log_level_type* *level*) [inline]

This method is used to set the log level of a control point.

Parameters:

level log level associated with the logging control point

Returns:

NONE

7.250.3.17 OSCL_IMPORT_REF void PVLogger::SetLogLevelAndPropagate (*log_level_type* *level*)

This method is used to set the log level of a control point, as well as to propagate the level to all the descendants of this control point.

Parameters:

level log level associated with the logging control point

Returns:

NONE

7.250.3.18 void PVLogger::SetParent (PVLogger **parentLogger*) [inline, protected]**7.250.4 Friends And Related Function Documentation****7.250.4.1 friend class PVLoggerRegistry [friend]**

The documentation for this class was generated from the following file:

- [pvlogger.h](#)

7.251 PVLoggerAppender Class Reference

```
#include <pvlogger_accessories.h>
```

Public Types

- `typedef PVLogger::message_id_type message_id_type`

Public Methods

- `virtual ~PVLoggerAppender ()`
- `virtual void AppendString (message_id_type msgID, const char *fmt, va_list va)=0`
- `virtual void AppendBuffers (message_id_type msgID, int32 numPairs, va_list va)=0`

7.251.1 Detailed Description

Base class for all message appenders. This class defines the interface to the message appenders. There are two kinds of msg appender APIs, one to append text messages, and other to append opaque message buffers.

7.251.2 Member Typedef Documentation

7.251.2.1 `typedef PVLogger::message_id_type PVLoggerAppender::message_id_type`

7.251.3 Constructor & Destructor Documentation

7.251.3.1 `virtual PVLoggerAppender::~PVLoggerAppender () [inline, virtual]`

7.251.4 Member Function Documentation

7.251.4.1 `virtual void PVLoggerAppender::AppendBuffers (message_id_type msgID, int32 numPairs, va_list va) [pure virtual]`

7.251.4.2 `virtual void PVLoggerAppender::AppendString (message_id_type msgID, const char *fmt, va_list va) [pure virtual]`

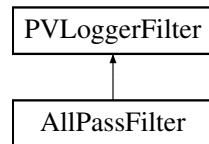
The documentation for this class was generated from the following file:

- `pvlogger_accessories.h`

7.252 PVLoggerFilter Class Reference

```
#include <pvlogger_accessories.h>
```

Inheritance diagram for PVLoggerFilter::



Public Types

- `typedef PVLogger::message_id_type message_id_type`
- `typedef PVLogger::log_level_type log_level_type`
- `typedef PVLogger::filter_status_type filter_status_type`

Public Methods

- `virtual ~PVLoggerFilter()`
- `virtual filter_status_type FilterString (char *tag, message_id_type msgID, log_level_type level)=0`
- `virtual filter_status_type FilterOpaqueMessge (char *tag, message_id_type msgID, log_level_type level)=0`

7.252.1 Detailed Description

Base class for all message filters. This class defines the interface to the message filters. There are two kinds of msg filtering APIs, one to filter text messages, and other to filter opaque message buffers.

7.252.2 Member Typedef Documentation

7.252.2.1 `typedef PVLogger::filter_status_type PVLoggerFilter::filter_status_type`

Reimplemented in [AllPassFilter](#).

7.252.2.2 `typedef PVLogger::log_level_type PVLoggerFilter::log_level_type`

Reimplemented in [AllPassFilter](#).

7.252.2.3 `typedef PVLogger::message_id_type PVLoggerFilter::message_id_type`

Reimplemented in [AllPassFilter](#).

7.252.3 Constructor & Destructor Documentation

7.252.3.1 `virtual PVLoggerFilter::~PVLoggerFilter () [inline, virtual]`

7.252.4 Member Function Documentation

7.252.4.1 `virtual filter_status_type PVLoggerFilter::FilterOpaqueMessge (char * tag, message_id_type msgID, log_level_type level) [pure virtual]`

Implemented in [AllPassFilter](#).

7.252.4.2 `virtual filter_status_type PVLoggerFilter::FilterString (char * tag, message_id_type msgID, log_level_type level) [pure virtual]`

Implemented in [AllPassFilter](#).

The documentation for this class was generated from the following file:

- [pvlogger_accessories.h](#)

7.253 PVLoggerLayout Class Reference

```
#include <pvlogger_accessories.h>
```

Public Types

- `typedef PVLogger::message_id_type message_id_type`

Public Methods

- `virtual ~PVLoggerLayout ()`
- `virtual int32 FormatString (char *formatBuf, int32 formatBufSize, message_id_type msgID, const char *fmt, va_list va)=0`
- `virtual int32 FormatOpaqueMessage (char *formatBuf, int32 formatBufSize, message_id_type msgID, int32 numPairs, va_list va)=0`

7.253.1 Detailed Description

Base class for all message formatters. This class defines the interface to the message formatter. There are two kinds of msg formatting APIs, one to format text messages, and other to format opaque message buffers.

7.253.2 Member Typedef Documentation

7.253.2.1 `typedef PVLogger::message_id_type PVLoggerLayout::message_id_type`

7.253.3 Constructor & Destructor Documentation

7.253.3.1 `virtual PVLoggerLayout::~PVLoggerLayout () [inline, virtual]`

7.253.4 Member Function Documentation

7.253.4.1 `virtual int32 PVLoggerLayout::FormatOpaqueMessage (char *formatBuf, int32 formatBufSize, message_id_type msgID, int32 numPairs, va_list va) [pure virtual]`

Formats the data and copies it to the given buffer.

Returns:

The length of the buffer used.

7.253.4.2 `virtual int32 PVLoggerLayout::FormatString (char *formatBuf, int32 formatBufSize, message_id_type msgID, const char *fmt, va_list va) [pure virtual]`

Formats the string and copies it to the given buffer.

Returns:

The length of the string not including the trailing '\0'

The documentation for this class was generated from the following file:

- [pvlogger_accessories.h](#)

7.254 PVLoggerRegistry Class Reference

```
#include <pvlogger_registry.h>
```

Public Types

- `typedef PVLogger::log_level_type log_level_type`
- `typedef PVLogger::alloc_type alloc_type`

Public Methods

- `OSCL_IMPORT_REF PVLoggerRegistry()`
- `virtual OSCL_IMPORT_REF ~PVLoggerRegistry()`
- `OSCL_IMPORT_REF PVLogger * GetPVLoggerObject (const char *tagIn)`
- `OSCL_IMPORT_REF PVLogger * CreatePVLogger (const char *tagIn, log_level_type level, bool oAppenderInheritance)`
- `OSCL_IMPORT_REF bool SetNodeLogLevelExplicit (char *tagIn, log_level_type level)`
- `OSCL_IMPORT_REF void SetNodeLogLevelExplicit (Oscl_TagTree< PVLogger *, alloc_type >::node_type *node, log_level_type level)`

Static Public Methods

- `OSCL_IMPORT_REF PVLoggerRegistry * GetPVLoggerRegistry()`

7.254.1 Detailed Description

Class: PVLoggerRegistry

PVLoggerRegistry class, maintains a repository of all the loggers, along with their associated tags, in a tag tree. Any request for a log control point is serviced by this class.

Memory Ownership: Creates log control points for each tag, and holds these pointers in the tag tree. `PVLogger` registry is responsible for calling the destructor on each of these loggers.

7.254.2 Member Typedef Documentation

7.254.2.1 `typedef PVLogger::alloc_type PVLoggerRegistry::alloc_type`

7.254.2.2 `typedef PVLogger::log_level_type PVLoggerRegistry::log_level_type`

7.254.3 Constructor & Destructor Documentation

7.254.3.1 `OSCL_IMPORT_REF PVLoggerRegistry::PVLoggerRegistry()`

PVLoggerRegistry Constructor

7.254.3.2 `virtual OSCL_IMPORT_REF PVLoggerRegistry::~PVLoggerRegistry () [virtual]`

PVLoggerRegistry Destructor

7.254.4 Member Function Documentation

7.254.4.1 OSCL_IMPORT_REF PVLogger* PVLoggerRegistry::CreatePVLogger (const char * *tagIn*, log_level_type *level*, bool *oAppenderInheritance*)

This method creates a log control point, with specified tag, and level

Parameters:

inputTag logger tag, viz. "x.y.z"
level log level associated with the logging control point
oAppenderInheritance

Returns:

PVLogger<alloc_type, TheLock>* Pointer to the logging control point

7.254.4.2 OSCL_IMPORT_REF PVLogger* PVLoggerRegistry::GetPVLoggerObject (const char * *tagIn*)

PVLoggerRegistry method to get access to a logging control point, associated with a tag. In case the logger for this tag does not exist, it is created afresh, else pointer to the existing one is returned.

Parameters:

inputTag logger tag, viz. "x.y.z"
level log level associated with the logging control point
oAppenderInheritance

Returns:

PVLogger<Alloc, TheLock>* Pointer to the logging control point

7.254.4.3 OSCL_IMPORT_REF PVLoggerRegistry* PVLoggerRegistry::GetPVLoggerRegistry () [static]

Get the logger registry. There is only one logger registry instance per thread.

7.254.4.4 OSCL_IMPORT_REF void PVLoggerRegistry::SetNodeLogLevelExplicit (Oscl_TagTree< PVLogger *, alloc_type >::node_type * *node*, log_level_type *level*)

This method recursively propagates the log level to all the descendants, of a node.

Parameters:

node Node ptr, associated with a logger, from the tag tree.
level log level associated with the logging control point

Returns:

NONE

7.254.4.5 OSCL_IMPORT_REF bool PVLoggerRegistry::SetNodeLogLevelExplicit (char * *tagIn*, *log_level_type level*)

This method propagates the log level to all the descendants of the node, with a specified tag.

Parameters:

tagIn logger tag, viz. "x.y.z"

level log level associated with the logging control point

Returns:

true on success, else false.

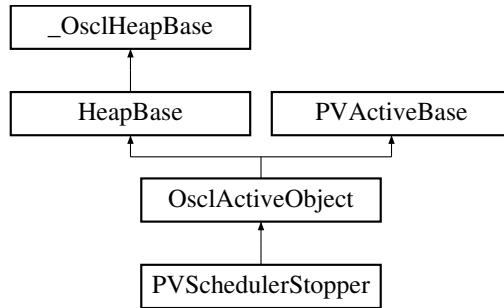
The documentation for this class was generated from the following file:

- [pvlogger_registry.h](#)

7.255 PVSchedulerStopper Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for PVSchedulerStopper::



Public Methods

- [PVSchedulerStopper \(\)](#)
- [~PVSchedulerStopper \(\)](#)

7.255.1 Detailed Description

Scheduler stopper AO class, for internal use by scheduler.

7.255.2 Constructor & Destructor Documentation

7.255.2.1 PVSchedulerStopper::PVSchedulerStopper ()

7.255.2.2 PVSchedulerStopper::~PVSchedulerStopper ()

The documentation for this class was generated from the following file:

- [oscl_scheduler.h](#)

7.256 PVSockBufRecv Class Reference

```
#include <oscl_socket_request.h>
```

Public Methods

- [PVSockBufRecv \(\)](#)
- [PVSockBufRecv \(uint8 *aPtr, uint32 aLen, uint32 aMax\)](#)
- [PVSockBufRecv \(const PVSockBufRecv &a\)](#)

Data Fields

- [uint8 * iPtr](#)
- [uint32 iLen](#)
- [uint32 iMaxLen](#)

7.256.1 Constructor & Destructor Documentation

7.256.1.1 PVSockBufRecv::PVSockBufRecv () [inline]

7.256.1.2 PVSockBufRecv::PVSockBufRecv (uint8 * *aPtr*, uint32 *aLen*, uint32 *aMax*) [inline]

7.256.1.3 PVSockBufRecv::PVSockBufRecv (const PVSockBufRecv & *a*) [inline]

7.256.2 Field Documentation

7.256.2.1 uint32 PVSockBufRecv::iLen

7.256.2.2 uint32 PVSockBufRecv::iMaxLen

7.256.2.3 uint8* PVSockBufRecv::iPtr

The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.257 PVSockBufSend Class Reference

```
#include <oscl_socket_request.h>
```

Public Methods

- [PVSockBufSend \(\)](#)
- [PVSockBufSend \(const uint8 *aPtr, uint32 aLen\)](#)
- [PVSockBufSend \(const PVSockBufSend &a\)](#)

Data Fields

- const uint8 * [iPtr](#)
- uint32 [iLen](#)

7.257.1 Constructor & Destructor Documentation

7.257.1.1 PVSockBufSend::PVSockBufSend () [inline]

7.257.1.2 PVSockBufSend::PVSockBufSend (const uint8 * *aPtr*, uint32 *aLen*) [inline]

7.257.1.3 PVSockBufSend::PVSockBufSend (const PVSockBufSend & *a*) [inline]

7.257.2 Field Documentation

7.257.2.1 uint32 PVSockBufSend::iLen

7.257.2.2 const uint8* PVSockBufSend::iPtr

The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.258 PVThreadContext Class Reference

```
#include <oscl_scheduler_threadcontext.h>
```

Public Methods

- OSCL_IMPORT_REF PVThreadContext ()
- OSCL_IMPORT_REF ~PVThreadContext ()
- OSCL_IMPORT_REF bool IsSameThreadContext ()
- OSCL_IMPORT_REF void EnterThreadContext ()
- OSCL_IMPORT_REF void ExitThreadContext ()

Static Public Methods

- OSCL_IMPORT_REF uint32 Id ()
- OSCL_IMPORT_REF bool ThreadHasScheduler ()

Friends

- class PVActiveBase
- class OsclActiveObject
- class OsclTimerObject
- class OsclExecScheduler
- class OsclCoeActiveScheduler
- class OsclExecSchedulerCommonBase
- class OsclExecSchedulerBase
- class OsclCoeActiveSchedulerBase

7.258.1 Constructor & Destructor Documentation

7.258.1.1 OSCL_IMPORT_REF PVThreadContext::PVThreadContext ()

7.258.1.2 OSCL_IMPORT_REF PVThreadContext::~PVThreadContext ()

7.258.2 Member Function Documentation

7.258.2.1 OSCL_IMPORT_REF void PVThreadContext::EnterThreadContext ()

enter and exit thread context.

7.258.2.2 OSCL_IMPORT_REF void PVThreadContext::ExitThreadContext ()

7.258.2.3 OSCL_IMPORT_REF uint32 PVThreadContext::Id () [static]

static routine to get a unique thread ID for caller's thread context.

7.258.2.4 OSCL_IMPORT_REF bool PVThreadContext::IsSameThreadContext ()

compare caller's thread context to this one.

7.258.2.5 OSCL_IMPORT_REF bool PVThreadContext::ThreadHasScheduler () [static]

a static utility to tell whether the calling thread has any scheduler— either Oscl scheduler or native scheduler.

7.258.3 Friends And Related Function Documentation**7.258.3.1 friend class OsclActiveObject [friend]****7.258.3.2 friend class OsclCoeActiveScheduler [friend]****7.258.3.3 friend class OsclCoeActiveSchedulerBase [friend]****7.258.3.4 friend class OsclExecScheduler [friend]****7.258.3.5 friend class OsclExecSchedulerBase [friend]****7.258.3.6 friend class OsclExecSchedulerCommonBase [friend]****7.258.3.7 friend class OsclTimerObject [friend]****7.258.3.8 friend class PVActiveBase [friend]**

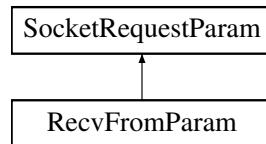
The documentation for this class was generated from the following file:

- [oscl_scheduler_threadcontext.h](#)

7.259 RecvFromParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for RecvFromParam::



Public Methods

- [RecvFromParam \(uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, uint32 flags, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource\)](#)

Data Fields

- [PVSockBufRecv iBufRecv](#)
- [uint32 iFlags](#)
- [OsclNetworkAddress & iAddr](#)
- [uint32 iMultiMaxLen](#)
- [Oscl_Vector< uint32, OsclMemAllocator > * iPacketLen](#)
- [Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * iPacketSource](#)

7.259.1 Constructor & Destructor Documentation

[7.259.1.1 RecvFromParam::RecvFromParam \(uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, uint32 flags, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource\) \[inline\]](#)

7.259.2 Field Documentation

[7.259.2.1 OsclNetworkAddress& RecvFromParam::iAddr](#)

[7.259.2.2 PVSockBufRecv RecvFromParam::iBufRecv](#)

[7.259.2.3 uint32 RecvFromParam::iFlags](#)

[7.259.2.4 uint32 RecvFromParam::iMultiMaxLen](#)

[7.259.2.5 Oscl_Vector<uint32, OsclMemAllocator>* RecvFromParam::iPacketLen](#)

[7.259.2.6 Oscl_Vector<OsclNetworkAddress, OsclMemAllocator>* RecvFromParam::iPacketSource](#)

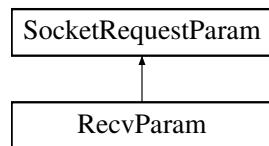
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.260 RecvParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for RecvParam::



Public Methods

- [RecvParam \(uint8 *&aPtr, uint32 aMaxLen, uint32 flags\)](#)

Data Fields

- [PVSockBufRecv iBufRecv](#)
- [uint32 iFlags](#)

7.260.1 Constructor & Destructor Documentation

7.260.1.1 RecvParam::RecvParam (uint8 *& aPtr, uint32 aMaxLen, uint32 flags) [inline]

7.260.2 Field Documentation

7.260.2.1 PVSockBufRecv RecvParam::iBufRecv

7.260.2.2 uint32 RecvParam::iFlags

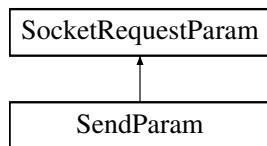
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.261 SendParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SendParam::



Public Methods

- [SendParam \(const uint8 *&aPtr, uint32 aLen, uint32 aFlags\)](#)

Data Fields

- [PVSockBufSend iBufSend](#)
- [uint32 iFlags](#)
- [uint32 iXferLen](#)

7.261.1 Detailed Description

Socket method parameter sets

7.261.2 Constructor & Destructor Documentation

7.261.2.1 SendParam::SendParam (const uint8 *& aPtr, uint32 aLen, uint32 aFlags) [inline]

7.261.3 Field Documentation

7.261.3.1 PVSockBufSend SendParam::iBufSend

7.261.3.2 uint32 SendParam::iFlags

7.261.3.3 uint32 SendParam::iXferLen

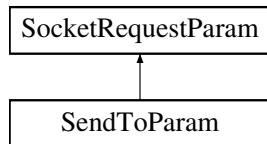
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.262 SendToParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SendToParam::



Public Methods

- [SendToParam \(const uint8 *&aPtr, uint32 aLen, OsclNetworkAddress &anAddr, uint32 flags\)](#)
- [~SendToParam \(\)](#)

Data Fields

- [PVSockBufSend iBufSend](#)
- [uint32 iFlags](#)
- [OsclNetworkAddress iAddr](#)
- [uint32 iXferLen](#)

7.262.1 Constructor & Destructor Documentation

7.262.1.1 SendToParam::SendToParam (const uint8 *& *aPtr*, uint32 *aLen*, OsclNetworkAddress & *anAddr*, uint32 *flags*) [inline]

7.262.1.2 SendToParam::~SendToParam () [inline]

7.262.2 Field Documentation

7.262.2.1 OsclNetworkAddress SendToParam::iAddr

7.262.2.2 PVSockBufSend SendToParam::iBufSend

7.262.2.3 uint32 SendToParam::iFlags

7.262.2.4 uint32 SendToParam::iXferLen

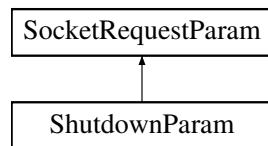
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.263 ShutdownParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for ShutdownParam::



Public Methods

- [ShutdownParam \(TPVSocketShutdown aHow\)](#)

Data Fields

- [TPVSocketShutdown iHow](#)

7.263.1 Constructor & Destructor Documentation

7.263.1.1 ShutdownParam::ShutdownParam ([TPVSocketShutdown aHow](#)) [inline]

7.263.2 Field Documentation

7.263.2.1 [TPVSocketShutdown ShutdownParam::iHow](#)

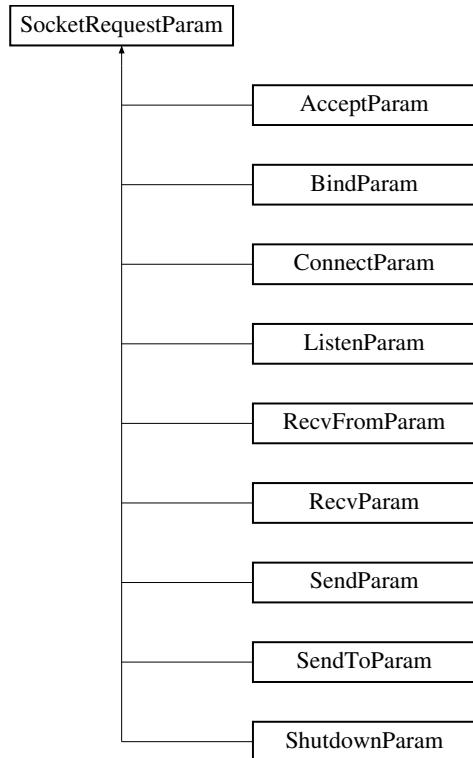
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

7.264 SocketRequestParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SocketRequestParam::



Public Methods

- [SocketRequestParam \(TPVSocketFxn aFxn\)](#)

Data Fields

- [TPVSocketFxn iFxn](#)

7.264.1 Detailed Description

Base class for all socket method parameter sets

7.264.2 Constructor & Destructor Documentation

7.264.2.1 `SocketRequestParam::SocketRequestParam (TPVSocketFxn aFxn)` [inline]

7.264.3 Field Documentation

7.264.3.1 `TPVSocketFxn SocketRequestParam::iFxn`

The documentation for this class was generated from the following file:

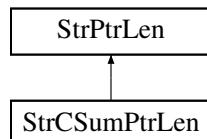
- `oscl_socket_request.h`

7.265 StrCSumPtrLen Struct Reference

same as [StrPtrLen](#), but includes checksum field and method to speed up querying

```
#include <oscl_str_ptr_len.h>
```

Inheritance diagram for StrCSumPtrLen::



Public Types

- [typedef int16 CheckSumType](#)

Public Methods

- [void setPtrLen \(const char *newPtr, uint32 newLen\)](#)
- [CheckSumType getCheckSum \(\) const](#)
- [OSCL_IMPORT_REF void setCheckSum \(\)](#)
- [StrCSumPtrLen \(\)](#)
- [StrCSumPtrLen \(const char *newPtr\)](#)
- [StrCSumPtrLen \(const char *newPtr, uint32 newLen\)](#)
- [StrCSumPtrLen \(const StrCSumPtrLen &rhs\)](#)
- [StrCSumPtrLen \(const StrPtrLen &rhs\)](#)
- [c_bool isCIEquivalentTo \(const StrCSumPtrLen &rhs\) const](#)
- [c_bool operator== \(const StrCSumPtrLen &rhs\) const](#)
- [c_bool operator!= \(const StrCSumPtrLen &rhs\) const](#)
- [StrCSumPtrLen & operator= \(const StrCSumPtrLen &rhs\)](#)
- [StrCSumPtrLen & operator= \(const StrPtrLen &rhs\)](#)
- [StrCSumPtrLen & operator= \(const char *rhs\)](#)

Protected Attributes

- [CheckSumType checkSum](#)

7.265.1 Detailed Description

same as [StrPtrLen](#), but includes checksum field and method to speed up querying

7.265.2 Member Typedef Documentation

7.265.2.1 `typedef int16 StrCSumPtrLen::CheckSumType`

7.265.3 Constructor & Destructor Documentation

7.265.3.1 `StrCSumPtrLen::StrCSumPtrLen () [inline]`

7.265.3.2 `StrCSumPtrLen::StrCSumPtrLen (const char * newPtr) [inline]`

7.265.3.3 `StrCSumPtrLen::StrCSumPtrLen (const char * newPtr, uint32 newLen) [inline]`

7.265.3.4 `StrCSumPtrLen::StrCSumPtrLen (const StrCSumPtrLen & rhs) [inline]`

7.265.3.5 `StrCSumPtrLen::StrCSumPtrLen (const StrPtrLen & rhs) [inline]`

7.265.4 Member Function Documentation

7.265.4.1 `CheckSumType StrCSumPtrLen::getCheckSum () const [inline]`

7.265.4.2 `c_bool StrCSumPtrLen::isCIEquivalentTo (const StrCSumPtrLen & rhs) const [inline]`

7.265.4.3 `c_bool StrCSumPtrLen::operator!= (const StrCSumPtrLen & rhs) const [inline]`

7.265.4.4 `StrCSumPtrLen& StrCSumPtrLen::operator= (const char * rhs) [inline]`

Reimplemented from [StrPtrLen](#).

7.265.4.5 `StrCSumPtrLen& StrCSumPtrLen::operator= (const StrPtrLen & rhs) [inline]`

Reimplemented from [StrPtrLen](#).

7.265.4.6 `StrCSumPtrLen& StrCSumPtrLen::operator= (const StrCSumPtrLen & rhs) [inline]`

7.265.4.7 `c_bool StrCSumPtrLen::operator== (const StrCSumPtrLen & rhs) const [inline]`

7.265.4.8 `OSCL_IMPORT_REF void StrCSumPtrLen::setCheckSum ()`

7.265.4.9 `void StrCSumPtrLen::setPtrLen (const char * newPtr, uint32 newLen) [inline]`

Reimplemented from [StrPtrLen](#).

7.265.5 Field Documentation

7.265.5.1 `CheckSumType StrCSumPtrLen::checkSum [protected]`

The documentation for this struct was generated from the following file:

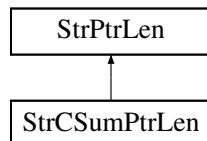
- [oscl_str_ptr_len.h](#)

7.266 StrPtrLen Struct Reference

This data structure encapsulates a set of functions used to perform.

```
#include <oscl_str_ptr_len.h>
```

Inheritance diagram for StrPtrLen::



Public Methods

- [StrPtrLen](#) (const char *newPtr)
- [StrPtrLen](#) (const char *newPtr, uint32 newLen)
- [StrPtrLen](#) ()
- [StrPtrLen](#) (const StrPtrLen &rhs)
- const char * [c_str](#) () const
- int32 [length](#) () const
- int32 [size](#) () const
- void [setPtrLen](#) (const char *newPtr, uint32 newLen)
- [c_bool](#) [isCIEquivalentTo](#) (const StrPtrLen &rhs) const
- [c_bool](#) [isCIPrefixOf](#) (const StrPtrLen &rhs) const
- int32 [operator==](#) (const StrPtrLen &rhs) const
- int32 [operator!=](#) (const StrPtrLen &rhs) const
- StrPtrLen & [operator=](#) (const StrPtrLen &rhs)
- StrPtrLen & [operator=](#) (const char *rhs)

Protected Methods

- bool [isLetter](#) (const char c) const

Protected Attributes

- const char * [ptr](#)
- int32 [len](#)

7.266.1 Detailed Description

This data structure encapsulates a set of functions used to perform.

standard string operations. It should be used for null-terminated constant (non-modifiable) strings of char type.

7.266.2 Constructor & Destructor Documentation

7.266.2.1 `StrPtrLen::StrPtrLen (const char * newPtr) [inline]`

7.266.2.2 `StrPtrLen::StrPtrLen (const char * newPtr, uint32 newLen) [inline]`

7.266.2.3 `StrPtrLen::StrPtrLen () [inline]`

7.266.2.4 `StrPtrLen::StrPtrLen (const StrPtrLen & rhs) [inline]`

7.266.3 Member Function Documentation

7.266.3.1 `const char* StrPtrLen::c_str () const [inline]`

7.266.3.2 `c_bool StrPtrLen::isCIEquivalentTo (const StrPtrLen & rhs) const [inline]`

7.266.3.3 `c_bool StrPtrLen::isCIPrefixOf (const StrPtrLen & rhs) const [inline]`

7.266.3.4 `bool StrPtrLen::isLetter (const char c) const [inline, protected]`

7.266.3.5 `int32 StrPtrLen::length () const [inline]`

7.266.3.6 `int32 StrPtrLen::operator!= (const StrPtrLen & rhs) const [inline]`

7.266.3.7 `StrPtrLen& StrPtrLen::operator= (const char * rhs) [inline]`

Reimplemented in [StrCSumPtrLen](#).

7.266.3.8 `StrPtrLen& StrPtrLen::operator= (const StrPtrLen & rhs) [inline]`

Reimplemented in [StrCSumPtrLen](#).

7.266.3.9 `int32 StrPtrLen::operator== (const StrPtrLen & rhs) const [inline]`

7.266.3.10 `void StrPtrLen::setPtrLen (const char * newPtr, uint32 newLen) [inline]`

Reimplemented in [StrCSumPtrLen](#).

7.266.3.11 `int32 StrPtrLen::size () const [inline]`

7.266.4 Field Documentation

7.266.4.1 `int32 StrPtrLen::len [protected]`

7.266.4.2 `const char* StrPtrLen::ptr [protected]`

The documentation for this struct was generated from the following file:

- [oscl_str_ptr_len.h](#)

7.267 TimeValue Class Reference

The TimeValue class represents a time value in a format native to the system.

```
#include <oscl_time.h>
```

Public Methods

- OSCL_COND_IMPORT_REF [TimeValue](#) ()

Create a TimeValue representing the current time.
- OSCL_COND_IMPORT_REF [TimeValue](#) (const [TimeValue](#) &Tv)

Copy constructor.
- OSCL_COND_IMPORT_REF [TimeValue](#) (long tv, [TimeUnits](#) units)

Create a TimeValue representing an interval of tv units.
- OSCL_COND_IMPORT_REF [TimeValue](#) (const [OsclBasicTimeStruct](#) &in_tv)

Create a TimeValue representing the absolute time specified by the BasicTimeStruct.
- OSCL_COND_IMPORT_REF [TimeValue](#) ([OsclBasicDateTimeStruct](#) in_ts)

Create a TimeValue representing the absolute time specified by the BasicDateTimeStruct.
- OSCL_COND_IMPORT_REF int32 [get_local_time](#) ()

Get the local time after having adjusted for daylight saving.
- OSCL_COND_IMPORT_REF void [set_to_zero](#) ()

Set the time value to zero (represents a zero interval).
- OSCL_COND_IMPORT_REF void [set_to_current_time](#) ()

Set the time value to the current system time.
- OSCL_COND_IMPORT_REF void [set_from_ntp_time](#) (const uint32 ntp_offset)

This method covers a 32-bit NTP offset to system time.
- OSCL_COND_IMPORT_REF uint32 [get_sec](#) () const

Get a 32 bit value representing the seconds since the (system dependent) epoch.
- OSCL_COND_IMPORT_REF int32 [to_msec](#) () const

Get a 32 bit value representing the number of microseconds in the time value.
- OSCL_IMPORT_REF char * [get_str_ctime](#) (CtimeStrBuf ctime_strbuf)

Get a string containing a text representation of this TimeValue object.
- OSCL_IMPORT_REF int [get_pv8601_str_time](#) (PV8601timeStrBuf time_strbuf)

Get a PV extended text representation of the Time based on the ISO 8601 format.
- OSCL_IMPORT_REF char * [get_rfc822_gmtime_str](#) (int max_time_strlen, char *time_str)

Get a string containing a text representation of this TimeValue object.

Get a text representation of the time in the GMT timezone based on the RFC 822 / RFC 1123 (also described in the HTTP spec RFC 2068 and RFC 2616.

- OSCL_COND_IMPORT_REF bool `is_zero ()`
Determine if the time value is zero.
- OSCL_COND_IMPORT_REF TimeValue & `operator= (const TimeValue &a)`
Assignment operator.
- OSCL_COND_IMPORT_REF TimeValue & `operator+= (const TimeValue &a)`
 $+ = \text{operator}$
- OSCL_COND_IMPORT_REF TimeValue & `operator-= (const TimeValue &a)`
 $- = \text{operator}$
- OSCL_COND_IMPORT_REF TimeValue & `operator *= (const int scale)`
This operator scales the time value by a constant.
- OSCL_COND_IMPORT_REF OsclBasicTimeStruct * `get_timeval_ptr ()`

Friends

- class `NTPTime`
- OSCL_COND_IMPORT_REF friend bool `operator== (const TimeValue &a, const TimeValue &b)`
- OSCL_COND_IMPORT_REF friend bool `operator!= (const TimeValue &a, const TimeValue &b)`
- OSCL_COND_IMPORT_REF friend bool `operator<= (const TimeValue &a, const TimeValue &b)`
- OSCL_COND_IMPORT_REF friend bool `operator>= (const TimeValue &a, const TimeValue &b)`
- OSCL_COND_IMPORT_REF friend bool `operator< (const TimeValue &a, const TimeValue &b)`
- OSCL_COND_IMPORT_REF friend bool `operator> (const TimeValue &a, const TimeValue &b)`

7.267.1 Detailed Description

The TimeValue class represents a time value in a format native to the system.

This class provides common time functions independent of any OS. The actual representation used is native to the system that the class is compiled on to increase efficiency. Macros used in this class:

- OSCL_HAS_ANSI_STRING_SUPPORT:

Definitions to determine the type of basic time structure used to store the time

- OSCL_HAS_UNIX_TIME_FUNCS
- OSCL_HAS_SYMBIAN_SUPPORT
- OSCL_HAS_MSWIN_SUPPORT

7.267.2 Constructor & Destructor Documentation

7.267.2.1 OSCL_COND_IMPORT_REF TimeValue::TimeValue ()

Create a TimeValue representing the current time.

7.267.2.2 OSCL_COND_IMPORT_REF TimeValue::TimeValue (const TimeValue & *Tv*)

Copy constructor.

7.267.2.3 OSCL_COND_IMPORT_REF TimeValue::TimeValue (long *tv*, TimeUnits *units*)

Create a TimeValue representing an interval of tv units.

Parameters:

tv The number of units in the interval to be represented by this TimeValue.

units The units of the tv argument. Must be in the enumeration TimeUnits.

7.267.2.4 OSCL_COND_IMPORT_REF TimeValue::TimeValue (const OsclBasicTimeStruct & *in_tv*)

Create a TimeValue representing the absolute time specified by the BasicTimeStruct.

Parameters:

in_tv OsclBasicTimeStruct as defined for each platform.

7.267.2.5 OSCL_COND_IMPORT_REF TimeValue::TimeValue (OsclBasicDateTimeStruct *in_ts*)

Create a TimeValue representing the absolute time specified by the BasicDateTimeStruct.

Parameters:

in_ts OsclBasicDateTimeStruct as defined for each platform provides the date in a readable format (i.e. day, date , week etc.) Notes: Implementation incomplete (= not done) on Win32, Wince, Symbian

7.267.3 Member Function Documentation

7.267.3.1 OSCL_COND_IMPORT_REF int32 TimeValue::get_local_time ()

Get the local time after having adjusted for daylight saving.

Notes: Implementation incomplete (= not done) on Win32, Wince, Symbian

7.267.3.2 OSCL_IMPORT_REF int TimeValue::get_pv8601_str_time (PV8601timeStrBuf *time_strbuf*)

Get a PV extended text representation of the Time based on the ISO 8601 format.

Parameters:

time_strbuf A PV8601timeStrBuf object to which the text representation will be written,

Returns:

The number of characters copied to the buffer, not including the terminating null. The returned string is of the form "19850412T101530.047Z".

7.267.3.3 OSCL_IMPORT_REF char* TimeValue::get_rfc822_gmtime_str (int max_time_strlen, char * time_str)

Get a text representation of the time in the GMT timezone based on the RFC 822 / RFC 1123 (also described in the HTTP spec RFC 2068 and RFC 2616).

Parameters:

max_time_strlen The maximum number of characters that can be written to the buffer.

time_str A pointer to the buffer to which the characters will be written.

Returns:

Returns a pointer to the buffer (same as *time_str*) containing a null terminated (c-style) string of the form "Wed, 30 Jun 1993 21:49:08 GMT".

7.267.3.4 OSCL_COND_IMPORT_REF uint32 TimeValue::get_sec ()

Get a 32 bit value representing the seconds since the (system dependent) epoch.

Returns:

This call returns a 32 bit value representing the number of seconds since the epoch. On unix systems this represents the number of seconds since the unix epoch Jan 1 1970. On Win32 this represents the number of seconds since Jan 1, 1601. This is intended to be used for intervals rather than for absolute time. (On Win32 for example, a 32 bit value would be too small to represent the number of seconds from the epoch until the current time.)

7.267.3.5 OSCL_IMPORT_REF char* TimeValue::get_str_ctime (CtimeStrBuf ctime_strbuf)

Get a string containing a text representation of this TimeValue object.

Parameters:

ctime_strbuf A CtimeStrBuf object to which the text representation will be written,

Returns:

A pointer to the input CtimeStrBuf is returned. This string is null terminated of the form "Wed Jun 30 21:49:08 1993".

7.267.3.6 OSCL_COND_IMPORT_REF OsclBasicTimeStruct* TimeValue::get_timeval_ptr ()**7.267.3.7 OSCL_COND_IMPORT_REF uint32 TimeValue::get_usec ()**

Get a 32 bit value representing the number of microseconds in the time value.

Returns:

Returns a uint32 value representing the number of microseconds in the time value after subtracting off the whole seconds.

7.267.3.8 OSCL_COND_IMPORT_REF bool TimeValue::is_zero ()

Determine if the time value is zero.

7.267.3.9 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator *= (const int scale)

This operator scales the time value by a constant.

7.267.3.10 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator+= (const TimeValue & a)

+= operator

7.267.3.11 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator-= (const TimeValue & a)

-= operator

7.267.3.12 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator= (const TimeValue & a)

Assignment operator.

7.267.3.13 OSCL_COND_IMPORT_REF void TimeValue::set_from_ntp_time (const uint32 ntp_offset)

This method converts a 32-bit NTP offset to system time.

This method takes a 32-bit ntp offset which is the number of seconds from 0 h Jan 1, 1900 and converts it to the system time

7.267.3.14 OSCL_COND_IMPORT_REF void TimeValue::set_to_current_time ()

Set the time value to the current system time.

7.267.3.15 OSCL_COND_IMPORT_REF void TimeValue::set_to_zero ()

Set the time value to zero (represents a zero interval).

7.267.3.16 OSCL_COND_IMPORT_REF int32 TimeValue::to_msec ()

7.267.4 Friends And Related Function Documentation

7.267.4.1 friend class NTPTime [friend]

7.267.4.2 OSCL_COND_IMPORT_REF friend bool operator!= (const TimeValue & a, const TimeValue & b) [friend]

7.267.4.3 OSCL_COND_IMPORT_REF friend bool operator< (const TimeValue & a, const TimeValue & b) [friend]

7.267.4.4 OSCL_COND_IMPORT_REF friend bool operator<= (const TimeValue & a, const TimeValue & b) [friend]

7.267.4.5 OSCL_COND_IMPORT_REF friend bool operator== (const TimeValue & a, const TimeValue & b) [friend]

7.267.4.6 OSCL_COND_IMPORT_REF friend bool operator> (const TimeValue & a, const TimeValue & b) [friend]

7.267.4.7 OSCL_COND_IMPORT_REF friend bool operator>= (const TimeValue & a, const TimeValue & b) [friend]

The documentation for this class was generated from the following file:

- [oscl_time.h](#)

7.268 TLSStorageOps Class Reference

```
#include <oscl_tls.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [save_registry \(TOsclTlsKey *key, OsclAny *ptr, int32 &\)](#)
- OSCL_IMPORT_REF [OsclAny * get_registry \(TOsclTlsKey *key\)](#)

7.268.1 Member Function Documentation

**7.268.1.1 OSCL_IMPORT_REF OsclAny* TLSStorageOps::get_registry (TOsclTlsKey * *key*)
[static]**

**7.268.1.2 OSCL_IMPORT_REF void TLSStorageOps::save_registry (TOsclTlsKey * *key*,
OsclAny * *ptr*, int32 &) [static]**

The documentation for this class was generated from the following file:

- [oscl_tls.h](#)

7.269 TReadyQueLink Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Public Methods

- [TReadyQueLink \(\)](#)

Data Fields

- int32 [iAOPriority](#)
- uint32 [iTimeToRunTicks](#)
- uint32 [iTimeQueuedTicks](#)
- uint32 [iSeqNum](#)
- OsclAny * [iIsIn](#)

7.269.1 Detailed Description

This class defines the queue link, which is common to both ready Q and timer Q. Each AO contains its own queue link object.

7.269.2 Constructor & Destructor Documentation

7.269.2.1 [TReadyQueLink::TReadyQueLink \(\) \[inline\]](#)

7.269.3 Field Documentation

7.269.3.1 [int32 TReadyQueLink::iAOPriority](#)

7.269.3.2 [OsclAny* TReadyQueLink::iIsIn](#)

7.269.3.3 [uint32 TReadyQueLink::iSeqNum](#)

7.269.3.4 [uint32 TReadyQueLink::iTimeQueuedTicks](#)

7.269.3.5 [uint32 TReadyQueLink::iTimeToRunTicks](#)

The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

7.270 WStrPtrLen Struct Reference

This data structure encapsulates a set of functions used to perform.

```
#include <oscl_str_ptr_len.h>
```

Public Methods

- [WStrPtrLen \(const oscl_wchar *newPtr\)](#)
- [WStrPtrLen \(const oscl_wchar *newPtr, uint32 newLen\)](#)
- [WStrPtrLen \(\)](#)
- [WStrPtrLen \(const WStrPtrLen &rhs\)](#)
- [const oscl_wchar * c_str \(\) const](#)
- [int32 length \(\) const](#)
- [int32 size \(\) const](#)
- [void setPtrLen \(const oscl_wchar *newPtr, uint32 newLen\)](#)
- [c_bool isCIEquivalentTo \(const WStrPtrLen &rhs\) const](#)
- [int32 operator== \(const WStrPtrLen &rhs\) const](#)
- [int32 operator!= \(const WStrPtrLen &rhs\) const](#)
- [WStrPtrLen & operator= \(const WStrPtrLen &rhs\)](#)
- [WStrPtrLen & operator= \(const oscl_wchar *rhs\)](#)

Protected Attributes

- [const oscl_wchar * ptr](#)
- [int32 len](#)

7.270.1 Detailed Description

This data structure encapsulates a set of functions used to perform.

standard string operations. It should be used for null-terminated constant strings (non-modifiable) of wchar type.

7.270.2 Constructor & Destructor Documentation

- 7.270.2.1 `WStrPtrLen::WStrPtrLen (const oscl_wchar * newPtr) [inline]`
- 7.270.2.2 `WStrPtrLen::WStrPtrLen (const oscl_wchar * newPtr, uint32 newLen) [inline]`
- 7.270.2.3 `WStrPtrLen::WStrPtrLen () [inline]`
- 7.270.2.4 `WStrPtrLen::WStrPtrLen (const WStrPtrLen & rhs) [inline]`

7.270.3 Member Function Documentation

- 7.270.3.1 `const oscl_wchar* WStrPtrLen::c_str () const [inline]`
- 7.270.3.2 `c_bool WStrPtrLen::isCIEquivalentTo (const WStrPtrLen & rhs) const [inline]`
- 7.270.3.3 `int32 WStrPtrLen::length () const [inline]`
- 7.270.3.4 `int32 WStrPtrLen::operator!= (const WStrPtrLen & rhs) const [inline]`
- 7.270.3.5 `WStrPtrLen& WStrPtrLen::operator= (const oscl_wchar * rhs) [inline]`
- 7.270.3.6 `WStrPtrLen& WStrPtrLen::operator= (const WStrPtrLen & rhs) [inline]`
- 7.270.3.7 `int32 WStrPtrLen::operator== (const WStrPtrLen & rhs) const [inline]`
- 7.270.3.8 `void WStrPtrLen::setPtrLen (const oscl_wchar * newPtr, uint32 newLen) [inline]`
- 7.270.3.9 `int32 WStrPtrLen::size () const [inline]`

7.270.4 Field Documentation

- 7.270.4.1 `int32 WStrPtrLen::len [protected]`
- 7.270.4.2 `const oscl_wchar* WStrPtrLen::ptr [protected]`

The documentation for this struct was generated from the following file:

- `oscl_str_ptr_len.h`

Chapter 8

oscl File Documentation

8.1 oscl_aostatus.h File Reference

Some basic types used with active objects.

```
#include "osclconfig_proc.h"  
#include "oscl_base.h"  
#include "oscl_aostatus.inl"
```

Data Structures

- class [OsclAOStatus](#)

Variables

- const int32 [OSCL_REQUEST_ERR_NONE](#) = 0
- const int32 [OSCL_REQUEST_PENDING](#) = (-0x7fffffff)
- const int32 [OSCL_REQUEST_ERR_CANCEL](#) = (-1)
- const int32 [OSCL_REQUEST_ERR_GENERAL](#) = (-2)

8.1.1 Detailed Description

Some basic types used with active objects.

8.2 oscl_assert.h File Reference

The file [oscl_assert.h](#) provides an OSCL_ASSERT macro to document assumptions and test them during development.

```
#include "oscl_base.h"  
#include "oscl_assert.inl"
```

Defines

- #define [OSCL_ASSERT](#)(_expr) ((_expr)?((void)0):OSCL Assert(# _expr,__FILE__,__LINE__))

Functions

- OSCL_COND_IMPORT_REF void [_OSCL_Abort](#) ()
This function terminates the current process abnormally.
- OSCL_IMPORT_REF void [OSCL Assert](#) (const char *expr, const char *filename, int line_number)
OSCL_ASSERT macro evaluates an expression and when the result is false, prints a diagnostic message and aborts the program.

8.2.1 Detailed Description

The file [oscl_assert.h](#) provides an OSCL_ASSERT macro to document assumptions and test them during development.

8.3 oscl_base.h File Reference

The file [oscl_base.h](#) is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros.

```
#include "osclconfig.h"
#include "oscl_base_macros.h"
#include "oscl_types.h"
#include "osclconfig_check.h"
#include "pv_config.h"
```

Defines

- `#define OSCL_HAS_SINGLETON_SUPPORT 1`

Functions

- `void PVOsclBase_Init ()`
- `void PVOsclBase_Cleanup ()`

8.3.1 Detailed Description

The file [oscl_base.h](#) is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros.

8.4 oscl_base_alloc.h File Reference

A basic allocator that does not rely on other modules.

```
#include "osclconfig.h"  
#include "oscl_defalloc.h"  
#include "osclconfig_memory.h"
```

Data Structures

- class [_OsclBasicAllocator](#)

8.4.1 Detailed Description

A basic allocator that does not rely on other modules.

8.5 oscl_base_macros.h File Reference

This file defines common macros and constants for basic compilation support.

```
#include "osclconfig.h"
```

Defines

- `#define NULL_TERM_CHAR '\0'`
The NULL_TERM_CHAR is used to terminate c-style strings.
- `#define NULL (0)`
if the NULL macro isn't already defined, then define it as zero.
- `#define OSCL_INLINE inline`
- `#define OSCL_COND_EXPORT_REF`
- `#define OSCL_COND_IMPORT_REF`
- `#define OSCL_CONST_CAST(type, exp) ((type)(exp))`
Type casting macros.
- `#define OSCL_STATIC_CAST(type, exp) ((type)(exp))`
- `#define OSCL_REINTERPRET_CAST(type, exp) ((type)(exp))`
- `#define OSCL_DYNAMIC_CAST(type, exp) ((type)(exp))`
- `#define OSCL_UNUSED_ARG(vbl) (void)(vbl)`
- `#define OSCL_UNUSED_RETURN(value) return value`
- `#define OSCL_MIN(a, b) ((a) < (b) ? (a) : (b))`
- `#define OSCL_MAX(a, b) ((a) > (b) ? (a) : (b))`
- `#define OSCL_ABS(a) ((a) > (0) ? (a) : -(a))`
- `#define OSCL_TEMPLATED_DESTRUCTOR_CALL(type, simple_type) type :: ~simple_type ()`
- `#define OSCL_UNSIGNED_CONST(x) x`
- `#define OSCL_PACKED_VAR "error"`

8.5.1 Detailed Description

This file defines common macros and constants for basic compilation support.

8.6 oscl_bin_stream.h File Reference

Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.

```
#include "oscl_base.h"
#include "oscl_bin_stream.inl"
```

Data Structures

- class [OsclBinIStream](#)
- class [OsclBinIStreamBigEndian](#)
- class [OsclBinIStreamLittleEndian](#)
- class [OsclBinOStream](#)

Class OsclBinOStream implements the basic stream functions for an output stream.

- class [OsclBinOStreamBigEndian](#)

Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering.

- class [OsclBinOStreamLittleEndian](#)

Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.

- class [OsclBinStream](#)

8.6.1 Detailed Description

Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.

8.7 oscl_byte_order.h File Reference

This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).

```
#include "oscl_base.h"  
#include "oscl_byte_order.inl"
```

Functions

- void [little_endian_to_host](#) (char *data, uint32 size)
Convert little endian to host format.
- void [host_to_little_endian](#) (char *data, unsigned int size)
Convert host to little endian format.
- void [big_endian_to_host](#) (char *data, unsigned int size)
Convert big endian to host format.
- void [host_to_big_endian](#) (char *data, unsigned int size)
Convert host to big endian format.

8.7.1 Detailed Description

This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).

8.8 oscl_defalloc.h File Reference

The file defines simple default memory allocator classes. These allocators are used by the [Oscl_Vector](#) and [Oscl_Map](#) class, etc.

```
#include "oscl_base.h"  
#include "osclconfig_compiler_warnings.h"  
#include "oscl_mem_inst.h"
```

Data Structures

- class [Oscl_Alloc](#)
- class [Oscl_Dealloc](#)
- class [Oscl_DefAlloc](#)
- class [Oscl_TAlloc](#)
- class [OsclAllocDestructDealloc](#)
- class [OsclDestructDealloc](#)
- struct [rebind](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [ALLOCATE\(n\)](#) [allocate_fl\(n,__FILE__,__LINE__\)](#)
- #define [ALLOC_AND_CONSTRUCT\(n\)](#) [alloc_and_construct_fl\(n,__FILE__,__LINE__\)](#)

8.8.1 Detailed Description

The file defines simple default memory allocator classes. These allocators are used by the [Oscl_Vector](#) and [Oscl_Map](#) class, etc.

8.9 oscl_dll.h File Reference

Defines a DLL entry point.

```
#include "osclconfig.h"
```

Defines

- #define **OSCL_DLL_ENTRY_POINT()** void oscl_dll_entry_point() {}
- #define **OSCL_DLL_ENTRY_POINT_DEFAULT()**

8.9.1 Detailed Description

Defines a DLL entry point.

8.10 oscl_dns.h File Reference

The file [oscl_socket.h](#) defines the OSCL DNS APIs.

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_defalloc.h"
#include "oscl_socket.h"
```

Data Structures

- class [OsclIDNS](#)
- class [OsclDNSObserver](#)

Enumerations

- enum [TPVDNSFxn](#) { [EPVDNSGetHostByName](#) }
- enum [TPVDNSEvent](#) { [EPVDNSSuccess](#), [EPVDNSPending](#), [EPVDNSTimeout](#), [EPVDNSFailure](#), [EPVDNSCancel](#) }

8.10.1 Detailed Description

The file [oscl_socket.h](#) defines the OSCL DNS APIs.

8.11 oscl_dns_gethostbyname.h File Reference

```
#include "oscl_dns_method.h"
#include "oscl_dns.h"
#include "osclconfig_io.h"
```

Data Structures

- class [OsclGetHostByNameMethod](#)
- class [OsclGetHostByNameRequest](#)

8.12 oscl_dns_imp.h File Reference

```
#include "oscl_dns_tuneables.h"  
#include "oscl_dns_imp_pv.h"
```

8.13 oscl_dns_imp_base.h File Reference

```
#include "oscl_socket_imp.h"
#include "oscl_dns_request.h"
#include "oscl_dns.h"
```

Data Structures

- class [OsclDNSIBase](#)

8.14 oscl_dns_imp_pv.h File Reference

```
#include "oscl_socket_imp_base.h"
#include "oscl_dns_request.h"
#include "oscl_dns_imp_base.h"
```

Data Structures

- class [OsclDNSI](#)

8.15 oscl_dns_method.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_dns.h"
#include "pvlogger.h"
```

Data Structures

- class [OsclIDNSMethod](#)
- class [OsclDNSRequestAO](#)

8.16 oscl_dns_param.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_dns_tuneables.h"
#include "oscl_namestring.h"
#include "oscl_dns.h"
#include "oscl_mutex.h"
#include "oscl_semaphore.h"
```

Data Structures

- class [DNSRequestParam](#)
- class [GetHostByNameParam](#)

Typedefs

- typedef [OsclMemAllocator TDNSRequestParamAllocator](#)

8.16.1 Typedef Documentation

8.16.1.1 typedef [OsclMemAllocator TDNSRequestParamAllocator](#)

8.17 oscl_dns_request.h File Reference

```
#include "oscl_dns_tuneables.h"
#include "oscl_namestring.h"
#include "oscl_dns.h"
#include "oscl_socket_types.h"
```

Data Structures

- class [OsclDNSRequest](#)

8.18 oscl_dns_tuneables.h File Reference

```
#include "osclconfig_io.h"
#include "osclconfig_proc.h"
```

Defines

- #define PV_DNS_SERVER 1
- #define PV_DNS_IS_THREAD OSCL_HAS_THREAD_SUPPORT

8.18.1 Define Documentation

8.18.1.1 #define PV_DNS_IS_THREAD OSCL_HAS_THREAD_SUPPORT

PV_DNS_IS_THREAD chooses either the threaded or AO-based implementation of the PV DNS request.
Note: AO-based option is not good here, since some DNS requests will block the caller until completion.

8.18.1.2 #define PV_DNS_SERVER 1

Enable/disable the PV DNS server here.

8.19 oscl_double_list.h File Reference

Internal use types for scheduler.

```
#include "osclconfig_proc.h"
#include "oscl_base.h"
#include "oscl_assert.h"
#include "oscl_double_list.inl"
```

Data Structures

- class [OsclDoubleLink](#)
- class [OsclDoubleList](#)
- class [OsclDoubleListBase](#)
- class [OsclDoubleRunner](#)
- class [OsclPriorityLink](#)
- class [OsclPriorityList](#)

Defines

- #define [QUE_ITER_BEGIN](#)(_type, _qname)
- #define [QUE_ITER_END](#)(_qname)

Functions

- template<class T, class S> T * [OsclPtrAdd](#) (T *aPtr, S aVal)
- template<class T, class S> T * [OsclPtrSub](#) (T *aPtr, S aVal)

8.19.1 Detailed Description

Internal use types for scheduler.

8.20 oscl_errno.h File Reference

Defines functions to access additional information on errors where supported through an errno or similar service.

```
#include "oscl_base.h"
#include "osclconfig_error.h"
#include "oscl_errno.inl"
```

Functions

- OSCL_IMPORT_REF bool [OSCL_IsErrnoSupported \(\)](#)
This function determines if a particular system saves the error number that occurs on a system call.
- OSCL_IMPORT_REF int [OSCL_GetLastError \(\)](#)
This function returns the value of the system's global error number variable.
- OSCL_IMPORT_REF bool [OSCL_SetLastError \(int newVal\)](#)
This function sets the last error code for the system.
- OSCL_IMPORT_REF char * [OSCL_StrError \(int errnum\)](#)
This function maps an error number to an error-message string.

8.20.1 Detailed Description

Defines functions to access additional information on errors where supported through an errno or similar service.

8.21 oscl_error.h File Reference

OSCL Error trap and cleanup include file.

```
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_error_codes.h"
#include "oscl_singleton.h"
#include "oscl_assert.h"
#include "oscl_tls.h"
```

Data Structures

- class [OsclError](#)
- class [OsclErrorTrap](#)
- class [OsclTLSEx](#)
- class [OsclTLSRegistryEx](#)

Defines

- #define [OSCL_TRAPSTACK_PUSH\(a\)](#) OsclError::PushL(a)
- #define [OSCL_TRAPSTACK_POP\(\)](#) OsclError::Pop()
- #define [OSCL_TRAPSTACK_POPDEALLOC\(\)](#) OsclError::PopDealloc()

8.21.1 Detailed Description

OSCL Error trap and cleanup include file.

8.22 oscl_error_allocator.h File Reference

Defines a memory allocation class used by the oscl error layer.

```
#include "oscl_base.h"  
#include "oscl_base_macros.h"  
#include "osclconfig_error.h"  
#include "oscl_assert.h"  
#include "oscl_defalloc.h"
```

Data Structures

- class [OsclErrorAllocator](#)

This class provides static methods to invoke the user defined memory allocation routines.

8.22.1 Detailed Description

Defines a memory allocation class used by the oscl error layer.

8.23 oscl_error_codes.h File Reference

Defines basic error and leave codes.

Defines

- #define `OsclErrNone` 0
- #define `OsclErrGeneral` 100
- #define `OsclErrNoMemory` 101
- #define `OsclErrCancelled` 102
- #define `OsclErrNotSupported` 103
- #define `OsclErrArgument` 104
- #define `OsclErrBadHandle` 105
- #define `OsclErrAlreadyExists` 106
- #define `OsclErrBusy` 107
- #define `OsclErrNotReady` 108
- #define `OsclErrCorrupt` 109
- #define `OsclErrTimeout` 110
- #define `OsclErrOverflow` 111
- #define `OsclErrUnderflow` 112
- #define `OsclErrInvalidState` 113
- #define `OsclErrNoResources` 114
- #define `OsclErrNotInstalled` 115
- #define `OsclErrAlreadyInstalled` 116
- #define `OsclErrSystemCallFailed` 117
- #define `OsclErrNoHandler` 118
- #define `OsclErrThreadContextIncorrect` 119
- #define `OSCL_ERR_NONE` `OsclErrNone`
- #define `OSCL_BAD_ALLOC_EXCEPTION_CODE` `OsclErrNoMemory`
- #define `OsclSuccess` 0
- #define `OsclPending` 1
- #define `OsclFailure` -1

Typedefs

- typedef int32 `OsclLeaveCode`
- typedef int32 `OsclReturnCode`

8.23.1 Detailed Description

Defines basic error and leave codes.

8.24 oscl_error_imp.h File Reference

Internal error implementation support.

```
#include "osclconfig_error.h"  
#include "oscl_error_imp_jumps.h"
```

Defines

- #define PVERROR_IMP_JUMPS

8.24.1 Detailed Description

Internal error implementation support.

8.25 oscl_error_imp_cppexceptions.h File Reference

Implementation File for Leave using C++ exceptions.

```
#include "oscl_error_trapcleanup.h"
```

Data Structures

- class [internalLeave](#)

Defines

- #define [PVError_DoLeave\(\)](#) [internalLeave](#) __ilv;__ilv.a=0;throw(__ilv)
- #define [_PV_TRAP](#)(__r, __s)
- #define [_PV_TRAP_NO_TLS](#)(__trapimp, __r, __s)

8.25.1 Detailed Description

Implementation File for Leave using C++ exceptions.

8.26 oscl_error_imp_fatalerror.h File Reference

Implementation File for Leave using system fatal error.

```
#include "oscl_assert.h"
```

Defines

- #define PVError_DoLeave() _OSCL_Abort()
- #define _PV_TRAP(__r, __s)
- #define _PV_TRAP_NO_TLS(__tr, __r, __s)

8.26.1 Detailed Description

Implementation File for Leave using system fatal error.

8.26.2 Define Documentation

8.26.2.1 #define _PV_TRAP(__r, __s)

Value:

```
__r=OsclErrNone; \
{__s;}
```

8.26.2.2 #define _PV_TRAP_NO_TLS(__tr, __r, __s)

Value:

```
__r=OsclErrNone; \
{__s;}
```

8.26.2.3 #define PVError_DoLeave() _OSCL_Abort()

8.27 oscl_error_imp_jumps.h File Reference

Implementation of using Setjmp / Longjmp.

```
#include "oscl_error_trapcleanup.h"
#include "oscl_assert.h"
#include "osclconfig_error.h"
#include "oscl_defalloc.h"
#include "oscl_error.h"
```

Data Structures

- class [OsclJump](#)

Defines

- #define OSCL_JUMP_MAX_JUMP_MARKS OSCL_MAX_TRAP_LEVELS
- #define internalLeave (-1)
- #define PVError_DoLeave() OsclJump::StaticJump(internalLeave)
- #define _PV_TRAP(__r, __s)
- #define _PV_TRAP_NO_TLS(__trapimp, __r, __s)

8.27.1 Detailed Description

Implementation of using Setjmp / Longjmp.

8.27.2 Define Documentation

8.27.2.1 #define _PV_TRAP(__r, __s)

Value:

```
__r=OsclErrNone; \
{ \
    OsclErrorTrapImp* __trap=OsclErrorTrapImp::Trap(); \
    if(!__trap){__s;}else{ \
        int __tr=setjmp(*(__trap->iJumpData->Top())); \
        if (__tr==0)\ 
            {__s;} \
        else if (__tr==internalLeave)\ 
            {__r=__trap->iLeave;} \
            __trap->UnTrap();} \
}
```

8.27.2.2 #define _PV_TRAP_NO_TLS(__trapimp, __r, __s)

Value:

```
__r=OsclErrNone; \
{ \
    OsclErrorTrapImp* __trap=OsclErrorTrapImp::TrapNoTls(__trapimp); \
    if(!__trap){__s;}else{ \
        int __tr=setjmp(*(__trap->iJumpData->Top())); \
        if (__tr==0)\ 
            {__s;} \
        else if (__tr==internalLeave)\ 
            {__r=__trap->iLeave;} \
        __trap->UnTrap();} \
}
```

8.27.2.3 #define PVError_DoLeave() OsclJump::StaticJump(internalLeave)

8.28 oscl_error_trapcleanup.h File Reference

OSCL Error trap and cleanup implementation include file.

```
#include "osclconfig_error.h"  
  
#include "oscl_heapbase.h"  
  
#include "oscl_defalloc.h"  
  
#include "oscl_assert.h"  
  
#include "oscl_error.h"  
  
#include "oscl_base_alloc.h"  
  
#include "oscl_tls.h"  
  
#include "oscl_singleton.h"  
  
#include "oscl_error_imp.h"
```

Data Structures

- class [OsclErrorTrapImp](#)
- class [OsclTrapStack](#)
- class [OsclTrapStackItem](#)

Defines

- #define [OSCL_MAX_TRAP_LEVELS](#) 20
- #define [PVERRORTRAP_REGISTRY_ID](#) [OSCL_TLS_ID_PVERRORTRAP](#)
- #define [PVERRORTRAP_REGISTRY](#) [OsclTLSRegistry](#)

8.28.1 Detailed Description

OSCL Error trap and cleanup implementation include file.

8.29 oscl_exception.h File Reference

contains all the exception handling macros and classes

```
#include "oscl_error.h"
#include "oscl_error_imp.h"
```

Data Structures

- class [OsclException](#)

oscl_exception.h contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

Defines

- #define [OSCL_LEAVE\(_leave_status\)](#) OsclError::Leave(_leave_status)
Use this macro to cause a Leave. It terminates the execution of the current active function.
- #define [OSCL_TRY\(_leave_status, _statements\)](#) _PV_TRAP(_leave_status,_statements)
This macro will be used to set up a try block.
- #define [OSCL_TRY_NO_TLS\(_trapimp, _leave_status, _statements\)](#) _PV_TRAP_NO_TLS(_-
 $_trapimp, _leave_status, _statements)$
• #define [OSCL_FIRST_CATCH_ANY\(_leave_status, _statements\)](#) if (_leave_status!=OsclErrNone){ _statements; }
This section defines the macros to be used in the catch block following the try block Use this macro to call a function that handles all exception types thrown in the preceding try block.
- #define [OSCL_FIRST_CATCH\(_leave_status, _catch_value, _statements\)](#) if (_leave_status!=OsclErrNone && _leave_status == _catch_value){_statements;}
Use this macro to define a block of code that catches the first exception type thrown in the preceding try block.
- #define [OSCL_CATCH\(_leave_status, _catch_value, _statements\)](#) else if (_leave_status!=OsclErrNone && _leave_status == _catch_value){_statements;}
Use this macro to define a block of code for catching additional exception types.
- #define [OSCL_CATCH_ANY\(_leave_status, _statements\)](#) else if (_leave_status!=OsclErrNone){ _-
 $_statements;$
Use this macro to call a function that will catch all remaining exception types.
- #define [OSCL_LAST_CATCH\(_leave_status\)](#) else if (_leave_status!=OsclErrNone){OSCL_-
 $LEAVE(_leave_status);$
Use this macro if OSCL_CATCH_ANY has not been used. It will mark the end of the catch block.

8.29.1 Detailed Description

contains all the exception handling macros and classes

8.30 oscl_exclusive_ptr.h File Reference

This file defines the [OsclExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

```
#include "oscl_defalloc.h"
```

Data Structures

- class [OsclExclusiveArrayPtr](#)

The OsclExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory.

- class [OsclExclusivePtr](#)

The OsclExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory.

- class [OsclExclusivePtrA](#)

The OsclExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory.

8.30.1 Detailed Description

This file defines the [OsclExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

8.31 oscl_file_async_read.h File Reference

```
#include "oscl_base.h"
#include "osclconfig_io.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_scheduler_ao.h"
#include "oscl_file_io.h"
#include "oscl_semaphore.h"
```

Data Structures

- class [OsclAsyncFile](#)
- class [OsclAsyncFileBuffer](#)
- class [OsclBuf](#)
- class [OsclPtr](#)
- class [OsclPtrC](#)

8.32 oscl_file_cache.h File Reference

The file [oscl_file_cache.h](#) defines the class [OsclFileCache](#).

```
#include "osclconfig_io.h"  
#include "oscl_base.h"  
#include "oscl_file_io.h"
```

Data Structures

- class [OsclFileCache](#)
- class [OsclFileCacheBuffer](#)

8.32.1 Detailed Description

The file [oscl_file_cache.h](#) defines the class [OsclFileCache](#).

8.33 oscl_file_dir_utils.h File Reference

The file `oscl_file_dir_utils.h` defines some unix-style directory ops.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
```

Data Structures

- struct `oscl_fsstat`
- struct `oscl_stat_buf`

Typedefs

- typedef `oscl_fsstat` `OSCL_FSSTAT`
- typedef `oscl_stat_buf` `OSCL_STAT_BUF`

Enumerations

- enum `OSCL_FILEMGMT_PERMS` { `OSCL_FILEMGMT_PERMS_READ` = 0x1, `OSCL_FILEMGMT_PERMS_WRITE` = 0x2, `OSCL_FILEMGMT_PERMS_EXECUTE` = 0x4 }
- enum `OSCL_FILEMGMT_MODES` { `OSCL_FILEMGMT_MODE_DIR` = 0x1 }
- enum `OSCL_FILEMGMT_ERR_TYPE` { `OSCL_FILEMGMT_E_OK` = 0, `OSCL_FILEMGMT_E_PATH_TOO_LONG`, `OSCL_FILEMGMT_E_PATH_NOT_FOUND`, `OSCL_FILEMGMT_E_ALREADY_EXISTS`, `OSCL_FILEMGMT_E_NOT_EMPTY`, `OSCL_FILEMGMT_E_PERMISSION_DENIED`, `OSCL_FILEMGMT_E_NO_MATCH`, `OSCL_FILEMGMT_E_UNKNOWN`, `OSCL_FILEMGMT_E_SYS_SPECIFIC`, `OSCL_FILEMGMT_E_NOT_IMPLEMENTED` }

Functions

- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_getcwd (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 [OsclFileHandle](#).

```
#include "osclconfig_io.h"  
#include "oscl_base.h"
```

Data Structures

- class [OsclFileHandle](#)

TypeDefs

- [typedef FILE * TOsclFileHandle](#)

8.35.1 Detailed Description

The file [oscl_file_handle.h](#) defines the class [OsclFileHandle](#).

8.36 oscl_file_io.h File Reference

The file [oscl_file_io.h](#) defines the class [Oscl_File](#). This is the public API to the basic file I/O operations.

```
#include "osclconfig_io.h"  
#include "oscl_base.h"  
#include "oscl_mem.h"  
#include "oscl_vector.h"  
#include "oscl_file_server.h"  
#include "oscl_file_find.h"  
#include "oscl_file_dir_utils.h"  
#include "oscl_file_handle.h"
```

Data Structures

- class [Oscl_File](#)
- class [OsclFixedCacheParam](#)
- class [OsclCacheObserver](#)

Defines

- #define [TOsclFileOffsetInt32](#) int32

8.36.1 Detailed Description

The file [oscl_file_io.h](#) defines the class [Oscl_File](#). This is the public API to the basic file I/O operations.

8.37 oscl_file_native.h File Reference

The file [oscl_file_native.h](#) defines the class [OsclNativeFile](#). This is the porting layer for basic file I/O operations.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_aostatus.h"
#include "oscl_file_io.h"
```

Data Structures

- class [OsclNativeFile](#)

8.37.1 Detailed Description

The file [oscl_file_native.h](#) defines the class [OsclNativeFile](#). This is the porting layer for basic file I/O operations.

8.38 oscl_file_server.h File Reference

The file [oscl_file_server.h](#) defines the class [Oscl_FileServer](#). This is the porting layer for file server implementations.

```
#include "osclconfig_io.h"  
#include "oscl_base.h"
```

Data Structures

- class [Oscl_FileServer](#)

8.38.1 Detailed Description

The file [oscl_file_server.h](#) defines the class [Oscl_FileServer](#). This is the porting layer for file server implementations.

8.39 oscl_file_stats.h File Reference

File stats class.

```
#include "oscl_base.h"
#include "osclconfig_io.h"
```

Data Structures

- class [OsclFileStats](#)
- class [OsclFileStatsItem](#)

Defines

- #define [OSCL_FILE_STATS_LOGGER_NODE](#) "OsclFileStats"

Enumerations

- enum [TOsclFileOp](#) { [EOsclFileOp_Open](#), [EOsclFileOp_Close](#), [EOsclFileOp_Read](#), [EOsclFileOp_Write](#), [EOsclFileOp_Seek](#), [EOsclFileOp_Tell](#), [EOsclFileOp_Size](#), [EOsclFileOp_Flush](#), [EOsclFileOp_EndOfFile](#), [EOsclFileOp_NativeOpen](#), [EOsclFileOp_NativeClose](#), [EOsclFileOp_NativeRead](#), [EOsclFileOp_NativeWrite](#), [EOsclFileOp_NativeSeek](#), [EOsclFileOp_NativeTell](#), [EOsclFileOp_NativeSize](#), [EOsclFileOp_NativeFlush](#), [EOsclFileOp_NativeEndOfFile](#), [EOsclFileOp_Last](#) }

8.39.1 Detailed Description

File stats class.

8.40 oscl_file_types.h File Reference

The file [oscl_file_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.

Data Structures

- class [OsclNativeFileParams](#)

Defines

- #define [OSCL_IO_FILENAME_MAXLEN](#) 512
- #define [OSCL_IO_EXTENSION_MAXLEN](#) 512
- #define [OSCL_FILE_WCHAR_PATH_DELIMITER](#) _STRLIT("/")
- #define [OSCL_FILE_CHAR_PATH_DELIMITER](#) _STRLIT_CHAR("/")

8.40.1 Detailed Description

The file [oscl_file_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.

8.41 oscl_heapbase.h File Reference

OSCL Heap Base include file.

```
#include "osclconfig_error.h"
#include "oscl_base.h"
#include "oscl_heapbase.inl"
```

Data Structures

- class [_OsclHeapBase](#)
- class [OsclTrapItem](#)

Typedefs

- [typedef void\(* OsclTrapOperation \)\(OsclAny *\)](#)

8.41.1 Detailed Description

OSCL Heap Base include file.

8.42 oscl_init.h File Reference

Global oscl initialization.

```
#include "oscl_base.h"  
#include <stdio.h>
```

Data Structures

- class [OsclInit](#)
- class [OsclSelect](#)

8.42.1 Detailed Description

Global oscl initialization.

8.43 oscl_int64_utils.h File Reference

```
#include "oscl_base.h"
```

Data Structures

- class [Oscl_Int64_Utils](#)
The Oscl_Int64_Utils class provides a wrapper for commonly used int64/uint64 operations.
- struct [OsclInteger64Transport](#)

Typedefs

- typedef [OsclInteger64Transport _OsclInteger64Transport](#)

8.43.1 Typedef Documentation

8.43.1.1 typedef struct [OsclInteger64Transport _OsclInteger64Transport](#)

[OsclInteger64Transport](#) Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

8.44 oscl_ip_socket.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

Data Structures

- class [OsclIPSocketI](#)

8.45 oscl_linked_list.h File Reference

The file [oscl_linked_list.h](#) defines the template class [Oscl_Linked_List](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
#include "oscl_opaque_type.h"
#include "oscl_assert.h"
```

Data Structures

- class [LinkedListElement](#)
- class [Oscl_Linked_List](#)
- class [Oscl_Linked_List_Base](#)
- class [Oscl_MTLinked_List](#)

8.45.1 Detailed Description

The file [oscl_linked_list.h](#) defines the template class [Oscl_Linked_List](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

8.46 oscl_lock_base.h File Reference

This file defines an abstract lock class, [OsclLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OsclNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OsclScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.

Data Structures

- class [OsclLockBase](#)
- class [OsclNullLock](#)
- class [OsclScopedLock](#)

The OsclScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope.

8.46.1 Detailed Description

This file defines an abstract lock class, [OsclLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OsclNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OsclScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.

8.47 oscl_map.h File Reference

The file `oscl_map.h` defines the template class `Oscl_Map` which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_tree.h"
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- struct `Oscl_Less`
- class `Oscl_Map`
- struct `Oscl_Select1st`
- class `value_compare`

Defines

- `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

8.47.1 Detailed Description

The file `oscl_map.h` defines the template class `Oscl_Map` which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

8.47.2 Define Documentation

8.47.2.1 `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

8.48 oscl_math.h File Reference

Provides math functions.

```
#include "osclconfig_util.h"  
#include "oscl_base.h"  
#include "oscl_math.inl"
```

Functions

- OSCL_COND_IMPORT_REF double `oscl_log` (double value)
- OSCL_COND_IMPORT_REF double `oscl_log10` (double value)
- OSCL_COND_IMPORT_REF double `oscl_sqrt` (double value)
- OSCL_COND_IMPORT_REF double `oscl_pow` (double x, double y)
- OSCL_COND_IMPORT_REF double `oscl_exp` (double value)
- OSCL_COND_IMPORT_REF double `oscl_sin` (double value)
- OSCL_COND_IMPORT_REF double `oscl_cos` (double value)
- OSCL_COND_IMPORT_REF double `oscl_tan` (double value)
- OSCL_COND_IMPORT_REF double `oscl_asin` (double value)
- OSCL_COND_IMPORT_REF double `oscl_atan` (double value)
- OSCL_COND_IMPORT_REF double `oscl_floor` (double value)

8.48.1 Detailed Description

Provides math functions.

8.49 oscl_media_data.h File Reference

Defines a container class for media data made up of a collection of memory fragments.

```
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.h"  
#include "oscl_media_status.h"
```

Data Structures

- class [BufferFragment](#)
- class [BufferMgr](#)
- class [BufferState](#)
- class [BuffFragGroup](#)
- class [MediaData](#)
- class [MemAllocator](#)

Typedefs

- typedef void(* [BufferFreeFuncPtr](#))(void *)
- typedef uint32 [MediaTimestamp](#)

8.49.1 Detailed Description

Defines a container class for media data made up of a collection of memory fragments.

8.50 oscl_media_status.h File Reference

Defines a status values for the [MediaData](#) containers.

Data Structures

- class [BufFragStatusClass](#)
- class [MediaStatusClass](#)

Variables

- const int32 [APPEND_MEDIA_AT_END](#) = -1

8.50.1 Detailed Description

Defines a status values for the [MediaData](#) containers.

8.51 oscl_mem.h File Reference

This file contains basic memory definitions for common use across platforms.

```
#include "osclconfig_memory.h"
#include "oscl_base.h"
#include "oscl_types.h"
#include "oscl_assert.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_lock_base.h"
#include "osclconfig_compiler_warnings.h"
#include "oscl_mem_inst.h"
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_refcounter.h"
#include "oscl_error.h"
#include "oscl_exception.h"
#include "oscl_mem.inl"
```

Data Structures

- class [HeapBase](#)
- class [OsclAuditCB](#)
- class [OsclMem](#)
- class [OsclMemAllocator](#)
- class [OsclMemAllocDestructDealloc](#)
- class [OsclMemBasicAllocator](#)
- class [OsclMemBasicAllocDestructDealloc](#)
- class [OsclMemGlobalAuditObject](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [OSCL_HAS_GLOBAL_NEW_DELETE](#) 1
- #define [OSCL_CLEANUP_BASE_CLASS\(T\)](#) _OSCL_CLEANUP_BASE_CLASS(T)
- #define [OSCL_ALLOC_NEW\(T_allocator, T, params\)](#) new(T_allocator.allocate(1)) T params
- #define [OSCL_TRAP_ALLOC_NEW\(T_ptr, T_allocator, T, params\)](#) _OSCL_TRAP_NEW(T_allocator.allocate(1),T_allocator.deallocate,T_ptr,T,params)
- #define [OSCL_ALLOC_DELETE\(ptr, T_allocator, T\)](#)
- #define [OSCL_MALLOC\(count\)](#) _oscl_default_audit_malloc(count)
- #define [oscl_malloc\(a\)](#) OSCL_MALLOC(a)
- #define [OSCL_DEFAULT_MALLOC\(x\)](#) OSCL_MALLOC(x)
- #define [OSCL_AUDIT_MALLOC\(auditCB, count\)](#) _oscl_audit_malloc(count, auditCB)
- #define [OSCL_CALLOC\(num, size\)](#) _oscl_default_audit_calloc(num,size)
- #define [oscl_calloc\(a, b\)](#) OSCL_CALLOC(a,b)

- #define **OSCL_AUDIT_CALLOC**(auditCB, num, size) _oscl_audit_malloc(num,size, auditCB)
- #define **OSCL_REALLOC**(ptr, new_size) _oscl_default_audit_realloc(ptr,new_size)
- #define **oscl_realloc**(a, b) OSCL_REALLOC(a,b)
- #define **OSCL_AUDIT_REALLOC**(auditCB, ptr, new_size) _oscl_audit_realloc(ptr,new_size, auditCB)
- #define **OSCL_FREE**(ptr) _oscl_audit_free(ptr)
- #define **oscl_free**(x) OSCL_FREE(x)
- #define **OSCL_DEFAULT_FREE**(x) OSCL_FREE(x)
- #define **OSCL_NEW**(T, params) new T params
- #define **OSCL_PLACEMENT_NEW**(ptr, constructor) new(ptr) constructor
- #define **OSCL_TRAP_NEW**(T_ptr, T, params) _OSCL_TRAP_NEW(_oscl_default_audit_new(sizeof(T)),_oscl_audit_free,T_ptr,T,params)
- #define **OSCL_AUDIT_NEW**(auditCB, T, params) new(_oscl_audit_new(sizeof(T),auditCB)) T params
- #define **OSCL_TRAP_AUDIT_NEW**(T_ptr, auditCB, T, params) _OSCL_TRAP_NEW(_oscl_audit_new(sizeof(T),auditCB),_oscl_audit_free,T_ptr,T,params)
- #define **OSCL_DELETE**(ptr)
- #define **OSCL_AUDIT_ARRAY_NEW**(auditCB, T, count) new(_oscl_audit_new(sizeof(T)*(count),auditCB)) T
- #define **OSCL_ARRAY_NEW**(T, count) new T[count]
- #define **OSCL_ARRAY_DELETE**(ptr) delete [] ptr
- #define **OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE**
- #define **_OSCL_TRAP_NEW**(exp, freeFunc, T_ptr, T, params)
- #define **_OSCL_CLEANUP_BASE_CLASS**(T) this → T::~T()

Functions

- **OSCL_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_malloc** (**size_t**, **size_t**, **OsclAuditCB** &, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_audit_realloc** (**void** *, **size_t**, **OsclAuditCB** &, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_audit_new** (**size_t**, **OsclAuditCB** &, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_default_audit_malloc** (**size_t**, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_default_audit_malloc** (**size_t**, **size_t**, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_default_audit_realloc** (**void** *, **size_t**, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_default_audit_new** (**size_t**, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void** **_oscl_audit_free** (**void** *)
- **void *** **operator new** (**size_t** aSize, **const char** *aFile, **int** aLine)
- **void *** **operator new** (**size_t** aSize)
- **void operator delete** (**void** *aPtr)
- **void *** **operator new[]** (**size_t** aSize, **const char** *aFile, **int** aLine)
- **void *** **operator new[]** (**size_t** aSize)
- **void operator delete[]** (**void** *aPtr)

8.51.1 Detailed Description

This file contains basic memory definitions for common use across platforms.

This is the main entry point header file for the OSCL memory library. It should be the only one users directly include. Basic memory copy, compare, and move functions are defined here as well as the allocation functions.

8.51.2 Define Documentation

8.51.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

Previously this was in oscl_mem_imp.h

8.51.3 Function Documentation

8.51.3.1 void operator delete (void * *aPtr*) [inline]

8.51.3.2 void* operator new (size_t *aSize*) [inline]

8.52 oscl_mem_align.h File Reference

8.53 oscl_mem_audit.h File Reference

This file contains the definition and partial implementation of MM_Audit class.

```
#include "oscl_lock_base.h"
#include "oscl_base_alloc.h"
#include "oscl_tagtree.h"
#include "oscl_mem.h"
#include "oscl_mem_auto_ptr.h"
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- struct [MM_AllocInfo](#)
- struct [MM_AllocNode](#)
- struct [MM_AllocQueryInfo](#)
- class [MM_Audit_Imp](#)
- struct [MM_AuditOverheadStats](#)
- struct [MM_FailInsertParam](#)
- struct [MM_Stats_CB](#)
- struct [MM_Stats_t](#)
- class [OsclMemAudit](#)
- class [OsclMemStatsNode](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [MM_ALLOC_MAX_QUERY_FILENAME_LEN](#) 128
- #define [MM_ALLOC_MAX_QUERY_TAG_LEN](#) 64
- #define [MM_AUDIT_VALIDATE_BLOCK](#) 1
- #define [MM_AUDIT_PREFILL_FLAG](#) 0x1
- #define [MM_AUDIT_POSTFILL_FLAG](#) 0x2
- #define [MM_AUDIT_VALIDATE_ALL_HEAP_FLAG](#) 0x4
- #define [MM_AUDIT_VALIDATE_ON_FREE_FLAG](#) 0x8
- #define [MM_AUDIT_ALLOC_NODE_ENABLE_FLAG](#) 0x10
- #define [MM_AUDIT_SUPPRESS_FILENAME_FLAG](#) 0x20
- #define [DEFAULT_MM_AUDIT_MODE](#) 0

Typedefs

- typedef [OSCLMemAutoPtr< char, Oscl_TAlloc< char, OsclMemBasicAllocator > >](#) [MMAudit_CharAutoPtr](#)
- typedef [OSCLMemAutoPtr< uint8, Oscl_TAlloc< uint8, _OsclBasicAllocator > >](#) [MMAudit_Uint8AutoPtr](#)
- typedef [OSCLMemAutoPtr< MM_AllocNode, Oscl_TAlloc< MM_AllocNode, OsclMemBasicAllocator > >](#) [MM_AllocNodeAutoPtr](#)
- typedef [OSCLMemAutoPtr< OsclMemStatsNode, Oscl_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > >](#) [MM_StatsNodeTagTreeType](#)

- `typedef OSCLMemAutoPtr< OsclMemStatsNode, Oscl_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > > OsclMemStatsNodeAutoPtr`
- `typedef Oscl_TAlloc< MM_StatsNodeTagTreeType, OsclMemBasicAllocator > TagTree_Allocator`
- `typedef Oscl_TagTree< MM_StatsNodeTagTreeType, TagTree_Allocator > OsclTagTreeType`

8.53.1 Detailed Description

This file contains the definition and partial implementation of MM_Audit class.

8.53.2 Define Documentation

8.53.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

8.54 oscl_mem_audit_internals.h File Reference

This file contains the internal definitions for the mem audit library.

```
#include "oscl_base.h"
#include "oscl_mem_audit.h"
#include "oscl_mem_inst.h"
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- struct [MM_AllocBlockFence](#)
- struct [MM_AllocBlockHdr](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [MM_AUDIT_ALLOC_NODE_SUPPORT](#) 1
- #define [MM_AUDIT_FENCE_SUPPORT](#) 0
- #define [MM_AUDIT_INCLUDE_ALL_HEAP_VALIDATION](#) 1
- #define [MM_AUDIT_FILL_SUPPORT](#) 0
- #define [MM_AUDIT_FAILURE_SIMULATION_SUPPORT](#) 1
- #define [FENCE_PATTERN](#) 0xAA
- #define [MIN_FENCE_SIZE](#) 4
- #define [MEM_ALIGN_SIZE](#) 8
- #define [COMPUTE_MEM_ALIGN_SIZE](#)(x, y, z) (y+(((x+y)%z) ? (z - (x+y)%z) : 0))
- #define [DEFAULT_PREFILL_PATTERN](#) 0x96
- #define [DEFAULT_POSTFILL_PATTERN](#) 0x5A

8.54.1 Detailed Description

This file contains the internal definitions for the mem audit library.

8.54.2 Define Documentation

8.54.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

8.55 oscl_mem_auto_ptr.h File Reference

This file defines the oscl_mem_auto_ptr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

```
#include "osclconfig_memory.h"  
#include "osclconfig_compiler_warnings.h"  
#include "oscl_mem.h"
```

Data Structures

- class [OSCLMemAutoPtr](#)

The oscl_auto_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the oscl_auto_ptr expires, its destructor uses delete to free the memory.

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT](#)

8.55.1 Detailed Description

This file defines the oscl_mem_auto_ptr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

8.55.2 Define Documentation

8.55.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

8.56 oscl_mem_basic_functions.h File Reference

This file contains prototypes for the basic memory functions.

```
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.inl"
```

Functions

- OSCL_COND_IMPORT_REF void * [_oscl_malloc](#) (int32 count)
- OSCL_COND_IMPORT_REF void * [_oscl_calloc](#) (int32 nelems, int32 size)
- OSCL_COND_IMPORT_REF void * [_oscl_realloc](#) (void *src, int32 count)
- OSCL_COND_IMPORT_REF void [_oscl_free](#) (void *src)
- OSCL_COND_IMPORT_REF void * [oscl_memcpy](#) (void *dest, const void *src, uint32 count)
- OSCL_COND_IMPORT_REF void * [oscl_memmove](#) (void *dest, const void *src, uint32 count)
- OSCL_COND_IMPORT_REF void * [oscl_memmove32](#) (void *dest, const void *src, uint32 count)
- OSCL_COND_IMPORT_REF void * [oscl_memset](#) (void *dest, uint8 val, uint32 count)
- OSCL_COND_IMPORT_REF int [oscl_memcmp](#) (const void *buf1, const void *buf2, uint32 count)

8.56.1 Detailed Description

This file contains prototypes for the basic memory functions.

8.57 oscl_mem_inst.h File Reference

The file defines default memory instrumentation level.

```
#include "osclconfig_memory.h"
```

Defines

- #define **PVMEM_INST_LEVEL** 1

8.57.1 Detailed Description

The file defines default memory instrumentation level.

8.58 oscl_mem_mempool.h File Reference

This file contains the definition of memory pool allocators.

```
#include "oscl_mem.h"  
#include "oscl_defalloc.h"  
#include "oscl_vector.h"
```

Data Structures

- struct [MemPoolBlockInfo](#)
- struct [MemPoolBufferInfo](#)
- class [OsclMemPoolFixedChunkAllocator](#)
- class [OsclMemPoolFixedChunkAllocatorObserver](#)
- class [OsclMemPoolResizableAllocator](#)
- class [OsclMemPoolResizableAllocatorMemoryObserver](#)
- class [OsclMemPoolResizableAllocatorObserver](#)

8.58.1 Detailed Description

This file contains the definition of memory pool allocators.

8.59 oscl_mempool_allocator.h File Reference

This file contains the definition of memory pool allocator for leave/trap.

```
#include "oscl_defalloc.h"
```

Data Structures

- class [OsclMemPoolAllocator](#)

8.59.1 Detailed Description

This file contains the definition of memory pool allocator for leave/trap.

8.60 oscl_mutex.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"  
#include "oscl_types.h"  
#include "oscl_base.h"  
#include "oscl_thread.h"  
#include "oscl_lock_base.h"
```

Data Structures

- class [OsclMutex](#)
- class [OsclThreadLock](#)

Typedefs

- typedef [OsclMutex OsclNoYieldMutex](#)

8.60.1 Detailed Description

This file provides implementation of mutex.

8.60.2 Typedef Documentation

8.60.2.1 typedef [OsclMutex OsclNoYieldMutex](#)

Class [OsclNoYieldMutex](#) can be used in use cases where there will be no CPU-yielding operation done while the Mutex is locked.

CPU-yielding operations include [OsclMutex::Lock](#), [OsclSemaphore::Wait](#), [OsclThread::Sleep](#), and [OsclBrewThreadUtil::BThreadYield](#).

The behavior of [OsclNoYieldMutex](#) depends on whether the threading model is pre-emptive or not. When threading is pre-emptive, it is identical to [OsclMutex](#). When threading is non-pre-emptive, it is a NO-OP.

An example of this type of use case is for simple data protection.

8.61 oscl_namestring.h File Reference

Name string class include file.

```
#include "oscl_base.h"
```

Data Structures

- class [OsclNameString](#)

8.61.1 Detailed Description

Name string class include file.

8.62 oscl_opaque_type.h File Reference

The file [oscl_opaque_type.h](#) defines pure virtual classes for working with opaque types.

```
#include "oscl_base.h"
```

Data Structures

- class [Oscl_Opaque_Type_Alloc](#)
- class [Oscl_Opaque_Type_Alloc_LL](#)
- class [Oscl_Opaque_Type_Compare](#)

8.62.1 Detailed Description

The file [oscl_opaque_type.h](#) defines pure virtual classes for working with opaque types.

8.63 oscl_priqueue.h File Reference

Implements a priority queue data structure similar to STL.

```
#include "oscl_base.h"  
#include "oscl_vector.h"
```

Data Structures

- class [OsclCompareLess](#)
- class [OsclPriorityQueue](#)
- class [OsclPriorityQueueBase](#)

8.63.1 Detailed Description

Implements a priority queue data structure similar to STL.

Implements a priority queue data structure similar to the STL class. The properties of the class include O(Log_2(N)) insertion and deletion complexity and O(1) complexity to access the top priority item.

8.64 oscl_procstatus.h File Reference

Data Structures

- class [OsclProcStatus](#)

8.65 oscl_queue.h File Reference

The file [oscl_queue.h](#) defines the template class [Oscl_Queue](#). It is similar to the STL::queue class, with some differences: - less complete - based on array rather than a deque - some interfaces modeled on oscl_vector, for ease of transition Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.h"  
#include "oscl_assert.h"  
#include "oscl_opaque_type.h"
```

Data Structures

- class [Oscl_Queue](#)
- class [Oscl_Queue_Base](#)

8.65.1 Detailed Description

The file [oscl_queue.h](#) defines the template class [Oscl_Queue](#). It is similar to the STL::queue class, with some differences: - less complete - based on array rather than a deque - some interfaces modeled on oscl_vector, for ease of transition Memory allocation is abstracted through the use of an allocator template parameter.

8.66 oscl_rand.h File Reference

Provides pseudo-random number generation.

```
#include "osclconfig_util.h"  
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.h"  
#include "oscl_rand.inl"
```

Data Structures

- class [OsclRand](#)

8.66.1 Detailed Description

Provides pseudo-random number generation.

8.67 oscl_refcounter.h File Reference

A general purpose reference counter to object lifetimes.

```
#include "oscl_assert.h"  
#include "oscl_defalloc.h"
```

Data Structures

- class [Oscl_DefAllocWithRefCounter](#)
- class [OsclRefCounter](#)
- class [OsclRefCounterDA](#)
- class [OsclRefCounterMTDA](#)
- class [OsclRefCounterMTSA](#)
- class [OsclRefCounterSA](#)

8.67.1 Detailed Description

A general purpose reference counter to object lifetimes.

8.68 oscl_refcounter_memfrag.h File Reference

This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount.

```
#include "oscl_base.h"  
#include "oscl_refcounter.h"
```

Data Structures

- class [OsclRefCounterMemFrag](#)

8.68.1 Detailed Description

This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount.

8.69 oscl_registry_access_client.h File Reference

Client-side implementation Registry Access implementation.

```
#include "oscl_registry_types.h"
#include "oscl_string_containers.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

Data Structures

- class [OsclRegistryAccessClient](#)

8.69.1 Detailed Description

Client-side implementation Registry Access implementation.

8.70 oscl_registry_client.h File Reference

Client-side implementation of OsclRegistry.

```
#include "oscl_registry_types.h"
#include "oscl_mem.h"
#include "oscl_string.h"
```

Data Structures

- class [OsclRegistryClient](#)

8.70.1 Detailed Description

Client-side implementation of OsclRegistry.

8.71 oscl_registry_client_impl.h File Reference

Client-side implementation of OsclRegistryInterface.

```
#include "oscl_base.h"
#include "osclconfig_proc.h"
#include "oscl_vector.h"
#include "oscl_string.h"
#include "oscl_registry_types.h"
#include "oscl_registry_serv_impl_tls.h"
```

Data Structures

- class [OsclRegistryAccessClientImpl](#)
- class [OsclRegistryAccessClientTlsImpl](#)
- class [OsclRegistryClientImpl](#)
- class [OsclRegistryClientTlsImpl](#)

8.71.1 Detailed Description

Client-side implementation of OsclRegistryInterface.

8.72 oscl_registry_serv_impl.h File Reference

Server-side implementation of OsclRegistry interfaces.

```
#include "oscl_base.h"
#include "osclconfig_proc.h"
#include "oscl_registry_types.h"
#include "oscl_string.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
```

Data Structures

- class [OsclComponentRegistry](#)
- class [OsclComponentRegistryData](#)
- class [OsclComponentRegistryElement](#)

8.72.1 Detailed Description

Server-side implementation of OsclRegistry interfaces.

8.73 oscl_registry_serv_impl_global.h File Reference

```
#include "osclconfig_proc.h"  
#include "oscl_base.h"
```

8.74 oscl_registry_serv_impl_tls.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_registry_serv_impl.h"
#include "oscl_registry_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

Data Structures

- class [OsclRegistryServTlsImpl](#)

8.75 oscl_registry_types.h File Reference

Common types used in Oscl registry interfaces.

```
#include "oscl_types.h"  
#include "oscl_string_containers.h"
```

Data Structures

- class [OsclRegistryAccessElement](#)

TypeDefs

- typedef [OsclAny](#) * [OsclComponentFactory](#)

8.75.1 Detailed Description

Common types used in Oscl registry interfaces.

8.76 oscl_scheduler.h File Reference

```
#include "oscl_scheduler_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_defalloc.h"
#include "oscl_mem.h"
```

Data Structures

- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclScheduler](#)
- class [OsclSchedulerObserver](#)
- class [PVSchedulerStopper](#)

Defines

- #define [PVSCHEDNAMELEN](#) 30

8.77 oscl_scheduler_ao.h File Reference

Oscl Scheduler user execution object classes.

```
#include "oscl_scheduler_aobase.h"
#include "oscl_mem.h"
#include "oscl_scheduler_types.h"
```

Data Structures

- class [OsclActiveObject](#)
- class [OsclTimerObject](#)

8.77.1 Detailed Description

Oscl Scheduler user execution object classes.

8.78 oscl_scheduler_aobase.h File Reference

Oscl Scheduler internal active object classes.

```
#include "oscl_namestring.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_scheduler_readyq.h"
#include "oscl_string_containers.h"
#include "oscl_scheduler_types.h"
```

Data Structures

- class [PVActiveBase](#)
- class [PVActiveStats](#)

Defines

- #define [OSCL_ZEROIZE](#)(ptr, size) oscl_memset(ptr, 0, size)
- #define [PVEEXECNAMELEN](#) 30

8.78.1 Detailed Description

Oscl Scheduler internal active object classes.

8.79 oscl_scheduler_readyq.h File Reference

ready q types for oscl scheduler

```
#include "oscl_scheduler_tuneables.h"
#include "oscl_priqueue.h"
#include "oscl_base_alloc.h"
#include "oscl_semaphore.h"
#include "oscl_mem.h"
#include "oscl_string_containers.h"
#include "oscl_scheduler_types.h"
#include "oscl_mutex.h"
```

Data Structures

- class [OsclReadyAlloc](#)
- class [OsclReadyCompare](#)
- class [OsclReadyQ](#)
- class [OsclTimerCompare](#)
- class [OsclTimerQ](#)
- class [TReadyQueLink](#)

Typedefs

- typedef [PVActiveBase](#) * TOsclReady

8.79.1 Detailed Description

ready q types for oscl scheduler

8.80 oscl_scheduler_threadcontext.h File Reference

Thread context functions needed by oscl scheduler.

```
#include "oscl_double_list.h"  
#include "oscl_mutex.h"  
#include "oscl_aostatus.h"
```

Data Structures

- class [PVThreadContext](#)

Enumerations

- enum [TPVThreadContext](#) { [EPVThreadContext_InThread](#), [EPVThreadContext_OsclThread](#), [EPVThreadContext_NonOsclThread](#), [EPVThreadContext_Undetermined](#) }

8.80.1 Detailed Description

Thread context functions needed by oscl scheduler.

8.81 oscl_scheduler_tuneables.h File Reference

Tuneable settings for Oscl Scheduler.

```
#include "osclconfig_proc.h"
```

Defines

- #define PV_SCHED_ENABLE_AO_STATS 1
- #define PV_SCHED_ENABLE_LOOP_STATS 0
- #define PV_SCHED_ENABLE_PERF_LOGGING 1
- #define PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS 1
- #define PV_SCHED_LOG_Q 0
- #define PV_SCHED_CHECK_Q 0
- #define PV_SCHED_FAIR_SCHEDULING 1
- #define OSCL_PERF_SUMMARY_LOGGING 0

8.81.1 Detailed Description

Tuneable settings for Oscl Scheduler.

8.82 oscl_scheduler_types.h File Reference

Scheduler common types include file.

```
#include "osclconfig_proc.h"  
#include "oscl_aostatus.h"  
#include "oscl_heapbase.h"
```

Data Structures

- class [OsclExecSchedulerBase](#)

8.82.1 Detailed Description

Scheduler common types include file.

8.83 oscl_semaphore.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"  
#include "oscl_thread.h"
```

Data Structures

- class [OsclSemaphore](#)

8.83.1 Detailed Description

This file provides implementation of mutex.

8.84 oscl_shared_ptr.h File Reference

This file defines a template class [OsclSharedPtr](#) which is a "smart pointer" to the parameterized type.

```
#include "oscl_base.h"
#include "oscl_refcounter.h"
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- class [OsclSharedPtr](#)
A parameterized smart pointer class.

Defines

- #define [OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT](#)

8.84.1 Detailed Description

This file defines a template class [OsclSharedPtr](#) which is a "smart pointer" to the parameterized type.

8.85 oscl_singleton.h File Reference

This file defines the [OsclSingleton](#) class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

Data Structures

- class [OsclSingleton](#)
- class [OsclSingletonRegistry](#)
- class [SingletonTable](#)

Variables

- const uint32 [OSCL_SINGLETON_ID_TEST](#) = 0
- const uint32 [OSCL_SINGLETON_ID_OSCLMEM](#) = 1
- const uint32 [OSCL_SINGLETON_ID_PVLOGGER](#) = 2
- const uint32 [OSCL_SINGLETON_ID_PVSCHEDULER](#) = 3
- const uint32 [OSCL_SINGLETON_ID_PVERRORTRAP](#) = 4
- const uint32 [OSCL_SINGLETON_ID_SDPMEDIAPARSER](#) = 5
- const uint32 [OSCL_SINGLETON_ID_PAYLOADPARSER](#) = 6
- const uint32 [OSCL_SINGLETON_ID_CPM_PLUGIN](#) = 7
- const uint32 [OSCL_SINGLETON_ID_PVMFRECOGNIZER](#) = 8
- const uint32 [OSCL_SINGLETON_ID_OSCLREGISTRY](#) = 9
- const uint32 [OSCL_SINGLETON_ID_OMX](#) = 10
- const uint32 [OSCL_SINGLETON_ID_OMXMASTERCORE](#) = 11
- const uint32 [OSCL_SINGLETON_ID_TICKCOUNT](#) = 12
- const uint32 [OSCL_SINGLETON_ID_WMDRMLOCK](#) = 13
- const uint32 [OSCL_SINGLETON_ID_LAST](#) = 14

8.85.1 Detailed Description

This file defines the [OsclSingleton](#) class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

[OsclSingleton](#) is initialized in [OsclBase::Init](#).

8.85.2 Variable Documentation

- 8.85.2.1 `const uint32 OSCL_SINGLETON_ID_CPM_PLUGIN = 7`
- 8.85.2.2 `const uint32 OSCL_SINGLETON_ID_LAST = 14`
- 8.85.2.3 `const uint32 OSCL_SINGLETON_ID_OMX = 10`
- 8.85.2.4 `const uint32 OSCL_SINGLETON_ID_OMXMASTERCORE = 11`
- 8.85.2.5 `const uint32 OSCL_SINGLETON_ID_OSCLMEM = 1`
- 8.85.2.6 `const uint32 OSCL_SINGLETON_ID_OSCLREGISTRY = 9`
- 8.85.2.7 `const uint32 OSCL_SINGLETON_ID_PAYLOADPARSER = 6`
- 8.85.2.8 `const uint32 OSCL_SINGLETON_ID_PVERRORTRAP = 4`
- 8.85.2.9 `const uint32 OSCL_SINGLETON_ID_PVLOGGER = 2`
- 8.85.2.10 `const uint32 OSCL_SINGLETON_ID_PVMFRECOGNIZER = 8`
- 8.85.2.11 `const uint32 OSCL_SINGLETON_ID_PVSCHEDULER = 3`
- 8.85.2.12 `const uint32 OSCL_SINGLETON_ID_SDPMEDIAPARSER = 5`
- 8.85.2.13 `const uint32 OSCL_SINGLETON_ID_TEST = 0`
- 8.85.2.14 `const uint32 OSCL_SINGLETON_ID_TICKCOUNT = 12`
- 8.85.2.15 `const uint32 OSCL_SINGLETON_ID_WMDRMLOCK = 13`

8.86 oscl_snprintf.h File Reference

Provides a portable implementation of snprintf.

```
#include "oscl_base.h"  
#include "osclconfig_util.h"
```

Functions

- OSCL_IMPORT_REF int32 [oscl_snprintf](#) (char *str, uint32 count, const char *fmt,...)
- OSCL_IMPORT_REF int32 [oscl_snprintf](#) ([oscl_wchar](#) *str, uint32 count, const [oscl_wchar](#) *fmt,...)
- OSCL_IMPORT_REF int32 [oscl_vsnprintf](#) (char *str, uint32 count, const char *fmt, va_list args)
- OSCL_IMPORT_REF int32 [oscl_vsnprintf](#) ([oscl_wchar](#) *str, uint32 count, const [oscl_wchar](#) *fmt, va_list args)

8.86.1 Detailed Description

Provides a portable implementation of snprintf.

8.87 oscl_socket.h File Reference

The file [oscl_socket.h](#) defines the OSCL Socket APIs.

```
#include "osclconfig_io.h"  
#include "oscl_heapbase.h"  
#include "oscl_defalloc.h"  
#include "oscl_vector.h"  
#include "oscl_mem.h"  
#include "oscl_socket_types.h"
```

Data Structures

- class [OsclSocketServ](#)
- class [OsclTCPSocket](#)
- class [OsclUDPSocket](#)

8.87.1 Detailed Description

The file [oscl_socket.h](#) defines the OSCL Socket APIs.

8.88 oscl_socket_accept.h File Reference

```
#include "oscl_socket_imp.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclAcceptMethod](#)
- class [OsclAcceptRequest](#)

8.89 oscl_socket_bind.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclBindMethod](#)
- class [OsclBindRequest](#)

8.90 oscl_socket_connect.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclConnectMethod](#)
- class [OsclConnectRequest](#)

8.91 oscl_socket_imp.h File Reference

```
#include "oscl_socket_tuneables.h"  
#include "oscl_socket_imp_pv.h"
```

8.92 oscl_socket_imp_base.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_request.h"
#include "oscl_defalloc.h"
#include "oscl_mutex.h"
#include "oscl_socket_stats.h"
#include "oscl_base.h"
```

Data Structures

- class [OsclSocketIBase](#)

8.93 oscl_socket_imp_pv.h File Reference

```
#include "oscl_socket_imp_base.h"
```

Data Structures

- class [OsclSocketI](#)

Defines

- #define [PVSOCK_ERR_BAD_PARAM](#) (-1)
- #define [PVSOCK_ERR SOCK_NOT_OPEN](#) (-2)
- #define [PVSOCK_ERR SOCK_NO_SERV](#) (-3)
- #define [PVSOCK_ERR SERV_NOT_CONNECTED](#) (-4)
- #define [PVSOCK_ERR SOCK_NOT_CONNECTED](#) (-5)
- #define [PVSOCK_ERR NOT_IMPLEMENTED](#) (-6)

8.93.1 Define Documentation

8.93.1.1 #define PVSOCK_ERR_BAD_PARAM (-1)

some error codes for request completion these are negative so they won't conflict with errors from the OS socket layer.

8.93.1.2 #define PVSOCK_ERR_NOT_IMPLEMENTED (-6)

8.93.1.3 #define PVSOCK_ERR_SERV_NOT_CONNECTED (-4)

8.93.1.4 #define PVSOCK_ERR SOCK_NO_SERV (-3)

8.93.1.5 #define PVSOCK_ERR SOCK_NOT_CONNECTED (-5)

8.93.1.6 #define PVSOCK_ERR SOCK_NOT_OPEN (-2)

8.94 oscl_socket_listen.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclListenMethod](#)
- class [OsclListenRequest](#)

Defines

- #define [OSCL_SOCKET_LISTEN_H_INCLUDEDd](#)

8.94.1 Define Documentation

8.94.1.1 #define OSCL_SOCKET_LISTEN_H_INCLUDEDd

8.95 oscl_socket_method.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_socket_request.h"
#include "pvlogger.h"
#include "oscl_socket_tuneables.h"
#include "oscl_ip_socket.h"
#include "oscl_socket_imp.h"
```

Data Structures

- class [OsclSocketMethod](#)
- class [OsclSocketRequestAO](#)

Defines

- #define [MSEC_TO_MICROSEC](#) 1000

8.95.1 Define Documentation

8.95.1.1 #define MSEC_TO_MICROSEC 1000

8.96 oscl_socket_recv.h File Reference

```
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclRecvMethod](#)
- class [OsclRecvRequest](#)

8.97 oscl_socket_recv_from.h File Reference

```
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclRecvFromMethod](#)
- class [OsclRecvFromRequest](#)

8.98 oscl_socket_request.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_tuneables.h"
```

Data Structures

- class [AcceptParam](#)
- class [BindParam](#)
- class [ConnectParam](#)
- class [ListenParam](#)
- class [OsclSocketRequest](#)
- class [PVSockBufRecv](#)
- class [PVSockBufSend](#)
- class [RecvFromParam](#)
- class [RecvParam](#)
- class [SendParam](#)
- class [SendToParam](#)
- class [ShutdownParam](#)
- class [SocketRequestParam](#)

8.99 oscl_socket_send.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclSendMethod](#)
- class [OsclSendRequest](#)

8.100 oscl_socket_send_to.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclSendToMethod](#)
- class [OsclSendToRequest](#)

8.101 oscl_socket_serv_imp.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_tuneables.h"
#include "oscl_socket_serv_imp_pv.h"
```

8.102 oscl_socket_serv_imp_base.h File Reference

```
#include "oscl_base.h"
#include "oscl_socket_stats.h"
```

Data Structures

- class [OsclSocketServIBase](#)

8.103 oscl_socket_serv_imp_pv.h File Reference

```
#include "oscl_socket_serv_imp_base.h"
#include "oscl_socket_serv_imp_reqlist.h"
#include "oscl_socket_tuneables.h"
#include "oscl_scheduler_ao.h"
```

Data Structures

- class [OsclSocketServI](#)

Defines

- #define [OSCL_READSET_FLAG](#) 0x04
- #define [OSCL_WRITESET_FLAG](#) 0x02
- #define [OSCL_EXCEPTSET_FLAG](#) 0x01

8.103.1 Define Documentation

8.103.1.1 #define OSCL_EXCEPTSET_FLAG 0x01

8.103.1.2 #define OSCL_READSET_FLAG 0x04

A bitmask for socket select operations

8.103.1.3 #define OSCL_WRITESET_FLAG 0x02

8.104 oscl_socket_serv_imp_reqlist.h File Reference

```
#include "oscl_socket_tuneables.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

Data Structures

- class [OsclSocketServRequestList](#)
- class [OsclSocketServRequestQElem](#)

8.105 oscl_socket_shutdown.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclShutdownMethod](#)
- class [OsclShutdownRequest](#)

8.106 oscl_socket_stats.h File Reference

```
#include "oscl_base.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
#include "oscl_socket_tuneables.h"
```

Enumerations

- enum TOsclSocketStatEvent { EOscSocket_RequestAO_Success, EOscSocket_RequestAO_Canceled, EOscSocket_RequestAO_Error, EOscSocket_RequestAO_Timeout, EOscSocket_ServRequestIssued, EOscSocket_ServPoll, EOscSocket_OS, EOscSocket_Readable, EOscSocket_Writable, EOscSocket_Except, EOscSocket_DataRecv, EOscSocket_DataSent, EOscSocket_ServRequestComplete, EOscSocket_ServRequestCancelIssued, EOscSocketServ_LoopsockOk, EOscSocketServ_LoopsockError }
- enum TOsclSocketServStatEvent { EOscSocketServ_SelectNoActivity = 0, EOscSocketServ_SelectActivity, EOscSocketServ_SelectRescheduleAsap, EOscSocketServ_SelectReschedulePoll, EOscSocketServ_LastEvent }

8.106.1 Enumeration Type Documentation

8.106.1.1 enum TOsclSocketServStatEvent

Enumeration values:

EOscSocketServ_SelectNoActivity
EOscSocketServ_SelectActivity
EOscSocketServ_SelectRescheduleAsap
EOscSocketServ_SelectReschedulePoll
EOscSocketServ_LastEvent

8.106.1.2 enum TOsclSocketStatEvent

Socket diagnostics.

Enumeration values:

EOscSocket_RequestAO_Success
EOscSocket_RequestAO_Canceled
EOscSocket_RequestAO_Error
EOscSocket_RequestAO_Timeout
EOscSocket_ServRequestIssued
EOscSocket_ServPoll
EOscSocket_OS
EOscSocket_Readable
EOscSocket_Writable

EOselSocket_Except
EOselSocket_DataRecv
EOselSocket_DataSent
EOselSocket_ServRequestComplete
EOselSocket_ServRequestCancelIssued
EOselSocketServ_LoopsockOk
EOselSocketServ_LoopsockError

8.107 oscl_socket_tuneables.h File Reference

```
#include "osclconfig_io.h"
#include "osclconfig_proc.h"
```

Defines

- #define PV_SOCKET_REQUEST_AO_PRIORITY OsclActiveObject::EPriorityNominal
- #define PV_OSCL_SOCKET_STATS_LOGGING 0
- #define PV_SOCKET_SERVER 1
- #define PV_SOCKET_SERVER_IS_THREAD OSCL_HAS_THREAD_SUPPORT
- #define PV_SOCKET_SERVER_SELECT 0
- #define PV_SOCKET_SERVER_THREAD_PRIORITY ThreadPriorityAboveNormal
- #define PV_SOCKET_SERVER_SELECT_TIMEOUT_MSEC (-1)
- #define PV_SOCKET_SERVER_SELECT_LOOPBACK_SOCKET 0
- #define PV_SOCKET_SERVER_AO_PRIORITY (OsclActiveObject::EPriorityNominal)
- #define PV_SOCKET_SERVER_AO_INTERVAL_MSEC 5
- #define PV_OSCL_SOCKET_SERVER_LOGGER_OUTPUT 0
- #define PV_OSCL_SOCKET_1MB_RECV_BUF 0
- #define PV_SOCKET_SERVI_STATS 0

8.107.1 Define Documentation

8.107.1.1 #define PV_OSCL_SOCKET_1MB_RECV_BUF 0

Set this to 0 or 1 to enable/disable setting the socket receive buffer size to 1 MB in the Bind call. This setting only affects PV socket server implementations.

When set to 1, the code will use the OsclSetRecvBufferSize macro to set the buffer size in the Bind call.

This setting was found to improve streaming performance on WinMobile devices, but should not generally be used.

8.107.1.2 #define PV_OSCL_SOCKET_SERVER_LOGGER_OUTPUT 0

Set this to 0 or 1 to enable/disable [PVLogger](#) output from PV socket server. Note that socket server logging will appear in a different file when running threaded mode of socket server. This is quite a bit of logging, so it should generally be disabled.

8.107.1.3 #define PV_OSCL_SOCKET_STATS_LOGGING 0

Set this to 0 or 1 to enable/disable socket stats logging with "OsclSocketStats" node. This feature is fairly costly so should be off in production code.

8.107.1.4 #define PV_SOCKET_REQUEST_AO_PRIORITY OsclActiveObject::EPriority-Nominal

PV_SOCKET_REQUEST_AO_PRIORITY sets the priority of the socket request completion AOs.

8.107.1.5 #define PV_SOCKET_SERVER 1

Enable/disable the PV socket server here.

8.107.1.6 #define PV_SOCKET_SERVER_AO_INTERVAL_MSEC 5

PV_SOCKET_SERVER_AO_INTERVAL_MSEC sets the AO scheduling interval of the PV socket server AO for non-threaded implementations.

8.107.1.7 #define PV_SOCKET_SERVER_AO_PRIORITY (OsclActiveObject::EPriority-Nominal)

PV_SOCKET_SERVER_AO_PRIORITY sets priority of the PV socket server AO for non-threaded implementations.

8.107.1.8 #define PV_SOCKET_SERVER_IS_THREAD OSCL_HAS_THREAD_SUPPORT

PV_SOCKET_SERVER_IS_THREAD chooses either the threaded or AO-based implementation of the PV socket server

8.107.1.9 #define PV_SOCKET_SERVER_SELECT 0

PV_SOCKET_SERVER_SELECT chooses whether to use "select" call or not. In threaded mode, select call is required and is forced to "1". In AO mode, "select" call is an option that defaults to "0". Avoiding any "select" call was found to greatly reduce CPU usage on WinMobile devices.

8.107.1.10 #define PV_SOCKET_SERVER_SELECT_LOOPBACK_SOCKET 0

PV_SOCKET_SERVER_SELECT_LOOPBACK_SOCKET enables the feature to wakeup the select call by writing to a loopback socket each time a new request comes in. This option is required to support the blocking select option of threaded server mode. This option is forced to "0" in AO mode.

8.107.1.11 #define PV_SOCKET_SERVER_SELECT_TIMEOUT_MSEC (-1)

PV_SOCKET_SERVER_SELECT_TIMEOUT_MSEC sets duration of the select call in the PV socket server thread for the polling select loop implementation. When the timeout is -1, the select call will block forever waiting on a new request and will use a loopback socket to signal a new request. Note: if infinite wait is selected, but loopback socket is not available, the implementation will poll at 10 msec intervals.

8.107.1.12 #define PV_SOCKET_SERVER_THREAD_PRIORITY ThreadPriorityAboveNormal

PV_SOCKET_SERVER_THREAD_PRIORITY sets the priority of the PV socket server thread.

8.107.1.13 #define PV_SOCKET_SERVI_STATS 0

For detailed performance breakdown of time spend in [OsclSocketServI](#) AO. Output is logged under "OsclSchedulerPerfStats" node. Should be off in production code. This option is forced to "0" in threaded mode.

8.108 oscl_socket_types.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_types.h"
#include "oscl_scheduler_types.h"
#include "oscl_namestring.h"
#include "oscl_stdstring.h"
```

Data Structures

- class [OsclNetworkAddress](#)
- class [OsclSocketObserver](#)

Defines

- #define [PVNETWORKADDRESS_LEN](#) 50

Enumerations

- enum [TPVSocketFxn](#) { [EPVSocketSend](#) = 0, [EPVSocketSendTo](#), [EPVSocketRecv](#), [EPVSocketRecvFrom](#), [EPVSocketConnect](#), [EPVSocketAccept](#), [EPVSocketShutdown](#), [EPVSocketBind](#), [EPVSocketListen](#), [EPVSocket_Last](#) }
- enum [TPVSocketEvent](#) { [EPVSocketSuccess](#), [EPVSocketPending](#), [EPVSocketTimeout](#), [EPVSocketFailure](#), [EPVSocketCancel](#) }
- enum [TPVSocketShutdown](#) { [EPVSocketSendShutdown](#), [EPVSocketRecvShutdown](#), [EPVSocketBothShutdown](#) }

8.108.1 Define Documentation

8.108.1.1 #define PVNETWORKADDRESS_LEN 50

8.108.2 Enumeration Type Documentation

8.108.2.1 enum TPVSocketEvent

Return codes for asynchronous APIs

Enumeration values:

EPVSocketSuccess
EPVSocketPending
EPVSocketTimeout
EPVSocketFailure
EPVSocketCancel

8.108.2.2 enum TPVSocketFxn

Enumeration values:

- EPVSocketSend**
- EPVSocketSendTo**
- EPVSocketRecv**
- EPVSocketRecvFrom**
- EPVSocketConnect**
- EPVSocketAccept**
- EPVSocketShutdown**
- EPVSocketBind**
- EPVSocketListen**
- EPVSocket_Last**

8.108.2.3 enum TPVSocketShutdown

Enumeration values:

- EPVSocketSendShutdown**
- EPVSocketRecvShutdown**
- EPVSocketBothShutdown**

8.109 oscl_stdstring.h File Reference

This file provides standard string operations such as strlen, strncpy, etc. ANSI defines undefined behavior when the destination pointer is null for operations such as strncpy, strncat, etc. But, we chose to define one. In such cases, we return the destination as null.

```
#include "oscl_base.h"
```

Functions

- OSCL_IMPORT_REF uint32 `oscl_strlen` (const char *str)
- OSCL_IMPORT_REF uint32 `oscl_strlen` (const `oscl_wchar` *str)
- OSCL_IMPORT_REF char * `oscl_strncpy` (char *dest, const char *src, uint32 count)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_strncpy` (`oscl_wchar` *dest, const `oscl_wchar` *src, uint32 count)
- OSCL_IMPORT_REF int32 `oscl_strcmp` (const char *str1, const char *str2)
- OSCL_IMPORT_REF int32 `oscl_stremp` (const `oscl_wchar` *str1, const `oscl_wchar` *str2)
- OSCL_IMPORT_REF int32 `oscl_strncmp` (const char *str1, const char *str2, uint32 count)
- OSCL_IMPORT_REF int32 `oscl_strncmp` (const `oscl_wchar` *str1, const `oscl_wchar` *str2, uint32 count)
- OSCL_IMPORT_REF char * `oscl_strncat` (char *dest, const char *src, uint32 count)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_strncat` (`oscl_wchar` *dest, const `oscl_wchar` *src, uint32 count)
- OSCL_IMPORT_REF const char * `oscl_strchr` (const char *str, int32 c)
- OSCL_IMPORT_REF char * `oscl_strchr` (char *str, int32 c)
- OSCL_IMPORT_REF const `oscl_wchar` * `oscl_strchr` (const `oscl_wchar` *str, int32 c)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_strchr` (`oscl_wchar` *str, int32 c)
- OSCL_IMPORT_REF const char * `oscl strrchr` (const char *str, int32 c)
- OSCL_IMPORT_REF char * `oscl strrchr` (char *str, int32 c)
- OSCL_IMPORT_REF const `oscl_wchar` * `oscl strrchr` (const `oscl_wchar` *str, int32 c)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl strrchr` (`oscl_wchar` *str, int32 c)
- OSCL_IMPORT_REF char * `oscl_strset` (char *dest, char val, uint32 count)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_strset` (`oscl_wchar` *dest, `oscl_wchar` val, uint32 count)
- OSCL_IMPORT_REF int32 `oscl_CIstrcmp` (const char *str1, const char *str2)
- OSCL_IMPORT_REF int32 `oscl_CIstrcmp` (const `oscl_wchar` *str1, const `oscl_wchar` *str2)
- OSCL_IMPORT_REF int32 `oscl_CIstrncmp` (const char *str1, const char *str2, uint32 count)
- OSCL_IMPORT_REF int32 `oscl_CIstrncmp` (const `oscl_wchar` *str1, const `oscl_wchar` *str2, uint32 count)
- OSCL_IMPORT_REF char `oscl_tolower` (const char car)
- OSCL_IMPORT_REF `oscl_wchar` `oscl_tolower` (const `oscl_wchar` car)
- OSCL_IMPORT_REF bool `oscl_isLetter` (const char car)
- OSCL_IMPORT_REF const char * `oscl_strstr` (const char *str1, const char *str2)
- OSCL_IMPORT_REF char * `oscl_strstr` (char *str1, const char *str2)
- OSCL_IMPORT_REF const `oscl_wchar` * `oscl_strstr` (const `oscl_wchar` *str1, const `oscl_wchar` *str2)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_strstr` (`oscl_wchar` *str1, const `oscl_wchar` *str2)
- OSCL_IMPORT_REF char * `oscl_streat` (char *dest, const char *src)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_streat` (`oscl_wchar` *dest, const `oscl_wchar` *src)

8.109.1 Detailed Description

This file provides standard string operations such as strlen, strncpy, etc. ANSI defines undefined behavior when the destination pointer is null for operations such as strncpy, strncat, etc. But, we chose to define one. In such cases, we return the destination as null.

8.110 oscl_str_ptr_len.h File Reference

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

```
#include "oscl_base.h"
#include "oscl_stdstring.h"
```

Data Structures

- struct [StrCSumPtrLen](#)
same as [StrPtrLen](#), but includes checksum field and method to speed up querying
- struct [StrPtrLen](#)
This data structure encapsulates a set of functions used to perform.
- struct [WStrPtrLen](#)
This data structure encapsulates a set of functions used to perform.

Typedefs

- typedef StrPtrLen [StrPtrLen](#)
This data structure encapsulates a set of functions used to perform.
- typedef WStrPtrLen [WStrPtrLen](#)
This data structure encapsulates a set of functions used to perform.
- typedef StrCSumPtrLen [StrCSumPtrLen](#)
same as [StrPtrLen](#), but includes checksum field and method to speed up querying
- typedef [WStrPtrLen](#) [OSCL_TStrPtrLen](#)

Variables

- const uint8 [OSCL_ASCII_CASE_MAGIC_BIT](#) = 0x20

8.110.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

8.111 oscl_string.h File Reference

Provides a standardized set of string containers that can be used in place of character arrays.

```
#include "oscl_base.h"  
#include "oscl_mem.h"
```

Data Structures

- class [OSCL_String](#)
- class [OSCL_wString](#)

Enumerations

- enum [TOSCL_StringOp](#) { [EOSCL_StringOp_CompressASCII](#), [EOSCL_StringOp_UTF16ToUTF8](#) }
- enum [TOSCL_wStringOp](#) { [EOSCL_wStringOp_ExpandASCII](#), [EOSCL_wStringOp_UTF8ToUTF16](#) }

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_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](#)

8.112.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

8.113 oscl_string_rep.h File Reference

Contains some internal implementation for string containers.

```
#include "oscl_defalloc.h"
```

Data Structures

- class [CFastRep](#)
- class [CHheapRep](#)
- class [CStackRep](#)

8.113.1 Detailed Description

Contains some internal implementation for string containers.

8.114 oscl_string_uri.h File Reference

Utilities to unescape URIs.

```
#include "oscl_base.h"  
#include "oscl_string.h"
```

Functions

- OSCL_IMPORT_REF bool [oscl_str_unescape_uri](#) (const char *str_buf_in, char *str_buf_out, uint32 max_out_buf_bytes, uint32 max_bytes, uint32 &out_buf_len)
unescape any of the special escape sequence in the uri string
- OSCL_IMPORT_REF bool [oscl_str_unescape_uri](#) (const [OSCL_String](#) &oscl_str_in, [OSCL_String](#) &oscl_str_out, uint32 &out_buf_len)
unescape any of the special escape sequence in the uri string

8.114.1 Detailed Description

Utilities to unescape URIs.

8.115 oscl_string_utf8.h File Reference

Utilities to validate and truncate UTF-8 encoded strings.

```
#include "oscl_base.h"
```

Functions

- OSCL_IMPORT_REF bool `oscl_str_is_valid_utf8` (const uint8 *str_buf, uint32 &num_valid_characters, uint32 max_bytes=0, uint32 max_char_2_valid=0, uint32 *num_byte_4_char=NULL)

Check if the input string contains any illegal UTF-8 character. The function scans the string and validate that each character is a valid utf-8. It stops at the first NULL character, invalid character or the max_byte value. The string is valid if and only if every character is a valid utf-8 character and the scanning stopped on a character boundary.

- OSCL_IMPORT_REF int32 `oscl_str_truncate_utf8` (uint8 *str_buf, uint32 max_char, uint32 max_bytes=0)

Truncates the UTF-8 string upto the required size.

8.115.1 Detailed Description

Utilities to validate and truncate UTF-8 encoded strings.

8.116 oscl_string_utils.h File Reference

Utilities to parse and convert strings.

```
#include "oscl_base.h"
```

Defines

- #define `oscl_isdigit(c)` ((c) >= '0' && (c) <= '9')

Functions

- OSCL_IMPORT_REF const char * `skip_whitespace` (const char *ptr)
- OSCL_IMPORT_REF char * `skip_whitespace` (char *ptr)
- OSCL_IMPORT_REF const char * `skip_whitespace` (const char *start, const char *end)
- OSCL_IMPORT_REF const char * `skip_to_whitespace` (const char *start, const char *end)
- OSCL_IMPORT_REF const char * `skip_to_line_term` (const char *start_ptr, const char *end_ptr)
- OSCL_IMPORT_REF const char * `skip_whitespace_and_line_term` (const char *start, const char *end)
- OSCL_IMPORT_REF int `extract_string` (const char *in_ptr, char *outstring, int maxsize)
- OSCL_IMPORT_REF int `extract_string` (const char *start, const char *end, char *outstring, int maxsize)
- OSCL_IMPORT_REF bool `PV_atoi` (const char *buf, const char new_format, uint32 &value)
- OSCL_IMPORT_REF bool `PV_atoi` (const char *buf, const char new_format, int length, uint32 &value)
- OSCL_IMPORT_REF bool `PV_atoi` (const char *buf, const char new_format, int length, `uint64` &value)
- OSCL_IMPORT_REF bool `PV_atof` (const char *buf, `OsclFloat` &value)
- OSCL_IMPORT_REF bool `PV_atof` (const char *buf, int length, `OsclFloat` &value)
- OSCL_IMPORT_REF int `oscl_abs` (int aVal)

8.116.1 Detailed Description

Utilities to parse and convert strings.

8.117 oscl_string_xml.h File Reference

Utilities to escape special characters in XML strings.

```
#include "oscl_base.h"
```

Functions

- OSCL_IMPORT_REF bool [oscl_str_need_escape_xml](#) (const char *str_buf, uint32 &num_escape_bytes, uint32 max_bytes=0)

Check if the input string contains any special ASCII character like &, <, >, ', ". The function scans the string and check if each character is a special character. It stops at the first NULL character (if max_bytes = 0), or the max_byte value.

- OSCL_IMPORT_REF int32 [oscl_str_escape_xml](#) (const char *str_buf_in, char *str_buf_out, uint32 max_out_buf_bytes, uint32 max_bytes=0, uint32 *num_bytes_written=NULL)

Escape any of the following special characters in the string Special ASCII characters: &, <, >, ', ".

8.117.1 Detailed Description

Utilities to escape special characters in XML strings.

8.118 oscl_tagtree.h File Reference

The file [oscl_tagtree.h](#) ...

```
#include "oscl_base.h"
#include "oscl_map.h"
#include "oscl_vector.h"
#include "oscl_stdstring.h"
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- struct [const_iterator](#)
- struct [iterator](#)
- struct [Node](#)
- struct [Oscl_Tag](#)
- struct [Oscl_Tag_Base](#)
- class [Oscl_TagTree](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)

8.118.1 Detailed Description

The file [oscl_tagtree.h](#) ...

8.118.2 Define Documentation

8.118.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

8.119 oscl_tcp_socket.h File Reference

```
#include "oscl_ip_socket.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_listen.h"
#include "oscl_socket_recv.h"
#include "oscl_socket_send.h"
#include "oscl_socket_accept.h"
#include "oscl_socket_shutdown.h"
#include "oscl_socket_connect.h"
#include "oscl_socket_bind.h"
```

Data Structures

- class [OsclTCPSocketI](#)

8.120 oscl_thread.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_procstatus.h"
#include "oscl_base.h"
```

Data Structures

- class [OsclThread](#)

Typedefs

- typedef [TOsclThreadFuncRet\(OSCL_THREAD_DECL * TOsclThreadFuncPtr \)\(TOsclThreadFuncArg\)](#)

Enumerations

- enum [OsclThread_State](#) { [Start_on_creation](#), [Suspend_on_creation](#) }
- enum [OsclThreadPriority](#) { [ThreadPriorityLowest](#), [ThreadPriorityLow](#), [ThreadPriorityBelowNormal](#), [ThreadPriorityNormal](#), [ThreadPriorityAboveNormal](#), [ThreadPriorityHighest](#), [ThreadPriorityTimeCritical](#) }

8.120.1 Detailed Description

. This file provides THREAD implementation that can be ported
to three OS LINUX, SYMBIAN, WIN32

8.120.2 Typedef Documentation

8.120.2.1 [typedef TOsclThreadFuncRet\(OSCL_THREAD_DECL * TOsclThreadFuncPtr\)\(TOsclThreadFuncArg\)](#)

8.120.3 Enumeration Type Documentation

8.120.3.1 enum [OsclThread_State](#)

Enumeration values:

[Start_on_creation](#)

[Suspend_on_creation](#)

8.120.3.2 enum [OsclThreadPriority](#)

Enumeration values:

[ThreadPriorityLowest](#)

[ThreadPriorityLow](#)

[ThreadPriorityBelowNormal](#)

ThreadPriorityNormal

ThreadPriorityAboveNormal

ThreadPriorityHighest

ThreadPriorityTimeCritical

8.121 oscl_tickcount.h File Reference

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

```
#include "oscl_base.h"
#include "oscl_tickcount.inl"
```

Data Structures

- class [OsclTickCount](#)

Defines

- #define [OSCLTICKCOUNT_MAX_TICKS](#) 0xffffffff

8.121.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

8.122 oscl_time.h File Reference

The file `oscl_time.h` defines two classes `NTPTime` and `TimeValue` for getting, manipulating, and formatting time values. The `TimeValue` class is based on the native system time format while `NTPTime` is used for the standard Network Time Protocol format.

```
#include "oscl_base.h"
#include "osclconfig_time.h"
#include "oscl_int64_utils.h"
#include "oscl_time.inl"
```

Data Structures

- class `NTPTime`

The NTPTime class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900.

- class `TimeValue`

The TimeValue class represents a time value in a format native to the system.

Typedefs

- typedef char `CtimeStrBuf [CTIME_BUFFER_SIZE]`
- typedef char `PV8601timeStrBuf [PV8601TIME_BUFFER_SIZE]`

Enumerations

- enum `TimeUnits` { `SECONDS` = 0, `MILLISECONDS` = 1, `MICROSECONDS` = 2 }

The TimeUnits enum can be used when constructing a `TimeValue` class.

Functions

- OSCL_IMPORT_REF void `PV8601ToRFC822` (`PV8601timeStrBuf` `pv8601_buffer`, `CtimeStrBuf` `ctime_buffer`)
- OSCL_IMPORT_REF void `RFC822ToPV8601` (`CtimeStrBuf` `ctime_buffer`, `PV8601timeStrBuf`)
- OSCL_COND_IMPORT_REF `TimeValue operator-` (`const TimeValue &a`, `const TimeValue &b`)

Variables

- const int `CTIME_BUFFER_SIZE` = 26
- const int `PV8601TIME_BUFFER_SIZE` = 21
- const long `USEC_PER_SEC` = 1000000
- const long `MSEC_PER_SEC` = 1000
- const uint32 `unix_ntp_offset` = 2208988800U

8.122.1 Detailed Description

The file [oscl_time.h](#) defines two classes [NTPTime](#) and [TimeValue](#) for getting, manipulating, and formatting time values. The [TimeValue](#) class is based on the native system time format while [NTPTime](#) is used for the standard Network Time Protocol format.

8.123 oscl_timer.h File Reference

```
#include "oscl_base.h"
#include "osclconfig_util.h"
#include "oscl_vector.h"
#include "oscl_tickcount.h"
#include "oscl_rand.h"
#include "oscl_scheduler_ao.h"
```

Data Structures

- struct [_TimerEntry](#)
- class [CallbackTimer](#)
- class [CallbackTimerObserver](#)
- class [OsclTimer](#)
- class [OsclTimerObserver](#)

8.124 oscl_tls.h File Reference

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

Data Structures

- class [OsclTLS](#)
- class [OsclTLSRegistry](#)
- class [TLSStorageOps](#)

Defines

- #define [OSCL_TLS_BASE_SLOTS](#) OSCL_TLS_ID_BASE_LAST +1
- #define [OSCL_TLS_EXTERNAL_SLOTS](#) 0
- #define [OSCL_TLS_MAX_SLOTS](#) (OSCL_TLS_BASE_SLOTS + OSCL_TLS_EXTERNAL_SLOTS)

Typedefs

- typedef [OsclAny](#) TOsclTlsKey

Variables

- const uint32 [OSCL_TLS_ID_MAGICNUM](#) = 0
- const uint32 [OSCL_TLS_ID_ERRORHOOK](#) = 1
- const uint32 [OSCL_TLS_ID_PVLOGGER](#) = 2
- const uint32 [OSCL_TLS_ID_TEST](#) = 3
- const uint32 [OSCL_TLS_ID_PVSCHEDULER](#) = 4
- const uint32 [OSCL_TLS_ID_PVERRORTRAP](#) = 5
- const uint32 [OSCL_TLS_ID_SDPMEDIAPARSER](#) = 6
- const uint32 [OSCL_TLS_ID_PAYLOADPARSER](#) = 7
- const uint32 [OSCL_TLS_ID_PVMFRECOGNIZER](#) = 8
- const uint32 [OSCL_TLS_ID_WMDRM](#) = 9
- const uint32 [OSCL_TLS_ID_OSCLREGISTRY](#) = 10
- const uint32 [OSCL_TLS_ID_SQLITE3](#) = 11
- const uint32 [OSCL_TLS_ID_BASE_LAST](#) = 11

8.125 oscl_tree.h File Reference

The file [oscl_tree.h](#) defines the template class [Oscl_Rb_Tree](#) which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the [Oscl_Map](#) class. Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_defalloc.h"  
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- struct [Oscl_Pair](#)
- class [Oscl_Rb_Tree](#)
- class [Oscl_Rb_Tree_Base](#)
- struct [Oscl_Rb_Tree_Const_Iterator](#)
- struct [Oscl_Rb_Tree_Iterator](#)
- struct [Oscl_Rb_Tree_Node](#)
- struct [Oscl_Rb_Tree_Node_Base](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)

8.125.1 Detailed Description

The file [oscl_tree.h](#) defines the template class [Oscl_Rb_Tree](#) which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the [Oscl_Map](#) class. Memory allocation is abstracted through the use of an allocator template parameter.

8.125.2 Define Documentation

8.125.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

8.126 oscl_types.h File Reference

This file contains basic type definitions for common use across platforms.

```
#include "osclconfig.h"
```

Data Structures

- struct [OsclMemoryFragment](#)

Typedefs

- [typedef int c_bool](#)
The c_bool type is mapped to an integer to provide a bool type for C interfaces.
- [typedef void OsclAny](#)
The OsclAny is meant to be used the context of a generic pointer (i.e., no specific type).
- [typedef char mbchar](#)
mbchar is multi-byte char (e.g., UTF-8) with null termination.
- [typedef unsigned int uint](#)
The uint type is a convenient abbreviation for unsigned int.
- [typedef uint8 octet](#)
The octet type is meant to be used for referring to a byte or collection bytes without suggesting anything about the underlying meaning of the bytes.
- [typedef float OsclFloat](#)
The Float type defined as OsclFloat.
- [typedef OSCL_NATIVE_INT64_TYPE int64](#)
- [typedef OSCL_NATIVE_UINT64_TYPE uint64](#)
- [typedef OSCL_NATIVE_WCHAR_TYPE oscl_wchar](#)
- [typedef oscl_wchar OSCL_TCHAR](#)
define OSCL_TCHAR

8.126.1 Detailed Description

This file contains basic type definitions for common use across platforms.

8.127 oscl_udp_socket.h File Reference

```
#include "oscl_ip_socket.h"
#include "oscl_defalloc.h"
#include "oscl_socket_recv_from.h"
#include "oscl_socket_send_to.h"
#include "oscl_socket_bind.h"
```

Data Structures

- class [OsclUDPSocketI](#)

8.128 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.

8.128.1 Detailed Description

Utilities to convert unicode to utf8 and vice versa.

8.129 oscl_uuid.h File Reference

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32.

```
#include "oscl_base.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_string_utils.h"
#include "oscl_stdstring.h"
```

Data Structures

- struct [OsclUuid](#)

Defines

- #define [BYTES_IN_UUID_ARRAY](#) 8

Typedefs

- typedef uint32 [OsclUid32](#)

Variables

- const char [PV_CHAR_CLOSE_BRACKET](#) = ')
- const char [PV_CHAR_COMMA](#) = ','

8.129.1 Detailed Description

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32.

8.129.2 Define Documentation

8.129.2.1 #define BYTES_IN_UUID_ARRAY 8

8.129.3 Typedef Documentation

8.129.3.1 typedef uint32 OsclUid32

8.129.4 Variable Documentation

8.129.4.1 const char PV_CHAR_CLOSE_BRACKET = ')

8.129.4.2 const char PV_CHAR_COMMA = ','

8.130 oscl_vector.h File Reference

The file `oscl_vector.h` defines the template class `Oscl_Vector` which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_mem_basic_functions.h"
#include "oscl_assert.h"
#include "oscl_opaque_type.h"
#include "oscl_defalloc.h"
#include "oscl_base.h"
```

Data Structures

- class `Oscl_Vector`
- class `Oscl_Vector_Base`

8.130.1 Detailed Description

The file `oscl_vector.h` defines the template class `Oscl_Vector` which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

8.131 osclconfig.h File Reference

This file contains configuration information for the linux platform.

```
#include <dirent.h>
#include <dlsfcn.h>
#include "osclconfig_limits_typedefs.h"
#include "osclconfig_unix_android.h"
#include "osclconfig_ix86.h"
#include "osclconfig_check.h"
```

Defines

- #define OSCL_HAS_ANDROID_SUPPORT 1
- #define OSCL_HAS_ANDROID_FILE_IO_SUPPORT 1
- #define OSCL_RELEASE_BUILD 0
- #define OSCL_UNSIGNED_CONST(x) x##u
- #define OSCL_NATIVE_UINT64_TYPE u_int64_t
- #define OSCL_TEMPLATED_DESTRUCTOR_CALL(type, simple_type) ~type()
- #define __TFS__ <>
- #define OSCL_HAS_PRAGMA_PACK 0
- #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_HAS_ANDROID_FILE_IO_SUPPORT 1

8.131.2.3 #define OSCL_HAS_ANDROID_SUPPORT 1

8.131.2.4 #define OSCL_NATIVE_UINT64_TYPE u_int64_t

8.131.2.5 #define OSCL_PACKED_STRUCT_BEGIN

8.131.2.6 #define OSCL_PACKED_STRUCT_END __attribute__((packed))

8.131.2.7 #define OSCL_PACKED_VAR(x) x __attribute__((packed))

8.131.2.8 #define OSCL_RELEASE_BUILD 0

8.131.2.9 #define OSCL_TEMPLATED_DESTRUCTOR_CALL(type, simple_type) ~type()

8.131.2.10 #define OSCL_UNSIGNED_CONST(x) x##u

8.132 osclconfig_ansi_memory.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include <memory.h>
```

Defines

- #define OSCL_HAS_ANSI_MEMORY_FUNCS 1

Typedefs

- typedef size_t oscl_memsize_t

8.132.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

8.132.2 Define Documentation

8.132.2.1 #define OSCL_HAS_ANSI_MEMORY_FUNCS 1

8.132.3 Typedef Documentation

8.132.3.1 typedef size_t oscl_memsize_t

8.133 osclconfig_check.h File Reference

Typedefs

- `typedef int8 __int8__check__`
- `typedef uint8 __uint8__check__`
- `typedef int16 __int16__check__`
- `typedef uint16 __uint16__check__`
- `typedef int32 __int32__check__`
- `typedef uint32 __uint32__check__`

8.134 osclconfig_compiler_warnings.h File Reference

This file contains the ability to turn off/on compiler warnings.

Defines

- #define OSCL_FUNCTION_PTR(x) (&x)

8.134.1 Detailed Description

This file contains the ability to turn off/on compiler warnings.

8.134.2 Define Documentation

8.134.2.1 #define OSCL_FUNCTION_PTR(x) (&x)

8.135 osclconfig_error.h File Reference

This file contains the common typedefs and header files needed to compile osclerror.

```
#include "osclconfig.h"  
#include <setjmp.h>  
#include <errno.h>  
#include "osclconfig_error_check.h"
```

Defines

- #define OSCL_HAS_EXCEPTIONS 1
- #define OSCL_HAS_ERRNO_H 1
- #define OSCL_HAS_SYMBIAN_ERRORTRAP 0
- #define OSCL_HAS_SETJMP_H 1

8.135.1 Detailed Description

This file contains the common typedefs and header files needed to compile osclerror.

8.135.2 Define Documentation

- 8.135.2.1 #define OSCL_HAS_ERRNO_H 1
- 8.135.2.2 #define OSCL_HAS_EXCEPTIONS 1
- 8.135.2.3 #define OSCL_HAS_SETJMP_H 1
- 8.135.2.4 #define OSCL_HAS_SYMBIAN_ERRORTRAP 0

8.136 osclconfig_error_check.h File Reference

8.137 osclconfig_global_new_delete.h File Reference

Functions

- void * [operator new](#) (size_t)
- void [operator delete](#) (void *)

8.138 osclconfig_global_placement_new.h File Reference

Functions

- void * [operator new](#) (size_t, void *ptr)

8.138.1 Function Documentation

8.138.1.1 void* operator new (size_t, void *ptr) [inline]

8.139 osclconfig_io.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include "osclconfig.h"
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <fcntl.h>
#include <signal.h>
#include <netdb.h>
#include <sys/mman.h>
#include <sys/types.h>
#include <errno.h>
#include <sys/vfs.h>
#include <dirent.h>
#include <sys/stat.h>
#include "osclconfig_io_check.h"
```

Defines

- #define OSCL_HAS_GLOB 0
- #define OSCL_HAS_ANSI_FILE_IO_SUPPORT 1
- #define OSCL_HAS_ANSI_64BIT_FILE_IO_SUPPORT 0
- #define OSCL_HAS_MSWIN_FILE_IO_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0
- #define OSCL_HAS_NATIVE_FILE_CACHE_ENABLE 1
- #define OSCL_FILE_BUFFER_MAX_SIZE 32768
- #define OSCL_HAS_PV_FILE_CACHE 0
- #define OSCL_HAS_LARGE_FILE_SUPPORT 1
- #define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0
- #define OSCL_HAS_SYMBIAN_DNS_SERVER 0
- #define OSCL_HAS_BERKELEY_SOCKETS 1
- #define OSCL_HAS_SOCKET_SUPPORT 1
- #define OsclValidInetAddr(addr) (inet_addr(addr)!=INADDR_NONE)
- #define OsclMakeSockAddr(sockaddr, port, addrstr, ok)
- #define OsclUnMakeSockAddr(sockaddr, addrstr) addrstr=inet_ntoa(sockaddr.sin_addr);
- #define OsclSetRecvBufferSize(s, val, ok, err)
- #define OsclBind(s, addr, ok, err)
- #define OsclJoin(s, addr, ok, err)
- #define OsclListen(s, size, ok, err)

- #define `OsclAccept`(s, accept_s, ok, err, wouldblock)
- #define `OsclSetNonBlocking`(s, ok, err)
- #define `OsclShutdown`(s, how, ok, err)
- #define `OsclSocket`(s, fam, type, prot, ok, err)
- #define `OsclSendTo`(s, buf, len, addr, ok, err, nbytes, wouldblock)
- #define `OsclSend`(s, buf, len, ok, err, nbytes, wouldblock)
- #define `OsclCloseSocket`(s, ok, err)
- #define `OsclConnect`(s, addr, ok, err, wouldblock)
- #define `OsclGetAsyncSockErr`(s, ok, err)
- #define `OsclConnectComplete`(s, wset, eset, success, fail, ok, err)
- #define `OsclRecv`(s, buf, len, ok, err, nbytes, wouldblock)
- #define `OsclRecvFrom`(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)
- #define `OsclSocketSelect`(nfds, rd, wr, ex, timeout, ok, err, nhandles)
- #define `OsclSocketStartup`(ok)
- #define `OsclSocketCleanup`(ok)
- #define `OsclGethostbyname`(name, hostent, ok, err)
- #define `OsclGetDottedAddr`(hostent, dottedaddr, ok)
- #define `OSCL_SD_RECEIVE` SHUT_RD
- #define `OSCL_SD_SEND` SHUT_WR
- #define `OSCL_SD_BOTH` SHUT_RDWR
- #define `OSCL_AF_INET` AF_INET
- #define `OSCL SOCK_STREAM` SOCK_STREAM
- #define `OSCL SOCK_DGRAM` SOCK_DGRAM
- #define `OSCL IPPROTO_TCP` IPPROTO_TCP
- #define `OSCL IPPROTO_UDP` IPPROTO_UDP
- #define `_FILE_OFFSET_BITS` 64

Typedefs

- typedef int `TOsclSocket`
- typedef sockaddr_in `TOsclSockAddr`
- typedef socklen_t `TOsclSockAddrLen`
- typedef hostent `TOsclHostent`
- typedef off_t `TOsclFileOffset`

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 _FILE_OFFSET_BITS 64`
- 8.139.2.2 `#define OSCL_AF_INET AF_INET`
- 8.139.2.3 `#define OSCL_FILE_BUFFER_MAX_SIZE 32768`
- 8.139.2.4 `#define OSCL_HAS_ANSI_64BIT_FILE_IO_SUPPORT 0`
- 8.139.2.5 `#define OSCL_HAS_ANSI_FILE_IO_SUPPORT 1`
- 8.139.2.6 `#define OSCL_HAS_BERKELEY_SOCKETS 1`
- 8.139.2.7 `#define OSCL_HAS_GLOB 0`
- 8.139.2.8 `#define OSCL_HAS_LARGE_FILE_SUPPORT 1`
- 8.139.2.9 `#define OSCL_HAS_MSWIN_FILE_IO_SUPPORT 0`
- 8.139.2.10 `#define OSCL_HAS_NATIVE_FILE_CACHE_ENABLE 1`
- 8.139.2.11 `#define OSCL_HAS_PV_FILE_CACHE 0`
- 8.139.2.12 `#define OSCL_HAS_SOCKET_SUPPORT 1`
- 8.139.2.13 `#define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0`
- 8.139.2.14 `#define OSCL_HAS_SYMBIAN_DNS_SERVER 0`
- 8.139.2.15 `#define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0`
- 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 OsclAccept(s, accept_s, ok, err, wouldblock)`

Value:

```
accept_s=accept(s,NULL,NULL); \
ok=(accept_s!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);} 
```

8.139.2.24 #define OsclBind(s, addr, ok, err)
Value:

```
TOsclSockAddr* tmpadr = &addr;\n    sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\n    ok=(bind(s,sadr,sizeof(addr))!=(-1));\n    if (!ok)err=errno
```

8.139.2.25 #define OsclCloseSocket(s, ok, err)
Value:

```
ok=(close(s)!=(-1));\n    if (!ok)err=errno
```

8.139.2.26 #define OsclConnect(s, addr, ok, err, wouldblock)
Value:

```
TOsclSockAddr* tmpadr = &addr;\n    sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\n    ok=(connect(s,sadr,sizeof(addr))!=(-1));\n    if (!ok){err=errno;wouldblock=(err==EINPROGRESS);}
```

8.139.2.27 #define OsclConnectComplete(s, wset, eset, success, fail, ok, err)
Value:

```
success=fail=false;\n    if (FD_ISSET(s,&eset))\\\n        {fail=true;OsclGetAsyncSockErr(s,ok,err);}\n    else if (FD_ISSET(s,&wset))\\\n        {OsclGetAsyncSockErr(s,ok,err);if (ok && err==0)success=true;else fail=true;}
```

8.139.2.28 #define OsclGetAsyncSockErr(s, ok, err)
Value:

```
int opterr;socklen_t optlen(sizeof(opterr));\n    ok=(getsockopt(s,SOL_SOCKET,SO_ERROR,(void *)&opterr,&optlen)!=(-1));\n    if(ok)err=opterr;else err=errno;
```

8.139.2.29 #define OsclGetDottedAddr(hostent, dottedaddr, ok)
Value:

```
long *_hostaddr=(long*)hostent->h_addr_list[0];\n    struct in_addr _inaddr;\n    _inaddr.s_addr=*(_hostaddr);\n    dottedaddr/inet_ntoa(_inaddr);\n    ok=(dottedaddr!=NULL);
```

8.139.2.30 #define OsclGethostbyname(name, hostent, ok, err)
Value:

```
hostent=gethostbyname((const char*)name); \
ok=(hostent!=NULL); \
if (!ok)err=errno;
```

8.139.2.31 #define OsclJoin(s, addr, ok, err)
Value:

```
{\
    struct ip_mreq mreq; \
    void* p = &addr; \
ok=(bind(s,(sockaddr*)p,sizeof(addr))!=(-1)); \
mreq.imr_multiaddr.s_addr = addr.sin_addr.s_addr ; \
mreq.imr_interface.s_addr = htonl(INADDR_ANY); \
ok=(setsockopt(s, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq, sizeof(struct ip_mreq))!=(-1)); \
if (!ok)err=errno; \
}
```

8.139.2.32 #define OsclListen(s, size, ok, err)
Value:

```
ok=(listen(iSocket,qSize)!=(-1)); \
if (!ok)err=errno
```

8.139.2.33 #define OsclMakeSockAddr(sockaddr, port, addrstr, ok)
Value:

```
sockaddr.sin_family=OSCL_AF_INET; \
sockaddr.sin_port=htons(port); \
int32 result/inet_aton((const char*)addrstr,&sockaddr.sin_addr); \
ok=(result!=0);
```

8.139.2.34 #define OsclRecv(s, buf, len, ok, err, nbytes, wouldblock)
Value:

```
nbytes=recv(s,(void *)buf),(size_t)(len),0); \
ok=(nbytes!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN);}
```

8.139.2.35 #define OsclRecvFrom(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)

Value:

```
\ 
void* p=paddr;\n
nbytes=recvfrom(s,(void*)(buf),(size_t)(len),0,(struct sockaddr*)p,paddrlen);\n
ok=(nbytes!=(-1));\n
if (!ok){err=errno;wouldblock=(err==EAGAIN);}\n
}
```

8.139.2.36 #define OsclSend(s, buf, len, ok, err, nbytes, wouldblock)

Value:

```
nbytes=send(s,(const void*)(buf),(size_t)(len),0);\n
ok=(nbytes!=(-1));\n
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

8.139.2.37 #define OsclSendTo(s, buf, len, addr, ok, err, nbytes, wouldblock)

Value:

```
TOsclSockAddr* tmpadr = &addr;\n
sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\n
nbytes=sendto(s,(const void*)(buf),(size_t)(len),0,sadr,(socklen_t)sizeof(addr));\n
ok=(nbytes!=(-1));\n
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

8.139.2.38 #define OsclSetNonBlocking(s, ok, err)

Value:

```
ok=(fcntl(s,F_SETFL,O_NONBLOCK)!=(-1));\n
if (!ok)err=errno
```

8.139.2.39 #define OsclSetRecvBufferSize(s, val, ok, err)

Value:

```
ok=(setsockopt(s,SOL_SOCKET,SO_RCVBUF,(char*)&val, sizeof(int)) !=-1);\n
if (!ok)err=errno
```

8.139.2.40 #define OsclShutdown(s, how, ok, err)

Value:

```
ok=(shutdown(iSocket,how)!=(-1));\n
if (!ok)err=errno
```

8.139.2.41 #define OsclSocket(s, fam, type, prot, ok, err)**Value:**

```
s=socket(fam,type,prot); \
ok=(s!=(-1)); \
if (!ok)err=errno
```

8.139.2.42 #define OsclSocketCleanup(ok)**Value:**

```
signal(SIGPIPE,SIG_DFL); \
ok=true
```

8.139.2.43 #define OsclSocketSelect(nfds, rd, wr, ex, timeout, ok, err, nhandles)**Value:**

```
nhandles=select(nfds,&rd,&wr,&ex,&timeout); \
ok=(nhandles!=(-1)); \
if (!ok)err=errno
```

8.139.2.44 #define OsclSocketStartup(ok)**Value:**

```
signal(SIGPIPE,SIG_IGN); \
ok=true
```

8.139.2.45 #define OsclUnMakeSockAddr(sockaddr, addrstr) addrstr=inet_ntoa(sockaddr.sin_addr);**8.139.2.46 #define OsclValidInetAddr(addr) (inet_addr(addr)!=INADDR_NONE)**

8.139.3 Typedef Documentation

8.139.3.1 typedef off_t TOsclFileOffset**8.139.3.2 typedef struct hostent TOsclHostent****8.139.3.3 typedef struct sockaddr_in TOsclSockAddr****8.139.3.4 typedef socklen_t TOsclSockAddrLen****8.139.3.5 typedef int TOsclSocket**

8.140 osclconfig_io_check.h File Reference

Typedefs

- `typedef TOsclFileOffset __verify__TOsclFileOffset__defined__`

8.140.1 Typedef Documentation

8.140.1.1 `typedef TOsclFileOffset __verify__TOsclFileOffset__defined__`

type `TOsclFileOffset` should be defined as the type used for file size and offsets on the target platform.
Example: `typedef size_t TOsclFileOffset;`

8.141 osclconfig_ix86.h File Reference

This file contains configuration information for the ix86 processor family.

Defines

- #define OSCL_INTEGERS_WORD_ALIGNED 1
- #define OSCL_BYTE_ORDER_BIG_ENDIAN 0
- #define OSCL_BYTE_ORDER_LITTLE_ENDIAN 1

8.141.1 Detailed Description

This file contains configuration information for the ix86 processor family.

8.142 osclconfig_lib.h File Reference

This file contains configuration information for the ANSI build.

```
#include "osclconfig_lib_check.h"
```

Defines

- #define OSCL_HAS_RUNTIME_LIB_LOADING_SUPPORT 1
- #define PV_RUNTIME_LIB_FILENAME_EXTENSION "so"
- #define OSCL_LIB_READ_DEBUG_LIBS 1
- #define PV_DYNAMIC_LOADING_CONFIG_FILE_PATH "./"

8.142.1 Detailed Description

This file contains configuration information for the ANSI build.

8.142.2 Define Documentation

8.142.2.1 #define OSCL_HAS_RUNTIME_LIB_LOADING_SUPPORT 1

8.142.2.2 #define OSCL_LIB_READ_DEBUG_LIBS 1

8.142.2.3 #define PV_DYNAMIC_LOADING_CONFIG_FILE_PATH "./"

8.142.2.4 #define PV_RUNTIME_LIB_FILENAME_EXTENSION "so"

8.143 osclconfig_lib_check.h File Reference

8.144 osclconfig_limits_typedefs.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include <limits.h>
```

Defines

- #define OSCL_CHAR_IS_UNSIGNED 1
- #define OSCL_CHAR_IS_SIGNED 0

8.144.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

8.144.2 Define Documentation

8.144.2.1 #define OSCL_CHAR_IS_SIGNED 0

8.144.2.2 #define OSCL_CHAR_IS_UNSIGNED 1

8.145 osclconfig_memory.h File Reference

```
#include "osclconfig.h"
#include "osclconfig_ansi_memory.h"
#include "osclconfig_memory_check.h"
```

Defines

- #define OSCL_BYPASS_MEMMGT 1
- #define OSCL_HAS_GLOBAL_NEW_DELETE 1
- #define PVMEM_INST_LEVEL 1
- #define OSCL_HAS_HEAP_BASE_SUPPORT 1
- #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0

8.145.1 Define Documentation

8.145.1.1 #define OSCL_BYPASS_MEMMGT 1

8.145.1.2 #define OSCL_HAS_GLOBAL_NEW_DELETE 1

8.145.1.3 #define OSCL_HAS_HEAP_BASE_SUPPORT 1

8.145.1.4 #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0

8.145.1.5 #define PVMEM_INST_LEVEL 1

8.146 osclconfig_memory_check.h File Reference

8.147 osclconfig_no_os.h File Reference

Defines

- #define OSCL_HAS_UNIX_SUPPORT 0
- #define OSCL_HAS_MSWIN_SUPPORT 0
- #define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_SUPPORT 0
- #define OSCL_HAS_SAVAJE_SUPPORT 0
- #define OSCL_HAS_PV_C_OS_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_ERRORTRAP 0
- #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0
- #define OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS 0
- #define OSCL_HAS_PV_C_OS_TIME_FUNCS 0
- #define OSCL_HAS_UNIX_TIME_FUNCS 0
- #define OSCL_HAS_SYMBIAN_TIMERS 0
- #define OSCL_HAS_SYMBIAN_MATH 0
- #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0
- #define OSCL_HAS_PTHREAD_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0
- #define OSCL_HAS_SAVAJE_IO_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0
- #define OSCL_HAS_SYMBIAN_DNS_SERVER 0
- #define OSCL_HAS_BERKELEY_SOCKETS 0

8.148 osclconfig_proc.h File Reference

This file contains configuration information for the linux platform.

```
#include "osclconfig.h"  
#include "osclconfig_proc_unix_android.h"  
#include "osclconfig_proc_check.h"
```

8.148.1 Detailed Description

This file contains configuration information for the linux platform.

8.149 osclconfig_proc_check.h File Reference

Typedefs

- `typedef TOsclThreadId __verify__TOsclThreadId_defined__`
- `typedef TOsclThreadFuncRet __verify__TOsclThreadFuncRet_defined__`
- `typedef TOsclThreadFuncArg __verify__TOsclThreadFuncArg_defined__`
- `typedef TOsclThreadObject __verify__TOsclThreadObject_defined__`
- `typedef TOsclMutexObject __verify__TOsclMutexObject_defined__`
- `typedef TOsclSemaphoreObject __verify__TOsclSemaphoreObject_defined__`
- `typedef TOsclConditionObject __verify__TOsclConditionObject_defined__`

8.149.1 Typedef Documentation

8.149.1.1 `typedef TOsclConditionObject __verify__TOsclConditionObject_defined__`

type `TOsclConditionObject` should be defined as the type used as a condition variable on the target platform.
Example: `typedef pthread_cond_t TOsclConditionObject;`

Note: Condition variables are only used with certain semaphore implementations. If the semaphore implementation does not require a condition variable, then this type can be defined as 'int' as follows: `typedef int TOsclConditionObject; //not used`

8.149.1.2 `typedef TOsclMutexObject __verify__TOsclMutexObject_defined__`

type `TOsclMutexObject` should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef pthread_mutex_t TOsclMutexObject;`

8.149.1.3 `typedef TOsclSemaphoreObject __verify__TOsclSemaphoreObject_defined__`

type `TOsclSemaphoreObject` should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef sem_t TOsclSemaphoreObject;`

8.149.1.4 `typedef TOsclThreadFuncArg __verify__TOsclThreadFuncArg_defined__`

type `TOsclThreadFuncArg` should be defined as the type used as a thread function argument on the target platform. Example: `typedef LPVOID TOsclThreadFuncArg;`

8.149.1.5 `typedef TOsclThreadFuncRet __verify__TOsclThreadFuncRet_defined__`

type `TOsclThreadFuncRet` should be defined as the type used as a thread function return value on the target platform. Example: `typedef DWORD TOsclThreadFuncRet;`

8.149.1.6 `typedef TOsclThreadId __verify__TOsclThreadId_defined__`

type `TOsclThreadId` should be defined as the type used as a thread ID on the target platform. Example: `typedef DWORD TOsclThreadId;`

8.149.1.7 `typedef TOsclThreadObject __verify__TOsclThreadObject_defined__`

type `TOsclThreadObject` should be defined as the type used as a thread object or handle on the target platform. Example: `typedef pthread_t TOsclThreadObject;`

8.150 osclconfig_proc_unix_android.h File Reference

```
#include <pthread.h>
#include <errno.h>
#include <signal.h>
```

Defines

- #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- #define OSCL_HAS_THREAD_SUPPORT 1
- #define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0
- #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0
- #define OSCL_HAS_PTHREAD_SUPPORT 1
- #define OSCL_THREAD_DECL

TypeDefs

- typedef pthread_t TOsclThreadId
- typedef void * TOsclThreadFuncArg
- typedef void * TOsclThreadFuncRet
- typedef pthread_t TOsclThreadObject
- typedef pthread_mutex_t TOsclMutexObject
- typedef int TOsclSemaphoreObject
- typedef pthread_cond_t TOsclConditionObject

8.150.1 Define Documentation

- 8.150.1.1 `#define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0`
- 8.150.1.2 `#define OSCL_HAS_PTHREAD_SUPPORT 1`
- 8.150.1.3 `#define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0`
- 8.150.1.4 `#define OSCL_HAS_SYMBIAN_SCHEDULER 0`
- 8.150.1.5 `#define OSCL_HAS_THREAD_SUPPORT 1`
- 8.150.1.6 `#define OSCL_THREAD_DECL`

8.150.2 Typedef Documentation

- 8.150.2.1 `typedef pthread_cond_t TOsclConditionObject`
- 8.150.2.2 `typedef pthread_mutex_t TOsclMutexObject`
- 8.150.2.3 `typedef int TOsclSemaphoreObject`
- 8.150.2.4 `typedef void* TOsclThreadFuncArg`
- 8.150.2.5 `typedef void* TOsclThreadFuncRet`
- 8.150.2.6 `typedef pthread_t TOsclThreadId`
- 8.150.2.7 `typedef pthread_t TOsclThreadObject`

8.151 osclconfig_proc_unix_common.h File Reference

```
#include <time.h>
#include <semaphore.h>
#include <pthread.h>
#include <errno.h>
```

Defines

- #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- #define OSCL_HAS_THREAD_SUPPORT 1
- #define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0
- #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 1
- #define OSCL_HAS_PTHREAD_SUPPORT 1
- #define OSCL_THREAD_DECL

Typedefs

- typedef pthread_t TOsclThreadId
- typedef void * TOsclThreadFuncArg
- typedef void * TOsclThreadFuncRet
- typedef pthread_t TOsclThreadObject
- typedef pthread_mutex_t TOsclMutexObject
- typedef sem_t TOsclSemaphoreObject
- typedef pthread_cond_t TOsclConditionObject

8.151.1 Define Documentation

- 8.151.1.1 #define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0
- 8.151.1.2 #define OSCL_HAS_PTHREAD_SUPPORT 1
- 8.151.1.3 #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 1
- 8.151.1.4 #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- 8.151.1.5 #define OSCL_HAS_THREAD_SUPPORT 1
- 8.151.1.6 #define OSCL_THREAD_DECL

8.151.2 Typedef Documentation

- 8.151.2.1 typedef pthread_cond_t TOsclConditionObject
- 8.151.2.2 typedef pthread_mutex_t TOsclMutexObject
- 8.151.2.3 typedef sem_t TOsclSemaphoreObject
- 8.151.2.4 typedef void* TOsclThreadFuncArg
- 8.151.2.5 typedef void* TOsclThreadFuncRet
- 8.151.2.6 typedef pthread_t TOsclThreadId
- 8.151.2.7 typedef pthread_t TOsclThreadObject

8.152 osclconfig_time.h File Reference

```
#include "osclconfig.h"
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include "osclconfig_time_check.h"
```

Defines

- #define OSCL_HAS_UNIX_TIME_FUNCS 1

TypeDefs

- typedef timeval OsclBasicTimeStruct
- typedef tm OsclBasicDateTimeStruct

8.152.1 Define Documentation

8.152.1.1 #define OSCL_HAS_UNIX_TIME_FUNCS 1

8.152.2 TypeDef Documentation

8.152.2.1 typedef tm OsclBasicDateTimeStruct

8.152.2.2 typedef struct timeval OsclBasicTimeStruct

8.153 osclconfig_time_check.h File Reference

Typedefs

- `typedef OsclBasicTimeStruct __Validate__BasicTimeStruct__`
- `typedef OsclBasicDateTimeStruct __Validate__BasicTimeDateStruct__`

8.153.1 Typedef Documentation

8.153.1.1 `typedef OsclBasicDateTimeStruct __Validate__BasicTimeDateStruct__`

`OsclBasicDateTimeStruct` type should be defined to the platform-specific date + time type.

8.153.1.2 `typedef OsclBasicTimeStruct __Validate__BasicTimeStruct__`

`OsclBasicTimeStruct` type should be defined to the platform-specific time of day type.

8.154 osclconfig_unix_android.h File Reference

```
#include <stdlib.h>
#include <stdarg.h>
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ctype.h>
#include <math.h>
```

Defines

- #define OSCL_DISABLE_INLINES 0
- #define OSCL_HAS_ANSI_STDLIB_SUPPORT 1
- #define OSCL_HAS_ANSI_MATH_SUPPORT 1
- #define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1
- #define OSCL_HAS_ANSI_STRING_SUPPORT 1
- #define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 0
- #define OSCL_HAS_ANSI_STDIO_SUPPORT 1
- #define OSCL_MEMFRAG_PTR_BEFORE_LEN 1
- #define OSCL_HAS_UNIX_SUPPORT 1
- #define OSCL_HAS_MSWIN_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_SUPPORT 0
- #define OSCL_HAS_NATIVE_INT64_TYPE 1
- #define OSCL_HAS_NATIVE_UINT64_TYPE 1
- #define OSCL_NATIVE_INT64_TYPE int64_t
- #define OSCL_NATIVE_UINT64_TYPE uint64_t
- #define INT64(x) x##LL
- #define UINT64(x) x##ULL
- #define INT64_HILO(high, low) (((high##LL))<<32)|low)
- #define UINT64_HILO(high, low) (((high##ULL))<<32)|low)
- #define OSCL_HAS_UNICODE_SUPPORT 1
- #define OSCL_NATIVE_WCHAR_TYPE wchar_t
- #define _STRLIT(x) L ## x
- #define _STRLIT_CHAR(x) x
- #define _STRLIT_WCHAR(x) L ## x
- #define OSCL_HAS_TLS_SUPPORT 1
- #define OSCL_TLS_IS_KEYED 1
- #define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)
- #define OSCL_TLS_KEY_DELETE_FUNC(key) pthread_key_delete(key)
- #define OSCL_TLS_STORE_FUNC(key, ptr) (pthread_setspecific(key,(const void*)ptr)==0)
- #define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)
- #define OSCL_HAS_BASIC_LOCK 1

Typedefs

- `typedef pthread_key_t TOsclTlsKey`
- `typedef pthread_mutex_t TOsclBasicLockObject`

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_MSWIN_SUPPORT 0`

8.154.1.15 `#define OSCL_HAS_NATIVE_INT64_TYPE 1`

8.154.1.16 `#define OSCL_HAS_NATIVE_UINT64_TYPE 1`

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_SYMBIAN_SUPPORT 0
- #define OSCL_HAS_NATIVE_INT64_TYPE 1
- #define OSCL_HAS_NATIVE_UINT64_TYPE 1
- #define OSCL_NATIVE_INT64_TYPE int64_t
- #define OSCL_NATIVE_UINT64_TYPE uint64_t
- #define INT64(x) x##LL
- #define UINT64(x) x##ULL
- #define INT64_HILO(high, low) (((high##LL))<<32)|low)
- #define UINT64_HILO(high, low) (((high##ULL))<<32)|low)
- #define OSCL_HAS_UNICODE_SUPPORT 1
- #define OSCL_NATIVE_WCHAR_TYPE wchar_t
- #define _STRLIT(x) L ## x
- #define _STRLIT_CHAR(x) x
- #define _STRLIT_WCHAR(x) L ## x
- #define OSCL_HAS_TLS_SUPPORT 1
- #define OSCL_TLS_IS_KEYED 1
- #define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)
- #define OSCL_TLS_KEY_DELETE_FUNC(key) pthread_key_delete(key)
- #define OSCL_TLS_STORE_FUNC(key, ptr) (pthread_setspecific(key,(const void*)ptr)==0)
- #define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)
- #define OSCL_HAS_BASIC_LOCK 1

Typedefs

- `typedef pthread_key_t TOsclTlsKey`
- `typedef pthread_mutex_t TOsclBasicLockObject`

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_SUPPORT 0

8.155.1.15 #define OSCL_HAS_NATIVE_INT64_TYPE 1

8.155.1.16 #define OSCL_HAS_NATIVE_UINT64_TYPE 1

8.155.1.17 #define OSCL_HAS_SYMBIAN_SUPPORT 0

8.155.1.18 #define OSCL_HAS_TLS_SUPPORT 1

8.155.1.19 #define OSCL_HAS_UNICODE_SUPPORT 1

8.155.1.20 #define OSCL_HAS_UNIX_SUPPORT 1

8.155.1.21 #define OSCL_MEMFRAG_PTR_BEFORE_LEN 1

8.155.1.22 #define OSCL_NATIVE_INT64_TYPE int64_t

8.155.1.23 #define OSCL_NATIVE_UINT64_TYPE uint64_t

8.155.1.24 #define OSCL_NATIVE_WCHAR_TYPE wchar_t

8.155.1.25 #define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)

8.155.1.26 #define OSCL_TLS_IS_KEYED 1

8.155.1.27 #define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)

8.156 osclconfig_util.h File Reference

```
#include "osclconfig.h"
#include <stdio.h>
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include "osclconfig_util_check.h"
```

Defines

- #define OSCL_CLOCK_HAS_DRIFT_CORRECTION 0
- #define OSCL_HAS_SYMBIAN_TIMERS 0
- #define OSCL_HAS_SYMBIAN_MATH 0
- #define OSCL RAND_MAX RAND_MAX
- #define SLEEP_ONE_SEC sleep(1)

8.156.1 Define Documentation

- 8.156.1.1 #define OSCL_CLOCK_HAS_DRIFT_CORRECTION 0
- 8.156.1.2 #define OSCL_HAS_SYMBIAN_MATH 0
- 8.156.1.3 #define OSCL_HAS_SYMBIAN_TIMERS 0
- 8.156.1.4 #define OSCL RAND_MAX RAND_MAX
- 8.156.1.5 #define SLEEP_ONE_SEC sleep(1)

8.157 osclconfig_util_check.h File Reference

8.158 pvlogger.h File Reference

This file contains basic logger interfaces for common use across platforms.

```
#include "oscl_base.h"
#include "oscl_vector.h"
#include "oscl_shared_ptr.h"
#include "oscl_base_alloc.h"
```

Data Structures

- class [PVLogger](#)

Defines

- #define [PVLOGMSG_INST_REL](#) 0
- #define [PVLOGMSG_INST_PROF](#) 1
- #define [PVLOGMSG_INST_HLDBG](#) 2
- #define [PVLOGMSG_INST_MLDBG](#) 3
- #define [PVLOGMSG_INST_LLDBG](#) 4
- #define [PVLOGGER_INST_LEVEL](#) 5
- #define [_PVLOGGER_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [_PVLOGGER_LOGMSG_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [_PVLOGGER_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [_PVLOGGER_LOGBIN_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_INST_LEVEL_SUPPORT](#) 1
- #define [PVLOGGER_LOGMSG_PVLOGMSG_INST_REL\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGMSG_V_PVLOGMSG_INST_REL\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGMSG_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGBIN_PVLOGMSG_INST_REL\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGBIN_V_PVLOGMSG_INST_REL\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGBIN_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGMSG_PVLOGMSG_INST_PROF\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGMSG_V_PVLOGMSG_INST_PROF\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGMSG_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGBIN_PVLOGMSG_INST_PROF\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGBIN_V_PVLOGMSG_INST_PROF\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGBIN_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGMSG_PVLOGMSG_INST_HLDBG\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGMSG\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGMSG_V_PVLOGMSG_INST_HLDBG\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGMSG_V\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGBIN_PVLOGMSG_INST_HLDBG\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGBIN\(LOGGER, LEVEL, MESSAGE\)](#)
- #define [PVLOGGER_LOGBIN_V_PVLOGMSG_INST_HLDBG\(LOGGER, LEVEL, MESSAGE\) _- PVLOGGER_LOGBIN_V\(LOGGER, LEVEL, MESSAGE\)](#)

- #define **PVLOGGER_LOGMSG_PVLOGMSG_INST_MLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGMSG_V_PVLOGMSG_INST_MLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN_PVLOGMSG_INST_MLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN_V_PVLOGMSG_V_INST_MLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGMSG_PVLOGMSG_INST_LLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGMSG_V_PVLOGMSG_INST_LLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN_PVLOGMSG_INST_LLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN_V_PVLOGMSG_INST_LLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGMSG**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGMSG_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGMSG_V**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGMSG_V_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGBIN_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN_V**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGBIN_V_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOG_USE_ONLY**(x) x
- #define **PVLOGGER_ENABLE** 1

Variables

- const int32 **PVLOGGER_LEVEL_UNINITIALIZED** = -1
- const **PVLogger::log_level_type** **PVLOGMSG_EMERG** = 0
- const **PVLogger::log_level_type** **PVLOGMSG_ALERT** = 1
- const **PVLogger::log_level_type** **PVLOGMSG_CRIT** = 2
- const **PVLogger::log_level_type** **PVLOGMSG_ERR** = 3
- const **PVLogger::log_level_type** **PVLOGMSG_WARNING** = 4
- const **PVLogger::log_level_type** **PVLOGMSG_NOTICE** = 5
- const **PVLogger::log_level_type** **PVLOGMSG_INFO** = 6
- const **PVLogger::log_level_type** **PVLOGMSG_STACK_TRACE** = 7
- const **PVLogger::log_level_type** **PVLOGMSG_DEBUG** = 8
- const **PVLogger::log_level_type** **PVLOGMSG_FATAL_ERROR** = **PVLOGMSG_EMERG**
- const **PVLogger::log_level_type** **PVLOGMSG_NONFATAL_ERROR** = **PVLOGMSG_ERR**
- const **PVLogger::log_level_type** **PVLOGMSG_STATISTIC** = **PVLOGMSG_INFO**
- const **PVLogger::log_level_type** **PVLOGMSG_VERBOSE** = **PVLOGMSG_DEBUG**

8.158.1 Detailed Description

This file contains basic logger interfaces for common use across platforms.

This is the main entry point header file for the logger library. It should be the only one users directly include.

8.158.2 Define Documentation

8.158.2.1 #define _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgBuffers MESSAGE; \
    }\
  }\
}
```

8.158.2.2 #define _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgBuffersV MESSAGE; \
    }\
  }\
}
```

8.158.2.3 #define _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgString MESSAGE; \
    }\
  }\
}
```

8.158.2.4 #define _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgStringV MESSAGE; \
    }\
  }\
}
```

8.158.2.5 #define PVLOGGER_ENABLE 1

In case logging is compiled out, there is no need to compile the logger runtime code either.

8.158.2.6 #define PVLOGGER_INST_LEVEL 5**8.158.2.7 #define PVLOGGER_INST_LEVEL_SUPPORT 1****8.158.2.8 #define PVLOGGER_LOG_USE_ONLY(x) x**

Used to compile in/out lines of code that are used only for **PVLogger** macros.

This code will be removed at compile time when **PVLogger** is disabled, i.e. Release mode. So do not put in any code that is necessary for correct functionality of the module

**8.158.2.9 #define PVLOGGER_LOGBIN(IL, LOGGER, LEVEL, MESSAGE)
PVLOGGER_LOGBIN_ ## IL (LOGGER, LEVEL, MESSAGE)**

This is a binary API to log messages

Parameters:

IL Instrumentation level.

LOGGER Pointer to the logger object, that acts as the logging control/interface point

LEVEL Log level of the message

MESSAGE Log Message which includes the message id, and message buffers that need to be logged.

Example Usage: `PVLOGGER_LOGBIN (PVLOGMSG_INST_LLDBG, logger_1, PVLOGMSG_WARNING, (10, 3, msgBuf1Size, msgBuf1, msgBuf2Size, msgBuf2, msgBuf3Size, msgBuf3));`

-This message contains THREE (ptr_len, ptr) pairs. Log level of this msg is PVLOGMSG_WARNING, message id is 10.

- 8.158.2.10 #define PVLOGGER_LOGBIN_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.11 #define PVLOGGER_LOGBIN_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.12 #define PVLOGGER_LOGBIN_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.13 #define PVLOGGER_LOGBIN_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.14 #define PVLOGGER_LOGBIN_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.15 #define PVLOGGER_LOGBIN_V(IL, LOGGER, LEVEL, MESSAGE)
PVLOGGER_LOGBIN_V_## IL (LOGGER, LEVEL, MESSAGE)
- 8.158.2.16 #define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.17 #define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.18 #define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.19 #define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.20 #define PVLOGGER_LOGBIN_V_PVLOGMSG_V_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.21 #define PVLOGGER_LOGMSG(IL, LOGGER, LEVEL, MESSAGE)
PVLOGGER_LOGMSG_## IL (LOGGER, LEVEL, MESSAGE)

This is the text based API to log messages

Parameters:

IL Instrumentation level.

LOGGER Pointer to the logger object, that acts as the logging control/interface point

LEVEL Log level of the message

MESSAGE Log Message which includes the message id, and any kind of formatting information

Example Usage: PVLOGGER_LOGMSG(PVLOGMSG_INST_LLDBG, logger_1, PVLOGMSG_WARNING, (13, "Test Messsage to Node 1

")); -This message of log level PVLOGMSG_WARNING, and has a message id of 13

- 8.158.2.22 #define PVLOGGER_LOGMSG_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- 8.158.2.23 #define PVLOGGER_LOGMSG_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- 8.158.2.24 #define PVLOGGER_LOGMSG_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- 8.158.2.25 #define PVLOGGER_LOGMSG_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- 8.158.2.26 #define PVLOGGER_LOGMSG_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- 8.158.2.27 #define PVLOGGER_LOGMSG_V(IL, LOGGER, LEVEL, MESSAGE)
PVLOGGER_LOGMSG_V_## IL (LOGGER, LEVEL, MESSAGE)
- 8.158.2.28 #define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.29 #define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.30 #define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.31 #define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.32 #define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.33 #define PVLOGMSG_INST_HLDBG 2

High Level Debug Layer

This layer should contain messages that have very minimal impact on performance, but are at lower level (i.e., provide more information) than would be appropriate in a shipping product. The messages are probably used to gather information and validate proper functionality at a high level as might be appropriate for IOT, stress testing, or QA testing.

8.158.2.34 #define PVLOGMSG_INST_LLDBG 4

Low Level Debug Layer

This layer should contain messages for early functional testing. The messages are typically at a very low-level and allow testing the functionality of individual modules and components. Messages at this layer will typically have a performance impact (sometimes significant) due to the fact that they are at such a low level.

8.158.2.35 #define PVLOGMSG_INST_MLDBG 3

Mid Level Debug Layer

This layer should contain messages that are useful in the middle stages of the development cycle where major components are being integrated. The components themselves should already be well-tested so the emphasis is on interfaces between these components and integration testing. Messages at this layer may have some performance impact.

8.158.2.36 #define PVLOGMSG_INST_PROF 1

Profile Layer

The profile layer is used for messages and information related to measuring and reporting performance-related information.

8.158.2.37 #define PVLOGMSG_INST_REL 0

Release Layer

The release layer should only be used for messages that should remain in the final release. In certain cases all messaging may be disabled depending on customer requirements. However, when allowed the release layer should contain information that will be useful diagnosing problems in a released product (perhaps after entering a diagnostic mode), but with absolutely minimal performance impact when disabled at runtime.

8.158.3 Variable Documentation

8.158.3.1 const int32 PVLOGGER_LEVEL_UNINITIALIZED = -1

8.158.3.2 const PVLogger::log_level_type PVLOGMSG_ALERT = 1

action must be taken immediately

8.158.3.3 const PVLogger::log_level_type PVLOGMSG_CRIT = 2

critical conditions

8.158.3.4 const PVLogger::log_level_type PVLOGMSG_DEBUG = 8

debug-level messages

8.158.3.5 const PVLogger::log_level_type PVLOGMSG_EMERG = 0

system is unusable

8.158.3.6 const PVLogger::log_level_type PVLOGMSG_ERR = 3

error conditions

8.158.3.7 const PVLogger::log_level_type PVLOGMSG_FATAL_ERROR = PVLOGMSG_EMERG

8.158.3.8 const PVLogger::log_level_type PVLOGMSG_INFO = 6

informational

8.158.3.9 const PVLogger::log_level_type PVLOGMSG_NONFATAL_ERROR = PVLOGMSG_ERR

8.158.3.10 const PVLogger::log_level_type PVLOGMSG_NOTICE = 5

normal but significant condition

8.158.3.11 const PVLogger::log_level_type PVLOGMSG_STACK_TRACE = 7

function enter and exit

8.158.3.12 const PVLogger::log_level_type PVLOGMSG_STATISTIC = PVLOGMSG_INFO

8.158.3.13 const PVLogger::log_level_type PVLOGMSG_VERBOSE = PVLOGMSG_DEBUG

8.158.3.14 const PVLogger::log_level_type PVLOGMSG_WARNING = 4

warning conditions

8.159 pvlogger_accessories.h File Reference

```
#include "oscl_base.h"
#include "pvlogger.h"
```

Data Structures

- class [AllPassFilter](#)
- class [PVLoggerAppender](#)
- class [PVLoggerFilter](#)
- class [PVLoggerLayout](#)

Variables

- const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_ACCEPT = 1](#)
- const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_REJECT = 2](#)
- const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_NEUTRAL = 3](#)

8.159.1 Variable Documentation

8.159.1.1 const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_ACCEPT = 1](#)

8.159.1.2 const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_NEUTRAL = 3](#)

8.159.1.3 const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_REJECT = 2](#)

8.160 pvlogger_c.h File Reference

This file contains basic logger interfaces for common use across platforms. C-callable version.

```
#include "osclconfig.h"
```

Defines

- #define PVLOGGER_C_INST_LEVEL 5
- #define PVLOGMSG_C_INST_REL 0
- #define PVLOGMSG_C_INST_PROF 1
- #define PVLOGMSG_C_INST_HLDBG 2
- #define PVLOGMSG_C_INST_MLDBG 3
- #define PVLOGMSG_C_INST_LLDBG 4
- #define PVLOGMSG_C_EMERG 0
- #define PVLOGMSG_C_ALERT 1
- #define PVLOGMSG_C_CRIT 2
- #define PVLOGMSG_C_ERR 3
- #define PVLOGMSG_C_WARNING 4
- #define PVLOGMSG_C_NOTICE 5
- #define PVLOGMSG_C_INFO 6
- #define PVLOGMSG_C_STACK_TRACE 7
- #define PVLOGMSG_C_STACK_DEBUG 8

Functions

- OSCL_IMPORT_REF void * [pvLogger_GetLoggerObject](#) (const char *tag)
- OSCL_IMPORT_REF int [pvLogger_IsActive](#) (void *logger, int log_level)
- OSCL_IMPORT_REF void [pvLogger_LogMsgString](#) (void *logger, int msgID, const char *fmt,...)

8.160.1 Detailed Description

This file contains basic logger interfaces for common use across platforms. C-callable version.

This is the main entry point header file for the logger library. It should be the only one users directly include.

8.160.2 Define Documentation

- 8.160.2.1 `#define PVLOGGER_C_INST_LEVEL 5`
- 8.160.2.2 `#define PVLOGMSG_C_ALERT 1`
- 8.160.2.3 `#define PVLOGMSG_C_CRIT 2`
- 8.160.2.4 `#define PVLOGMSG_C_EMERG 0`
- 8.160.2.5 `#define PVLOGMSG_C_ERR 3`
- 8.160.2.6 `#define PVLOGMSG_C_INFO 6`
- 8.160.2.7 `#define PVLOGMSG_C_INST_HLDBG 2`
- 8.160.2.8 `#define PVLOGMSG_C_INST_LLDBG 4`
- 8.160.2.9 `#define PVLOGMSG_C_INST_MLDBG 3`
- 8.160.2.10 `#define PVLOGMSG_C_INST_PROF 1`
- 8.160.2.11 `#define PVLOGMSG_C_INST_REL 0`
- 8.160.2.12 `#define PVLOGMSG_C_NOTICE 5`
- 8.160.2.13 `#define PVLOGMSG_C_STACK_DEBUG 8`
- 8.160.2.14 `#define PVLOGMSG_C_STACK_TRACE 7`
- 8.160.2.15 `#define PVLOGMSG_C_WARNING 4`

8.160.3 Function Documentation

- 8.160.3.1 `OSCL_IMPORT_REF void* pvLogger_GetLoggerObject (const char * tag)`
- 8.160.3.2 `OSCL_IMPORT_REF int pvLogger_IsActive (void * logger, int log_level)`
- 8.160.3.3 `OSCL_IMPORT_REF void pvLogger_LogMsgString (void * logger, int msgID, const char * fmt, ...)`

8.161 pvlogger_registry.h File Reference

```
#include "pvlogger.h"
#include "oscl_tagtree.h"
```

Data Structures

- class [PVLoggerRegistry](#)

Chapter 9

oscl Page Documentation

9.1 Todo List

Global **MAX_NUMBER_OF_BYTE_PER_UTF8** Handle 4-byte surrogate pair representation

Index

~AllPassFilter
 AllPassFilter, 113
~BufFragGroup
 BufFragGroup, 119
~BufferMgr
 BufferMgr, 116
~CallbackTimer
 CallbackTimer, 122
~CallbackTimerObserver
 CallbackTimerObserver, 124
~DNSRequestParam
 DNSRequestParam, 131
~GetHostNameParam
 GetHostNameParam, 133
~HeapBase
 HeapBase, 135
~MM_AllocInfo
 MM_AllocInfo, 147
~MM_AllocNode
 MM_AllocNode, 148
~MM_Audit_Imp
 MM_Audit_Imp, 151
~MediaData
 MediaData, 140
~MemAllocator
 MemAllocator, 143
~OSCLMemAutoPtr
 OSCLMemAutoPtr, 425
~OSCL_FastString
 OSCL_FastString, 173
~OSCL_HeapString
 osclutil, 82
~OSCL_HeapStringA
 OSCL_HeapStringA, 197
~OSCL_StackString
 osclutil, 82
~OSCL_String
 OSCL_String, 255
~OSCL_wFastString
 OSCL_wFastString, 290
~OSCL_wHeapString
 osclutil, 82
~OSCL_wHeapStringA
 OSCL_wHeapStringA, 295
~OSCL_wStackString
 osclutil, 82
~OSCL_wString
 OSCL_wString, 300
~OsclAcceptMethod
 OsclAcceptMethod, 303
~OsclActiveObject
 OsclActiveObject, 306
~OsclAllocDestructDealloc
 OsclAllocDestructDealloc, 309
~OsclAsyncFile
 OsclAsyncFile, 312
~OsclAsyncFileBuffer
 OsclAsyncFileBuffer, 315
~OsclBinIStream
 OsclBinIStream, 319
~OsclBinOStream
 OsclBinOStream, 326
~OsclBindMethod
 OsclBindMethod, 317
~OsclComponentRegistry
 OsclComponentRegistry, 339
~OsclComponentRegistryElement
 OsclComponentRegistryElement, 341
~OsclConnectMethod
 OsclConnectMethod, 343
~OsclDNS
 OsclDNS, 346
~OsclDNSI
 OsclDNSI, 348
~OsclDNSIBase
 OsclDNSIBase, 351
~OsclDNSObserver
 OsclDNSObserver, 356
~OsclDNSRequest
 OsclDNSRequest, 357
~OsclExclusiveArrayPtr
 OsclExclusiveArrayPtr, 376
~OsclExclusivePtr
 OsclExclusivePtr, 379
~OsclExclusivePtrA
 OsclExclusivePtrA, 382
~OsclExecSchedulerCommonBase
 OsclExecSchedulerCommonBase, 390
~OsclFileCache
 OsclFileCache, 397

~OsclGetHostByNameMethod
 OsclGetHostByNameMethod, [403](#)
 ~OsclIPSocketI
 OsclIPSocketI, [408](#)
 ~OsclJump
 OsclJump, [410](#)
 ~OsclListenMethod
 OsclListenMethod, [411](#)
 ~OsclLockBase
 OsclLockBase, [413](#)
 ~OsclMemAudit
 OsclMemAudit, [418](#)
 ~OsclMemPoolAllocator
 OsclMemPoolAllocator, [432](#)
 ~OsclMemPoolFixedChunkAllocator
 OsclMemPoolFixedChunkAllocator, [434](#)
 ~OsclMemPoolFixedChunkAllocatorObserver
 OsclMemPoolFixedChunkAllocator-
 Observer, [437](#)
 ~OsclMemPoolResizableAllocator
 OsclMemPoolResizableAllocator, [439](#)
 ~OsclMemPoolResizableAllocatorMemoryObserver
 OsclMemPoolResizableAllocatorMemory-
 Observer, [446](#)
 ~OsclMemPoolResizableAllocatorObserver
 OsclMemPoolResizableAllocatorObserver,
 [447](#)
 ~OsclMemStatsNode
 OsclMemStatsNode, [448](#)
 ~OsclMutex
 OsclMutex, [449](#)
 ~OsclNativeFile
 OsclNativeFile, [453](#)
 ~OsclNullLock
 OsclNullLock, [457](#)
 ~OsclPriorityQueue
 OsclPriorityQueue, [461](#)
 ~OsclPriorityQueueBase
 OsclPriorityQueueBase, [464](#)
 ~OsclRecvFromMethod
 OsclRecvFromMethod, [476](#)
 ~OsclRecvMethod
 OsclRecvMethod, [480](#)
 ~OsclRefCounter
 OsclRefCounter, [482](#)
 ~OsclRefCounterDA
 OsclRefCounterDA, [484](#)
 ~OsclRefCounterMTDA
 OsclRefCounterMTDA, [488](#)
 ~OsclRefCounterMTSA
 OsclRefCounterMTSA, [490](#)
 ~OsclRefCounterMemFrag
 OsclRefCounterMemFrag, [486](#)
 ~OsclRefCounterSA

OsclRefCounterSA, [492](#)
 ~OsclRegistryAccessClient
 OsclRegistryAccessClient, [494](#)
 ~OsclRegistryClient
 OsclRegistryClient, [499](#)
 ~OsclRegistryServTlsImpl
 OsclRegistryServTlsImpl, [505](#)
 ~OsclSchedulerObserver
 OsclSchedulerObserver, [507](#)
 ~OsclScopedLock
 OsclScopedLock, [508](#)
 ~OsclSemaphore
 OsclSemaphore, [511](#)
 ~OsclSendMethod
 OsclSendMethod, [513](#)
 ~OsclSendToMethod
 OsclSendToMethod, [515](#)
 ~OsclSharedPtr
 OsclSharedPtr, [518](#)
 ~OsclShutdownMethod
 OsclShutdownMethod, [520](#)
 ~OsclSingleton
 OsclSingleton, [522](#)
 ~OsclSocketI
 OsclSocketI, [526](#)
 ~OsclSocketIBase
 OsclSocketIBase, [531](#)
 ~OsclSocketMethod
 OsclSocketMethod, [536](#)
 ~OsclSocketObserver
 OsclSocketObserver, [538](#)
 ~OsclSocketRequestAO
 OsclSocketRequestAO, [541](#)
 ~OsclSocketServ
 OsclSocketServ, [544](#)
 ~OsclSocketServIBase
 OsclSocketServIBase, [549](#)
 ~OsclTCPSocket
 OsclTCPSocket, [554](#)
 ~OsclTCPSocketI
 OsclTCPSocketI, [560](#)
 ~OsclTLS
 OsclTLS, [579](#)
 ~OsclTLSEx
 OsclTLSEx, [581](#)
 ~OsclThread
 OsclThread, [562](#)
 ~OsclThreadLock
 OsclThreadLock, [566](#)
 ~OsclTimer
 OsclTimer, [570](#)
 ~OsclTimerObject
 OsclTimerObject, [574](#)
 ~OsclTimerObserver

OsclTimerObserver, 577
 ~OsclUDPSocket
 OsclUDPSocket, 588
 ~OsclUDPSocketI
 OsclUDPSocketI, 594
 ~Oscl_File
 Oscl_File, 178
 ~Oscl_FileFind
 Oscl_FileFind, 187
 ~Oscl_FileServer
 Oscl_FileServer, 190
 ~Oscl_Linked_List
 Oscl_Linked_List, 203
 ~Oscl_Linked_List_Base
 Oscl_Linked_List_Base, 208
 ~Oscl_MTLinked_List
 Oscl_MTLinked_List, 220
 ~Oscl_Queue
 Oscl_Queue, 231
 ~Oscl_Queue_Base
 Oscl_Queue_Base, 233
 ~Oscl_Rb_Tree
 Oscl_Rb_Tree, 238
 ~Oscl_TAlloc
 Oscl_TAlloc, 277
 ~Oscl_Tag
 Oscl_Tag, 260
 ~Oscl_TagTree
 Oscl_TagTree, 265
 ~Oscl_Vector
 Oscl_Vector, 281
 ~Oscl_Vector_Base
 Oscl_Vector_Base, 286
 ~PVActiveBase
 PVActiveBase, 598
 ~PVLogger
 PVLogger, 603
 ~PVLoggerAppender
 PVLoggerAppender, 608
 ~PVLoggerFilter
 PVLoggerFilter, 610
 ~PVLoggerLayout
 PVLoggerLayout, 611
 ~PVLoggerRegistry
 PVLoggerRegistry, 613
 ~PVSchedulerStopper
 PVSchedulerStopper, 616
 ~PVThreadContext
 PVThreadContext, 619
 ~SendToParam
 SendToParam, 625
 ~OsclBasicAllocator
 _OsclBasicAllocator, 107
 ~_OsclHeapBase

 _OsclHeapBase, 109
 _FILE_OFFSET_BITS
 osclconfig_io.h, 797
 _OSCL_Abort
 osclbase, 34
 _OSCL_CLEANUP_BASE_CLASS
 osclmemory, 48
 _OSCL_TRAP_NEW
 osclmemory, 48
 _OsclBasicAllocator, 106
 _OsclBasicAllocator
 ~_OsclBasicAllocator, 107
 allocate, 107
 deallocate, 107
 _OsclHeapBase, 108
 _OsclHeapBase, 109
 _OsclHeapBase
 ~_OsclHeapBase, 109
 _OsclHeapBase, 109
 PVCleanupStack, 109
 _OsclInteger64Transport
 oscl_int64_utils.h, 688
 _Ownership
 OSCLMemAutoPtr, 427
 _PVLOGGER_LOGBIN
 pvlogger.h, 831
 _PVLOGGER_LOGBIN_V
 pvlogger.h, 831
 _PVLOGGER_LOGMSG
 pvlogger.h, 831
 _PVLOGGER_LOGMSG_V
 pvlogger.h, 831
 _PV_TRAP
 oscl_error_imp_fatalerror.h, 669
 oscl_error_imp_jumps.h, 670
 osclerror, 86
 _PV_TRAP_NO_TLS
 oscl_error_imp_fatalerror.h, 669
 oscl_error_imp_jumps.h, 670
 osclerror, 86
 _Ptr
 OsclExclusiveArrayPtr, 377
 OsclExclusivePtr, 380
 OsclExclusivePtrA, 383
 OsclSingleton, 523
 OsclTLS, 580
 OsclTLSEx, 582
 _STRLIT
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
 _STRLIT_CHAR
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
 _STRLIT_WCHAR

```

osclconfig_unix_android.h, 822
osclconfig_unix_common.h, 826
__TFS__
  osclconfig.h, 787
__Validate_BasicTimeDateStruct__
  osclconfig_time_check.h, 818
__Validate_BasicTimeStruct__
  osclconfig_time_check.h, 818
__int16_check__
  osclconfig, 23
__int32_check__
  osclconfig, 23
__int8_check__
  osclconfig, 23
__uint16_check__
  osclconfig, 23
__uint32_check__
  osclconfig, 23
__uint8_check__
  osclconfig, 23
__verify_TOsclConditionObject_defined__
  osclconfig_proc_check.h, 811
__verify_TOsclFileOffset_defined__
  osclconfig_io_check.h, 802
__verify_TOsclMutexObject_defined__
  osclconfig_proc_check.h, 811
__verify_TOsclSemaphoreObject_defined__
  osclconfig_proc_check.h, 811
__verify_TOsclThreadFuncArg_defined__
  osclconfig_proc_check.h, 811
__verify_TOsclThreadFuncRet_defined__
  osclconfig_proc_check.h, 811
__verify_TOsclThreadId_defined__
  osclconfig_proc_check.h, 811
__verify_TOsclThreadObject_defined__
  osclconfig_proc_check.h, 811
_fixedCaches
  OsclFileCache, 397
_movableCache
  OsclFileCache, 397
_oscl_audit_calloc
  osclmemory, 57
_oscl_audit_free
  osclmemory, 57
_oscl_audit_malloc
  osclmemory, 57
_oscl_audit_new
  osclmemory, 57
_oscl_audit_realloc
  osclmemory, 58
_oscl_malloc
  osclmemory, 58
_oscl_default_audit_malloc
  osclmemory, 58
__oscl_default_audit_malloc
  osclmemory, 58
__oscl_default_audit_new
  osclmemory, 58
__oscl_default_audit_realloc
  osclmemory, 58
__oscl_free
  osclmemory, 58
__oscl_malloc
  osclmemory, 58
__oscl_realloc
  osclmemory, 58
a
  internalLeave, 136
Abort
  OsclDNSMethod, 354
  OsclDNSRequestAO, 359
  OsclSocketMethod, 536
  OsclSocketRequestAO, 541
AbortAll
  OsclDNSMethod, 354
  OsclSocketMethod, 536
Accept
  OsclAcceptMethod, 303
  OsclAcceptRequest, 304
  OsclSocketI, 526
  OsclSocketIBase, 531
  OsclTCPSocket, 554
  OsclTCPSocketI, 560
AcceptParam, 110
  AcceptParam, 110
AcceptParam
  AcceptParam, 110
  iBlankSocket, 110
AcceptRequest
  OsclAcceptMethod, 303
Activate
  OsclDNSRequest, 357
  OsclSocketRequest, 539
  PVActiveBase, 598
Add
  OsclSocketServRequestList, 550
  OsclTimerQ, 578
add_element
  Oscl_Linked_List, 204
  Oscl_Linked_List_Base, 208
  Oscl_MTLLinked_List, 221
add_ref
  CHeapRep, 128
add_to_front
  Oscl_Linked_List, 204
  Oscl_Linked_List_Base, 208
  Oscl_MTLLinked_List, 221

```

addAllocNode
 MM_Audit_Imp, 151
 AddAppender
 PVLogger, 603
 AddFilter
 PVLogger, 603
 AddFixedCache
 Oscl_File, 178
 OsclFileCache, 397
 AddFragment
 BufFragGroup, 119
 AddLocalFragment
 MediaData, 140
 addnewmempoolbuffer
 OsclMemPoolResizableAllocator, 439
 addRef
 Oscl_DefAllocWithRefCounter, 170
 OsclMemPoolFixedChunkAllocator, 434
 OsclMemPoolResizableAllocator, 439
 OsclRefCounter, 482
 OsclRefCounterDA, 485
 OsclRefCounterMTDA, 489
 OsclRefCounterMTSA, 491
 OsclRefCounterSA, 493
 address
 Oscl_TAlloc, 277
 AddToExecTimerQ
 OsclExecSchedulerCommonBase, 390
 AddToScheduler
 OsclActiveObject, 306
 OsclTimerObject, 574
 PVActiveBase, 598
 After
 OsclTimerObject, 574
 Alloc
 OsclIPSocketI, 408
 OsclSocketMethod, 536
 OsclSocketRequestAO, 541
 ALLOC_AND_CONSTRUCT
 osclbase, 31
 alloc_and_construct
 Oscl_TAlloc, 277
 alloc_and_construct_fl
 Oscl_TAlloc, 277
 ALLOC_NODE_FLAG
 osclmemory, 60
 alloc_type
 PVLogger, 603
 PVLoggerRegistry, 613
 ALLOCATE
 osclbase, 31
 allocate
 _OsclBasicAllocator, 107
 MemAllocator, 143
 Oscl_Alloc, 167
 Oscl_DefAlloc, 169
 Oscl_Opaque_Type_Alloc, 224
 Oscl_Opaque_Type_Alloc_LL, 225
 Oscl_TAlloc, 277
 OsclErrorAllocator, 369
 OsclMemAllocator, 415
 OsclMemAllocDestructDealloc, 416
 OSCLMemAutoPtr, 426
 OsclMemBasicAllocator, 428
 OsclMemBasicAllocDestructDealoc, 429
 OsclMemPoolFixedChunkAllocator, 434
 OsclMemPoolResizableAllocator, 440
 OsclReadyAlloc, 472
 allocate_fl
 Oscl_Alloc, 167
 Oscl_DefAlloc, 169
 Oscl_TAlloc, 277
 OsclMemAllocator, 415
 OsclMemAllocDestructDealloc, 416
 OsclReadyAlloc, 472
 allocateblock
 OsclMemPoolResizableAllocator, 440
 allocator, 111
 allocNum
 MM_AllocInfo, 147
 MM_AllocQueryInfo, 149
 AllPassFilter, 112
 AllPassFilter, 113
 AllPassFilter
 ~AllPassFilter, 113
 AllPassFilter, 113
 filter_status_type, 112
 FilterOpaqueMessge, 113
 FilterString, 113
 log_level_type, 112
 message_id_type, 112
 ALREADY_SUSPENDED_ERROR
 OsclProcStatus, 465
 Append
 OsclPtr, 467
 append
 CFastRep, 126
 CHeapRep, 128
 CStackRep, 130
 APPEND_MEDIA_AT_END
 osclutil, 82
 append_rep
 CHeapRep, 128
 OSCL_String, 255
 OSCL_wString, 300
 AppendBuffers
 PVLoggerAppender, 608
 AppendNext

BufFragGroup, 119
 AppendString
 PVLoggerAppender, 608
 assign
 CHheapRep, 128
 assign_vector
 Oscl_Vector_Base, 286
 asyncfilereadcancel_test
 Oscl_File, 183
 asyncfilereadwrite_test
 Oscl_File, 183
 Attach
 OsclBinStream, 332
 audit_type
 OsclMemGlobalAuditObject, 430
 available_localbuf
 MediaData, 141

 back
 Oscl_Queue, 231
 Oscl_Vector, 282
 BAD_THREADID_ADDR_ERROR
 OsclProcStatus, 465
 base_link_type
 Oscl_Rb_Tree_Base, 240
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
 Oscl_Rb_Tree_Node_Base, 248
 begin
 Oscl_Map, 214
 Oscl_Rb_Tree, 238
 Oscl_TagTree, 265
 Oscl_Vector, 282
 BeginScheduling
 OsclExecSchedulerCommonBase, 390
 BeginStats
 OsclExecSchedulerCommonBase, 390
 BFG_SUCCESS
 BufFragStatusClass, 121
 big_endian_to_host
 osclbase, 34
 Bind
 osclbase, 34
 OsclBindMethod, 317
 OsclBindRequest, 318
 OsclIPSocketI, 408
 OsclSocketI, 526
 OsclSocketIBase, 531
 OsclTCPSocket, 554
 OsclUDPSocket, 589
 bind
 BufferState, 117
 BindAsync
 OsclSocketIBase, 531

 OsclTCPSocket, 554
 OsclTCPSocketI, 560
 OsclUDPSocket, 589
 OsclUDPSocketI, 594
 BindParam, 114
 BindParam, 114
 BindParam
 BindParam, 114
 iAddr, 114
 BindRequest
 OsclBindMethod, 317
 black
 Oscl_Rb_Tree_Node_Base, 248
 BlockingLoopL
 OsclExecSchedulerCommonBase, 390
 bSetFailure
 MM_AllocInfo, 147
 Buffer
 OsclAsyncFileBuffer, 315
 buffer
 CFastRep, 126
 CHheapRep, 128
 CStackRep, 130
 buffer_states
 BufFragGroup, 120
 BufferFragment, 115
 BufferFreeFuncPtr
 osclutil, 67
 BufferMgr, 116
 BufferMgr
 ~BufferMgr, 116
 BufferReleased, 116
 BufferReleased
 BufferMgr, 116
 BufferState, 117
 BufferState, 117
 BufferState
 bind, 117
 BufferState, 117
 decrement_refcnt, 117
 get_buf_mgr, 117
 get_free_function, 117
 get_ptr, 117
 get_refcount, 117
 increment_refcnt, 117
 reset, 117
 BufFragGroup, 118
 BufFragGroup, 119
 BufFragGroup
 ~BufFragGroup, 119
 AddFragment, 119
 AppendNext, 119
 buffer_states, 120
 BufFragGroup, 119

Clear, 119
 fragments, 120
 GetLength, 119
 GetMaxFrags, 120
 GetNext, 120
 GetNumFrags, 120
 length, 120
 next, 120
 num_frags, 120
BufFragStatusClass, 121
 BFG_SUCCESS, 121
 EMPTY_FRAGMENT, 121
 FIXED_FRAG_LOC_FULL, 121
 INTERNAL_ERROR, 121
 INVALID_ID, 121
 NOT_ENOUGH_SPACE, 121
 NULL_INPUT, 121
 TOO_MANY_FRAGS, 121
BufFragStatusClass
 status_t, 121
bufsize
 Oscl_Queue_Base, 235
 Oscl_Vector_Base, 288
BYTES_IN_UUID_ARRAY
 oscl_uuid.h, 784

c
 OsclPriorityQueue, 463

c_bool
 osclbase, 33

c_str
 StrPtrLen, 633
 WStrPtrLen, 643

Callback
 OsclReadyQ, 475

callback_timer_type
 OsclTimer, 570

CallbackTimer, 122
 CallbackTimer, 122

CallbackTimer
 ~CallbackTimer, 122
 CallbackTimer, 122
 Run, 122

CallbackTimer< Alloc >
 OsclTimer, 571

CallbackTimerObserver, 124

CallbackTimerObserver
 ~CallbackTimerObserver, 124
 TimerBaseElapsed, 124

CallRunExec
 OsclExecSchedulerCommonBase, 390

Cancel
 OsclActiveObject, 306
 OsclTimer, 570

OsclTimerObject, 574
PVActiveBase, 598

CancelAccept
 OsclSocketIBase, 532
 OsclTCPSocket, 554
 OsclTCPSocketI, 560

CancelBind
 OsclSocketIBase, 532
 OsclTCPSocket, 555
 OsclTCPSocketI, 560
 OsclUDPSocket, 589
 OsclUDPSocketI, 594

CancelConnect
 OsclSocketIBase, 532
 OsclTCPSocket, 555
 OsclTCPSocketI, 560

CancelFreeChunkAvailableCallback
 OsclMemPoolFixedChunkAllocator, 434
 OsclMemPoolResizableAllocator, 440

CancelFreeMemoryAvailableCallback
 OsclMemPoolResizableAllocator, 440

CancelFxn
 OsclDNSIBase, 351
 OsclSocketIBase, 532

CancelGetHostByName
 OsclDNS, 346
 OsclDNSIBase, 351

CancelListen
 OsclSocketIBase, 532
 OsclTCPSocket, 555
 OsclTCPSocketI, 560

CancelMethod
 OsclDNSMethod, 354
 OsclSocketMethod, 536

CancelRecv
 OsclSocketIBase, 532
 OsclTCPSocket, 555
 OsclTCPSocketI, 560

CancelRecvFrom
 OsclSocketIBase, 532
 OsclUDPSocket, 589
 OsclUDPSocketI, 594

CancelRequest
 OsclDNSRequest, 357
 OsclSocketRequest, 539

CancelSend
 OsclSocketIBase, 532
 OsclTCPSocket, 555
 OsclTCPSocketI, 560

CancelSendTo
 OsclSocketIBase, 532
 OsclUDPSocket, 589
 OsclUDPSocketI, 594

CancelShutdown

OsclSocketIBase, 532
 OsclTCPSocket, 555
 OsclTCPSocketI, 560
 capacity
 Oscl_Queue_Base, 234
 Oscl_Vector_Base, 286
 OsclFileCacheBuffer, 399
 CFastRep, 125
 CFastRep, 126
 CFastRep
 append, 126
 buffer, 126
 CFastRep, 126
 maxsize, 126
 overwrite, 126
 set_r, 126
 set_w, 126
 size, 126
 writable, 126
 chartype
 OSCL_FastString, 173
 OSCL_HeapString, 194
 OSCL_HeapStringA, 196
 OSCL_StackString, 252
 OSCL_String, 255
 OSCL_wFastString, 289
 OSCL_wHeapString, 293
 OSCL_wHeapStringA, 295
 OSCL_wStackString, 298
 OSCL_wString, 300
 CHeapRep, 127
 CHeapRep, 128
 CHeapRep
 add_ref, 128
 append, 128
 append_rep, 128
 assign, 128
 buffer, 128
 CHeapRep, 128
 maxsize, 128
 refcount, 128
 remove_ref, 128
 set, 128
 set_rep, 128
 size, 128
 check_fence
 MM_AllocBlockFence, 144
 check_list
 Oscl_Linked_List, 204
 Oscl_Linked_List_Base, 208
 checkSum
 StrCSumPtrLen, 630
 CheckSumType
 StrCSumPtrLen, 630
 children
 Oscl_TagTree::Node, 275
 children_type
 Oscl_TagTree, 265
 Oscl_TagTree::Node, 275
 ChooseCurCache
 Oscl_File::OsclCacheObserver, 184
 CleanInUse
 OsclAsyncFileBuffer, 315
 Cleanup
 OsclErrorTrap, 371
 OsclInit, 405
 OsclMem, 414
 OsclScheduler, 506
 PVLogger, 604
 CleanupExecQ
 OsclExecSchedulerCommonBase, 390
 CleanupParam
 OsclSocketRequestAO, 541
 CleanupStatQ
 OsclExecSchedulerCommonBase, 390
 Clear
 BufFragGroup, 119
 MediaData, 140
 OsclTimer, 570
 clear
 Oscl_Map, 214
 Oscl_Queue, 231
 Oscl_Queue_Base, 234
 Oscl_Rb_Tree, 238
 Oscl_TagTree, 266
 Oscl_Vector, 282
 Close
 Oscl_File, 178
 Oscl_FileFind, 187
 Oscl_FileServer, 190
 OsclAsyncFile, 312
 OsclDNSI, 348
 OsclDNSIBase, 351
 OsclFileCache, 397
 OsclIPSocketI, 408
 OsclMutex, 449
 OsclNativeFile, 453
 OsclRegistryAccessClient, 494
 OsclRegistryClient, 499
 OsclRegistryClientImpl, 502
 OsclRegistryServTlsImpl, 505
 OsclSemaphore, 511
 OsclSocketI, 526
 OsclSocketIBase, 532
 OsclSocketServ, 544
 OsclSocketServI, 546
 OsclSocketServIBase, 549
 OsclSocketServRequestList, 550

OsclTCPSocket, 555
 OsclTCPSocketI, 560
 OsclUDPSocket, 589
 OsclUDPSocketI, 594
CloseSession
 OsclComponentRegistry, 339
color
 Oscl_Rb_Tree_Node_Base, 249
color_type
 Oscl_Rb_Tree_Node_Base, 248
comp
 Oscl_Map::value_compare, 218
 OsclPriorityQueue, 463
compare
 OsclCompareLess, 337
 OsclReadyCompare, 473
 OsclTimerCompare, 572
compare_data
 Oscl_Opaque_Type_Alloc_LL, 225
compare_EQ
 Oscl_Opaque_Type_Compare, 227
 OsclPriorityQueue, 461
compare_LT
 Oscl_Opaque_Type_Compare, 227
 OsclPriorityQueue, 461
CompareId
 OsclThread, 562
Complete
 OsclDNSRequest, 357
 OsclSocketRequest, 539
COMPUTE_MEM_ALIGN_SIZE
 osclmemory, 49
Connect
 Oscl_FileServer, 190
 OsclConnectMethod, 343
 OsclConnectRequest, 344
 OsclRegistryAccessClient, 494
 OsclRegistryClient, 499
 OsclRegistryClientImpl, 502
 OsclRegistryServTlsImpl, 505
 OsclSocketI, 526
 OsclSocketIBase, 532
 OsclSocketServ, 544
 OsclSocketServI, 546
 OsclSocketServIBase, 549
 OsclTCPSocket, 556
 OsclTCPSocketI, 560
ConnectParam, 129
 ConnectParam, 129
ConnectParam
 ConnectParam, 129
 iAddr, 129
ConnectRequest
 OsclConnectMethod, 343
 const_iterator
 Oscl_Map, 213
 Oscl_Rb_Tree, 238
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_TagTree::const_iterator, 269
 Oscl_Vector, 281
const_pointer
 Oscl_Rb_Tree, 238
 Oscl_TAlloc, 277
const_reference
 Oscl_Map, 213
 Oscl_Queue, 231
 Oscl_Rb_Tree, 238
 Oscl_TAlloc, 277
 Oscl_Vector, 281
 OsclPriorityQueue, 461
Construct
 OsclReadyQ, 475
 OsclTimerQ, 578
construct
 Oscl_Linked_List_Base, 208
 Oscl_Opaque_Type_Alloc, 224
 Oscl_Opaque_Type_Alloc_LL, 225
 Oscl_Queue_Base, 234
 Oscl_TAlloc, 277
 Oscl_Vector_Base, 286
 OsclPriorityQueueBase, 464
ConstructL
 OsclDNSMethod, 354
 OsclDNSRequestAO, 359
 OsclExecSchedulerCommonBase, 390
 OsclIPSocketI, 408
 OsclSocketMethod, 536
 OsclSocketRequestAO, 541
ConstructStatQ
 OsclExecSchedulerCommonBase, 390
container_type
 OsclPriorityQueue, 461
Contains
 Oscl_File::OsclFixedCacheParam, 185
 OsclFileCacheBuffer, 399
count
 Oscl_Map, 214
 Oscl_Rb_Tree, 238
 Oscl_TagTree, 266
CPVInterfaceProxy
 OsclErrorTrapImp, 373
Create
 GetHostByNameParam, 133
 OsclMutex, 449
 OsclSemaphore, 511
 OsclThread, 563
CreateMemPool
 OsclMemPoolAllocator, 432

createmempool
 OsclMemPoolFixedChunkAllocator, 434

CreatePVLogger
 PVLoggerRegistry, 614

createStatsNode
 MM_Audit_Imp, 151

CStackRep, 130
 CStackRep, 130

CStackRep
 append, 130
 buffer, 130
 CStackRep, 130
 maxsize, 130
 set, 130
 size, 130

CTIME_BUFFER_SIZE
 osclbase, 44

CtimeStrBuf
 osclbase, 33

Current
 OsclExecScheduler, 384

currentPos
 OsclFileCacheBuffer, 399

data
 LinkedListElement, 137

data1
 OsclUuid, 596

data2
 OsclUuid, 596

data3
 OsclUuid, 596

data4
 OsclUuid, 596

deallocate
 _OsclBasicAllocator, 107
 MemAllocator, 143
 Oscl_Dealloc, 168
 Oscl_DefAlloc, 169
 Oscl_Opaque_Type_Alloc, 224
 Oscl_Opaque_Type_Alloc_LL, 225
 Oscl_TAlloc, 277
 OsclErrorAllocator, 369
 OsclMemAllocator, 415
 OsclMemAllocDestructDealloc, 416
 OSCLMemAutoPtr, 426
 OsclMemBasicAllocator, 428
 OsclMemBasicAllocDestructDealloc, 429
 OsclMemPoolFixedChunkAllocator, 435
 OsclMemPoolResizableAllocator, 440
 OsclReadyAlloc, 472

deallocateblock
 OsclMemPoolResizableAllocator, 440

decrement_refcnt

BufferState, 117

DEFAULT_MM_AUDIT_MODE
 osclmemory, 50

DEFAULT_POSTFILL_PATTERN
 osclmemory, 50

DEFAULT_PREFILL_PATTERN
 osclmemory, 50

Delete
 Oscl_DefAllocWithRefCounter, 170
 OsclAsyncFile, 312
 OsclBuf, 336

Depth
 OsclReadyQ, 475

depth
 Oscl_TagTree::Node, 275

dequeue_element
 Oscl_Linked_List, 204
 Oscl_MTLINKED_List, 221

Des
 OsclBuf, 336

DesC
 OsclBuf, 336

Destroy
 DNSRequestParam, 131
 GetHostByNameParam, 133
 PVActiveBase, 598

destroy
 Oscl_Linked_List_Base, 208
 Oscl_Opaque_Type_Alloc, 224
 Oscl_Opaque_Type_Alloc_LL, 225
 Oscl_Queue_Base, 234
 Oscl_TAlloc, 277
 Oscl_Vector, 282
 Oscl_Vector_Base, 286

destroyallmempoolbuffers
 OsclMemPoolResizableAllocator, 440

DestroyMemPool
 OsclMemPoolAllocator, 432

destroymempool
 OsclMemPoolFixedChunkAllocator, 435

destruct_and_dealloc
 Oscl_TAlloc, 277
 OsclDestructDealloc, 345
 OsclMemAllocDestructDealloc, 416
 OsclMemBasicAllocDestructDealloc, 429

difference_type
 Oscl_Rb_Tree, 238

DIR_TYPE
 Oscl_FileFind, 186

DisableAppenderInheritance
 PVLogger, 604

DiscardAcceptedSocket
 OsclAcceptMethod, 303

DNSRequestParam, 131

DNSRequestParam, 131
 OsclDNSI, 349
 OsclDNSRequestAO, 360
DNSRequestParam
 ~DNSRequestParam, 131
 Destroy, 131
 DNSRequestParam, 131
 iDNSRequest, 132
 iFxn, 132
 InThread, 131
 iRefCount, 132
 RemoveRef, 132
DoCancel
 OsclActiveObject, 307
 OsclDNSRequestAO, 359
 OsclSocketRequestAO, 541
 OsclTimerObject, 574
 PVActiveBase, 598
E_BUFFER_TOO_SMALL
 Oscl_FileFind, 187
E_INVALID_ARG
 Oscl_FileFind, 186
E_INVALID_STATE
 Oscl_FileFind, 186
E_MEMORY_ERROR
 Oscl_FileFind, 187
E_NO_MATCH
 Oscl_FileFind, 187
E_NOT_IMPLEMENTED
 Oscl_FileFind, 187
E_OK
 Oscl_FileFind, 186
E_OTHER
 Oscl_FileFind, 187
E_PATH_NOT_FOUND
 Oscl_FileFind, 186
E_PATH_TOO_LONG
 Oscl_FileFind, 186
element_type
 Oscl_FileFind, 186
elems
 Oscl_Queue_Base, 235
 Oscl_Vector_Base, 288
empty
 Oscl_Map, 214
 Oscl_Queue_Base, 234
 Oscl_Rb_Tree, 238
 Oscl_TagTree, 266
 Oscl_Vector_Base, 286
 OsclPriorityQueue, 462
EMPTY_FRAGMENT
 BufFragStatusClass, 121
EnableKill
 OsclThread, 563
 enablenullpointerreturn
 OsclMemPoolFixedChunkAllocator, 435
 OsclMemPoolResizableAllocator, 440
End
 OsclFileStats, 401
end
 Oscl_Map, 214
 Oscl_Rb_Tree, 238
 Oscl_TagTree, 266
 Oscl_Vector, 282
EndOfFile
 Oscl_File, 178
 OsclAsyncFile, 312
 OsclFileCache, 397
 OsclNativeFile, 453
endPos
 OsclFileCacheBuffer, 399
EndScheduling
 OsclExecSchedulerCommonBase, 390
EndStats
 OsclExecSchedulerCommonBase, 390
EnterThreadContext
 PVThreadContext, 619
eof
 OsclBinStream, 332
EOF_STATE
 OsclBinStream, 332
EOSCL_StringOp_CompressASCII
 osclutil, 68
EOSCL_StringOp_UTF16ToUTF8
 osclutil, 68
EOSCL_wStringOp_ExpandASCII
 osclutil, 68
EOSCL_wStringOp_UTF8ToUTF16
 osclutil, 68
EOsclFileOp_Close
 osclio, 96
EOsclFileOp_EndOfFile
 osclio, 96
EOsclFileOp_Flush
 osclio, 96
EOsclFileOp_Last
 osclio, 97
EOsclFileOp_NativeClose
 osclio, 96
EOsclFileOp_NativeEndOfFile
 osclio, 97
EOsclFileOp_NativeFlush
 osclio, 97
EOsclFileOp_NativeOpen
 osclio, 96
EOsclFileOp_NativeRead
 osclio, 96

EOscIFileOp_NativeSeek
 osclio, 97
 EOscIFileOp_NativeSize
 osclio, 97
 EOscIFileOp_NativeTell
 osclio, 97
 EOscIFileOp_NativeWrite
 osclio, 96
 EOscIFileOp_Open
 osclio, 96
 EOscIFileOp_Read
 osclio, 96
 EOscIFileOp_Seek
 osclio, 96
 EOscIFileOp_Size
 osclio, 96
 EOscIFileOp_Tell
 osclio, 96
 EOscIFileOp_Write
 osclio, 96
 eOsclProcError
 OsclProcStatus, 465
 EOscISocket_DataRecv
 oscl_socket_stats.h, 756
 EOscISocket_DataSent
 oscl_socket_stats.h, 756
 EOscISocket_Except
 oscl_socket_stats.h, 755
 EOscISocket_OS
 oscl_socket_stats.h, 755
 EOscISocket_Readable
 oscl_socket_stats.h, 755
 EOscISocket_RequestAO_Canceled
 oscl_socket_stats.h, 755
 EOscISocket_RequestAO_Error
 oscl_socket_stats.h, 755
 EOscISocket_RequestAO_Success
 oscl_socket_stats.h, 755
 EOscISocket_RequestAO_Timeout
 oscl_socket_stats.h, 755
 EOscISocket_ServPoll
 oscl_socket_stats.h, 755
 EOscISocket_ServRequestCancelIssued
 oscl_socket_stats.h, 756
 EOscISocket_ServRequestComplete
 oscl_socket_stats.h, 756
 EOscISocket_ServRequestIssued
 oscl_socket_stats.h, 755
 EOscISocket_Writable
 oscl_socket_stats.h, 755
 EOscISocketServ_LastEvent
 oscl_socket_stats.h, 755
 EOscISocketServ_LoopsockError
 oscl_socket_stats.h, 756

 EOscISocketServ_LoopsockOk
 oscl_socket_stats.h, 756
 EOscISocketServ_SelectActivity
 oscl_socket_stats.h, 755
 EOscISocketServ_SelectNoActivity
 oscl_socket_stats.h, 755
 EOscISocketServ_SelectRescheduleAsap
 oscl_socket_stats.h, 755
 EOscISocketServ_SelectReschedulePoll
 oscl_socket_stats.h, 755
 EOscISocketServ_SelectReschedulePoll
 oscl_socket_stats.h, 755
 EOscIExecStats_Last
 OsclIExecSchedulerCommonBase, 389
 EOscIExecStats_NativeOS
 OsclIExecSchedulerCommonBase, 389
 EOscIExecStats_QueueTime
 OsclIExecSchedulerCommonBase, 389
 EOscIExecStats_ReleaseTime
 OsclIExecSchedulerCommonBase, 389
 EOscIExecStats_WaitTime
 OsclIExecSchedulerCommonBase, 389
 EPriorityHigh
 OsclActiveObject, 306
 EPriorityHighest
 OsclActiveObject, 306
 EPriorityIdle
 OsclActiveObject, 306
 EPriorityLow
 OsclActiveObject, 306
 EPriorityNominal
 OsclActiveObject, 306
 EPVDNSCancel
 osclio, 97
 EPVDNSFailure
 osclio, 97
 EPVDNSGetHostByName
 osclio, 97
 EPVDNSPending
 osclio, 97
 EPVDNSSuccess
 osclio, 97
 EPVDNSTimeout
 osclio, 97
 EPVSocket_Last
 oscl_socket_types.h, 760
 EPVSocketAccept
 oscl_socket_types.h, 760
 EPVSocketBind
 oscl_socket_types.h, 760
 EPVSocketBothShutdown
 oscl_socket_types.h, 760
 EPVSocketCancel
 oscl_socket_types.h, 759
 EPVSocketConnect
 oscl_socket_types.h, 760

EPVSocketFailure
 oscl_socket_types.h, 759
 EPVSocketListen
 oscl_socket_types.h, 760
 EPVSocketPending
 oscl_socket_types.h, 759
 EPVSocketRecv
 oscl_socket_types.h, 760
 EPVSocketRecvFrom
 oscl_socket_types.h, 760
 EPVSocketRecvShutdown
 oscl_socket_types.h, 760
 EPVSocketSend
 oscl_socket_types.h, 760
 EPVSocketSendShutdown
 oscl_socket_types.h, 760
 EPVSocketSendTo
 oscl_socket_types.h, 760
 EPVSocketShutdown
 oscl_socket_types.h, 760
 EPVSocketSuccess
 oscl_socket_types.h, 759
 EPVSocketTimeout
 oscl_socket_types.h, 759
 EPVThreadContext_InThread
 osclproc, 104
 EPVThreadContext_NonOsclThread
 osclproc, 104
 EPVThreadContext_OsclThread
 osclproc, 104
 EPVThreadContext_Undetermined
 osclproc, 104
 equal_range
 Oscl_Map, 214
 Oscl_Rb_Tree, 238
 erase
 Oscl_Map, 215
 Oscl_Rb_Tree, 238
 Oscl_TagTree, 266
 Oscl_Vector, 282
 Oscl_Vector_Base, 286, 287
 Error
 OsclExecSchedulerCommonBase, 390
 error_type
 Oscl_FileFind, 186
 ESocketServ_Connected
 OsclSocketServIBase, 548
 ESocketServ_Error
 OsclSocketServIBase, 549
 ESocketServ_Idle
 OsclSocketServIBase, 548
 ESymbianAccessMode_Rfile
 Oscl_File, 177
 ESymbianAccessMode_RfileBuf

 Oscl_File, 177
 EXCEED_MAX_COUNT_VARIABLE_ERROR
 OsclProcStatus, 466
 EXCEED_MAX_SEM_COUNT_ERROR
 OsclProcStatus, 466
 Exit
 OsclThread, 563
 ExitThreadContext
 PVThreadContext, 619
 extract_string
 osclutil, 68
 fail
 OsclBinStream, 333
 FAIL_STATE
 OsclBinStream, 332
 FENCE_PATTERN
 osclmemory, 50
 FILE_TYPE
 Oscl_FileFind, 186
 fileName
 MM_AllocQueryInfo, 149
 filePosition
 OsclFileCacheBuffer, 399
 FileSize
 OsclFileCache, 397
 fill_fence
 MM_AllocBlockFence, 144
 FillFromFile
 OsclFileCacheBuffer, 399
 filter_status_type
 AllPassFilter, 112
 PVLogger, 603
 PVLoggerFilter, 609
 FilterOpaqueMessge
 AllPassFilter, 113
 PVLoggerFilter, 610
 FilterString
 AllPassFilter, 113
 PVLoggerFilter, 610
 Find
 OsclComponentRegistryData, 340
 find
 Oscl_Map, 215
 Oscl_Rb_Tree, 238
 Oscl_TagTree, 266
 find_heap
 OsclPriorityQueue, 462
 OsclPriorityQueueBase, 464
 FindExact
 OsclComponentRegistry, 339
 FindFirst
 Oscl_FileFind, 187

findfreeblock
 OsclMemPoolResizableAllocator, 441
 FindHierarchical
 OsclComponentRegistry, 339
 FindNext
 Oscl_FileFind, 188
 FindPVBBase
 OsclExecSchedulerCommonBase, 390
 first
 Oscl_Pair, 229
 firstFragPtr
 OsclBinStream, 334
 FIXED_FRAG_LOC_FULL
 BufFragStatusClass, 121
 Flush
 Oscl_File, 178
 OsclAsyncFile, 312
 OsclFileCache, 397
 OsclNativeFile, 453
 FormatOpaqueMessage
 PVLoggerLayout, 611
 FormatString
 PVLoggerLayout, 611
 fragments
 BufFragGroup, 120
 fragsLeft
 OsclBinStream, 334
 freeblockavailable
 OsclMemPoolResizableAllocatorObserver,
 447
 freebytes
 oscl_fsstat, 192
 freechunkavailable
 OsclMemPoolFixedChunkAllocator-
 Observer, 437
 freememoryavailable
 OsclMemPoolResizableAllocatorMemory-
 Observer, 446
 front
 Oscl_Queue, 232
 Oscl_Vector, 283
 Fxn
 OsclSocketRequest, 539
 get
 OsclBinIStream, 319
 OsclExclusiveArrayPtr, 376
 OsclExclusivePtr, 379
 OsclExclusivePtrA, 382
 OSCLMemAutoPtr, 426
 get_buf_mgr
 BufferState, 117
 get_count
 OsclSharedPtr, 518
 get_cstr
 OSCL_FastString, 174
 OSCL_HeapStringA, 197
 OSCL_String, 255
 OSCL_wFastString, 290
 OSCL_wHeapStringA, 295
 OSCL_wString, 300
 osclutil, 68
 get_data
 Oscl_Opaque_Type_Alloc_LL, 226
 get_element
 Oscl_Linked_List, 204
 Oscl_Linked_List_Base, 208
 Oscl_MTLINKED_List, 221
 get_first
 Oscl_Linked_List, 204
 Oscl_Linked_List_Base, 209
 get_free_function
 BufferState, 117
 get_index
 Oscl_Linked_List, 205
 Oscl_Linked_List_Base, 209
 Oscl_MTLINKED_List, 221
 get_int64_lower32
 Oscl_Int64_Utils, 201
 get_int64_middle32
 Oscl_Int64_Utils, 201
 get_int64_upper32
 Oscl_Int64_Utils, 201
 get_local_time
 TimeValue, 636
 get_lower32
 NTPTime, 165
 get_maxsize
 OSCL_FastString, 174
 OSCL_HeapStringA, 197
 OSCL_String, 255
 OSCL_wFastString, 290
 OSCL_wHeapStringA, 295
 OSCL_wString, 300
 osclutil, 69
 get_middle32
 NTPTime, 165
 get_next
 Oscl_Linked_List, 205
 Oscl_Linked_List_Base, 209
 Oscl_Opaque_Type_Alloc_LL, 226
 get_num_elements
 Oscl_Linked_List, 205
 get_ptr
 BufferState, 117
 get_pv8601_str_time
 TimeValue, 636
 get_refcount

BufferState, 117
 get_registry
 TLSStorageOps, 640
 get_rfc822_gmtime_str
 TimeValue, 636
 get_sec
 TimeValue, 637
 get_size
 OSCL_FastString, 174
 OSCL_HeapStringA, 198
 OSCL_String, 256
 OSCL_wFastString, 290
 OSCL_wHeapStringA, 295
 OSCL_wString, 300
 osclutil, 69
 get_str
 OSCL_FastString, 174
 OSCL_HeapStringA, 198
 OSCL_String, 256
 OSCL_wFastString, 290
 OSCL_wHeapStringA, 296
 OSCL_wString, 300
 osclutil, 70
 get_str_ctime
 TimeValue, 637
 get_timeval_ptr
 TimeValue, 637
 get_uint64_lower32
 Oscl_Int64_Utils, 201
 get_uint64_middle32
 Oscl_Int64_Utils, 201
 get_uint64_upper32
 Oscl_Int64_Utils, 201
 get_upper32
 NTPTime, 165
 get_usec
 TimeValue, 637
 get_value
 NTPTime, 165
 GetAcceptedSocket
 OsclAcceptMethod, 303
 GetAcceptedSocketL
 OsclTCPSocket, 556
 OsclTCPSocketI, 560
 getAllocatedSize
 OsclMemPoolResizableAllocator, 441
 getAuditRoot
 MM_Audit_Imp, 151
 GetAvailableBufferSize
 MediaData, 140
 getAvailableSize
 OsclMemPoolResizableAllocator, 441
 getBufferSize
 OsclMemPoolResizableAllocator, 441
 GetBufferState
 osclutil, 70
 getCapacity
 OsclRefCounterMemFrag, 487
 getCheckSum
 StrCSumPtrLen, 630
 getCount
 Oscl_DefAllocWithRefCounter, 170
 OsclRefCounter, 482
 OsclRefCounterDA, 485
 OsclRefCounterMemFrag, 487
 OsclRefCounterMTDA, 489
 OsclRefCounterMTSA, 491
 OsclRefCounterSA, 493
 GetElementType
 Oscl_FileFind, 188
 GetError
 Oscl_File, 179
 OsclNativeFile, 453
 GetErrorTrapImp
 OsclErrorTrap, 371
 GetFactories
 OsclRegistryAccessClient, 494
 OsclRegistryClientImpl, 502
 OsclRegistryServTlsImpl, 505
 GetFactory
 OsclRegistryAccessClient, 494
 OsclRegistryClientImpl, 502
 OsclRegistryServTlsImpl, 505
 GetFragment
 osclutil, 70
 getGlobalMemAuditObject
 OsclMemGlobalAuditObject, 430
 getHead
 OsclDoubleListBase, 364
 GetHostName
 OsclDNS, 347
 OsclDNSI, 348
 OsclDNSIbase, 351
 OsclGetHostByNameMethod, 403
 GetHostNameParam, 133
 GetHostNameParam
 ~GetHostNameParam, 133
 Create, 133
 Destroy, 133
 iAddr, 133
 iName, 133
 GetHostNameSuccess
 OsclDNSI, 348
 OsclDNSIbase, 351
 GetId
 OsclExecSchedulerCommonBase, 390
 OsclThread, 563
 getInstance

OsclSingletonRegistry, 524
 OsclTLSRegistry, 583
 OsclTLSRegistryEx, 584
 getLargestContiguousFreeBlockSize
 OsclMemPoolResizableAllocator, 441
 GetLastError
 Oscl_FileFind, 188
 getLeaveCode
 OsclException, 374
 GetLength
 BufFragGroup, 119
 GetLocalBufsize
 MediaData, 141
 GetLocalFragment
 MediaData, 141
 GetLock
 OsclMemAudit, 419
 GetLoggerObject
 PVLogger, 604
 GetLogLevel
 PVLogger, 604
 GetMaxFrags
 BufFragGroup, 120
 GetMediaFragment
 MediaData, 141
 GetMediaSize
 MediaData, 141
 getMemFrag
 OsclRefCounterMemFrag, 487
 getMemFragPtr
 OsclRefCounterMemFrag, 487
 getMemFragSize
 OsclRefCounterMemFrag, 487
 getMemPoolBufferAllocatedSize
 OsclMemPoolResizableAllocator, 441
 getMemPoolBufferSize
 OsclMemPoolResizableAllocator, 441
 GetName
 OsclExecSchedulerCommonBase, 390
 GetNext
 BufFragGroup, 120
 GetNumAppenders
 PVLogger, 604
 GetNumFrags
 BufFragGroup, 120
 GetNumMediaFrags
 MediaData, 141
 getOffset
 OsclDoubleListBase, 364
 GetParent
 PVLogger, 605
 GetPriority
 OsclThread, 564
 GetPVLoggerObject

PVLoggerRegistry, 614
 GetPVLoggerRegistry
 PVLoggerRegistry, 614
 GetReadAsyncNumElements
 OsclNativeFile, 453
 GetRecvData
 OsclIPSocketI, 408
 OsclRecvFromMethod, 476
 OsclRecvFromRequest, 478
 OsclRecvMethod, 480
 OsclRecvRequest, 481
 OsclTCPSocket, 556
 OsclTCPSocketI, 560
 OsclUDPSocket, 590
 OsclUDPSocketI, 594
 GetRefCounter
 OsclSharedPtr, 518
 getRefCounter
 OsclRefCounterMemFrag, 487
 GetRep
 OsclSharedPtr, 518
 GetScheduler
 OsclExecSchedulerCommonBase, 390
 GetSendData
 OsclIPSocketI, 408
 OsclSendMethod, 513
 OsclSendRequest, 514
 OsclSendToMethod, 515
 OsclSendToRequest, 516
 OsclTCPSocket, 556
 OsclTCPSocketI, 560
 OsclUDPSocket, 590
 OsclUDPSocketI, 594
 GetShutdown
 OsclSocketIBase, 532
 getSize
 MM_Audit_Imp, 151
 GetSocketError
 OsclDNSRequestAO, 359
 OsclSocketRequestAO, 541
 getTagActualSize
 MM_Audit_Imp, 151
 GetTimestamp
 MediaData, 141
 good
 OsclBinStream, 333
 GOOD_STATE
 OsclBinStream, 332

Handle
 Oscl_File, 179
 OsclFileHandle, 400
 HandleDNSEvent
 OsclDNSObserver, 356

HandleSocketEvent
 OsclSocketObserver, 538
 HasAsyncBind
 OsclSocketIBase, 532
 HasAsyncListen
 OsclSocketIBase, 532
 HasAsyncRead
 OsclNativeFile, 453
 hash
 OSCL_String, 256
 OSCL_wString, 300
 HasThisOffset
 OsclAsyncFileBuffer, 315
 HaveRoomInCurrentBlock
 OsclBinStream, 333
 Head
 OsclDoubleList, 362
 OsclPriorityList, 459
 head
 Oscl_Linked_List_Base, 210
 HeapBase, 134
 HeapBase, 135
 HeapBase
 ~HeapBase, 135
 HeapBase, 135
 host_to_big_endian
 osclbase, 34
 host_to_little_endian
 osclbase, 34

iActive
 OsclIDNSRequest, 357
 iAddedNum
 PVActiveBase, 600
 iAddr
 BindParam, 114
 ConnectParam, 129
 GetHostNameParam, 133
 RecvFromParam, 621
 SendToParam, 625
 iAddress
 OsclIPSocketI, 409
 iAlloc
 OsclDNSIBase, 351
 OsclDNSMethod, 355
 OsclExecSchedulerCommonBase, 394
 OsclIPSocketI, 409
 OsclSocketIBase, 534
 OsclSocketServIBase, 549
 iAllocatedSz
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 445
 iAOPriority
 TReadyQueLink, 641

iAsyncReadBufferSize
 OsclNativeFileParams, 455
 iBlankSocket
 AcceptParam, 110
 iBlockBuffer
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 444
 iBlockInfoAlignedSize
 OsclMemPoolResizableAllocator, 443
 iBlockingMode
 OsclExecSchedulerCommonBase, 394
 iBlockPostFence
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 444
 iBlockPreFence
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 444
 iBlockSize
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 444
 iBuffer
 OsclBuf, 336
 iBufferInfoAlignedSize
 OsclMemPoolResizableAllocator, 443
 iBufferPostFence
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 445
 iBufferPreFence
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 445
 iBufferSize
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 445
 iBufRecv
 RecvFromParam, 621
 RecvParam, 623
 iBufSend
 SendParam, 624
 SendToParam, 625
 iBusy
 PVActiveBase, 600
 iCancel
 OsclSocketServRequestQElem, 552
 iCBase
 OsclTrapStackItem, 587
 iCheckFreeMemoryAvailable
 OsclMemPoolResizableAllocator, 443
 iCheckNextAvailable
 OsclMemPoolResizableAllocator, 443
 iCheckNextAvailableFreeChunk
 OsclMemPoolFixedChunkAllocator, 436
 iChunkAlignment
 OsclMemPoolFixedChunkAllocator, 436
 iChunkSize

OsclMemPoolFixedChunkAllocator, 436
 iChunkSizeMemAligned
 OsclMemPoolFixedChunkAllocator, 436
 iComponentId
 OsclComponentRegistryElement, 341
 iComponentIdCounter
 OsclComponentRegistry, 339
 iContainer
 OsclFileCacheBuffer, 399
 OsclSocketMethod, 537
 OsclSocketRequestAO, 543
 Id
 OsclAsyncFileBuffer, 315
 OsclSocketRequestAO, 542
 PVThreadContext, 619
 iData
 OsclComponentRegistry, 339
 iDebugLogger
 OsclExecSchedulerCommonBase, 394
 iDefAlloc
 OsclExecSchedulerCommonBase, 394
 iDelta
 OsclExecSchedulerCommonBase, 394
 iDNSFxN
 OsclDNSMethod, 355
 iDNSI
 OsclIDNSRequestAO, 360
 iDNSMethod
 OsclIDNSRequestAO, 360
 iDNSObserver
 OsclDNSMethod, 355
 iDNSRequest
 DNSRequestParam, 132
 iDNSRequestAO
 OsclDNSMethod, 355
 OsclDNSRequest, 357
 iDNSRequestParam
 OsclIDNSRequest, 357
 iDoStop
 OsclExecSchedulerCommonBase, 394
 iDoSuspend
 OsclExecSchedulerCommonBase, 394
 iEnableNullPtrReturn
 OsclMemPoolFixedChunkAllocator, 436
 OsclMemPoolResizableAllocator, 443
 iEndAddr
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 445
 iErrAlloc
 OsclSelect, 510
 iErrorTrapImp
 OsclExecSchedulerCommonBase, 394
 iExecTimerQ
 OsclExecSchedulerCommonBase, 394
 iExpectedNumBlocksPerBuffer
 OsclMemPoolResizableAllocator, 443
 iFactory
 OsclComponentRegistryElement, 341
 OsclRegistryAccessElement, 498
 iFilePosition
 Oscl_File::OsclFixedCacheParam, 185
 iFlags
 RecvFromParam, 621
 RecvParam, 623
 SendParam, 624
 SendToParam, 625
 iFreeMemChunkList
 OsclMemPoolFixedChunkAllocator, 436
 iFreeMemContextData
 OsclMemPoolResizableAllocator, 443
 iFreeMemPoolObserver
 OsclMemPoolResizableAllocator, 443
 ifront
 Oscl_Queue_Base, 235
 iFxn
 DNSRequestParam, 132
 SocketRequestParam, 628
 iGrandTotalTicks
 OsclExecSchedulerCommonBase, 394
 iHead
 OsclDoubleListBase, 364
 OsclDoubleRunner, 365
 iHeapCheck
 OsclSelect, 510
 iHigh
 OsclInteger64Transport, 406
 iHow
 ShutdownParam, 626
 iId
 OsclComponentRegistryElement, 341
 OsclDNSMethod, 355
 OsclIPSocketI, 409
 iIsIn
 TReadyQueLink, 641
 iJumpData
 OsclErrorTrapImp, 373
 iLeave
 OsclErrorTrapImp, 373
 iLen
 PVSockBufRecv, 617
 PVSockBufSend, 618
 iLength
 OsclBuf, 336
 iLogger
 OsclDNSMethod, 355
 OsclIDNSRequestAO, 360
 OsclExecSchedulerCommonBase, 394
 OsclIPSocketI, 409

OsclSocketServIBase, 549
 iLogPerfIndentStr
 OsclExecSchedulerCommonBase, 394
 iLogPerfIndentStrLen
 OsclExecSchedulerCommonBase, 394
 iLogPerfTotal
 OsclExecSchedulerCommonBase, 394
 iLow
 OsclInteger64Transport, 406
 iMaxLen
 PVSockBufRecv, 617
 iMaxLength
 OsclBuf, 336
 iMaxNewMemPoolBufferSz
 OsclMemPoolResizableAllocator, 443
 iMemPool
 OsclMemPoolFixedChunkAllocator, 436
 iMemPoolAligned
 OsclMemPoolFixedChunkAllocator, 436
 iMemPoolAllocator
 OsclMemPoolFixedChunkAllocator, 436
 iMemPoolBufferAllocator
 OsclMemPoolResizableAllocator, 443
 iMemPoolBufferList
 OsclMemPoolResizableAllocator, 443
 iMemPoolBufferNumLimit
 OsclMemPoolResizableAllocator, 443
 iMemPoolBufferSize
 OsclMemPoolResizableAllocator, 443
 iMimeType
 OsclRegistryAccessElement, 498
 iMultiMaxLen
 RecvFromParam, 621
 iMutex
 OsclComponentRegistry, 339
 iName
 GetHostNameParam, 133
 OsclExecSchedulerCommonBase, 394
 PVActiveBase, 600
 iNativeAccessMode
 OsclNativeFileParams, 455
 iNativeBufferSize
 OsclNativeFileParams, 455
 iNativeMode
 OsclExecSchedulerCommonBase, 394
 IncLogPerf
 OsclExecSchedulerCommonBase, 391
 increment_refcnt
 BufferState, 117
 iNext
 OsclDoubleLink, 361
 OsclDoubleRunner, 365
 OsclTrapStackItem, 587
 iNextAvailableContextData

OsclMemPoolFixedChunkAllocator, 436
 OsclMemPoolResizableAllocator, 443
 iNextFreeBlock
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 444
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 445
 Init
 OsclErrorTrap, 371
 OsclInit, 405
 OsclMem, 414
 OsclScheduler, 506
 PVLogger, 605
 InitExecQ
 OsclExecSchedulerCommonBase, 391
 Insert
 OsclDoubleListBase, 364
 OsclPriorityList, 459
 insert
 Oscl_Map, 215
 Oscl_TagTree, 267
 Oscl_Vector, 283
 Oscl_Vector_Base, 287
 insert_unique
 Oscl_Rb_Tree, 238
 InsertAfter
 OsclDoubleLink, 361
 InsertBefore
 OsclDoubleLink, 361
 InsertHead
 OsclDoubleList, 362
 OsclDoubleListBase, 364
 InsertTail
 OsclDoubleList, 362
 OsclDoubleListBase, 364
 InstallScheduler
 OsclExecSchedulerCommonBase, 391
 INT64
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
 int64
 osclbase, 33
 INT64_HILO
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
 INTERNAL_ERROR
 BuffFragStatusClass, 121
 internalLeave, 136
 osclerror, 86
 internalLeave
 a, 136
 InThread
 DNSRequestParam, 131
 iNumAOAdded

OscIExecSchedulerCommonBase, 394
iNumChunk
 OsclMemPoolFixedChunkAllocator, 436
iNumOfRun
 OsclAsyncFile, 313
iNumOfRunErr
 OsclAsyncFile, 313
iNumOutstanding
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 445
iNumSessions
 OsclComponentRegistry, 339
INVALID_ACCESS_ERROR
 OsclProcStatus, 466
INVALID_ARGUMENT_ERROR
 OsclProcStatus, 466
INVALID_FUNCTION_ERROR
 OsclProcStatus, 466
INVALID_HANDLE_ERROR
 OsclProcStatus, 466
INVALID_ID
 BufFragStatusClass, 121
INVALID_OPERATION_ERROR
 OsclProcStatus, 466
INVALID_PARAM_ERROR
 OsclProcStatus, 465
INVALID_POINTER_ERROR
 OsclProcStatus, 466
INVALID_PRIORITY_ERROR
 OsclProcStatus, 465
INVALID_THREAD_ERROR
 OsclProcStatus, 465
INVALID_THREAD_ID_ERROR
 OsclProcStatus, 465
INVALID_TYPE
 Oscl_FileFind, 186
iObserver
 OsclIPSocketI, 409
 OsclMemPoolFixedChunkAllocator, 436
 OsclMemPoolResizableAllocator, 443
iOffset
 OsclDoubleListBase, 364
 OsclDoubleRunner, 365
iOpCount
 OsclFileStatsItem, 402
iOsclBase
 OsclSelect, 510
iOsclErrorTrap
 OsclSelect, 510
iOsclLogger
 OsclSelect, 510
iOsclMemory
 OsclSelect, 510
iOsclScheduler
 OsclSelect, 510
iOtherExecStats
 OsclExecSchedulerCommonBase, 394
iOutputFile
 OsclSelect, 510
iPacketLen
 RecvFromParam, 621
iPacketSource
 RecvFromParam, 621
ipAddr
 OsclNetworkAddress, 456
iParam
 OsclFileStatsItem, 402
 OsclSocketRequest, 539
 OsclSocketRequestAO, 543
iParam2
 OsclFileStatsItem, 402
iParamSize
 OsclSocketRequestAO, 543
iParentBuffer
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 444
iPrev
 OsclDoubleLink, 361
iPrevFreeBlock
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 444
iPriority
 OsclPriorityLink, 458
iPtr
 PVSockBufRecv, 617
 PVSockBufSend, 618
iPVActiveStats
 PVActiveBase, 600
iPVReadyQLink
 PVActiveBase, 600
iPVStatQ
 OsclExecSchedulerCommonBase, 394
iPVStats
 OsclExecSchedulerCommonBase, 394
iQSize
 ListenParam, 138
iReadyQ
 OsclExecSchedulerCommonBase, 394
irear
 Oscl_Queue_Base, 235
iRefCount
 DNSRequestParam, 132
 OsclMemPoolFixedChunkAllocator, 436
 OsclMemPoolResizableAllocator, 443
iRequestedAvailableFreeMemSize
 OsclMemPoolResizableAllocator, 443
iRequestedNextAvailableSize
 OsclMemPoolResizableAllocator, 443

iResumeSem
 OsclExecSchedulerCommonBase, 394

is_writable
 OSCL_String, 256
 OSCL_wString, 301

is_zero
 TimeValue, 637

IsActive
 PVLogger, 605

IsAdded
 PVActiveBase, 598

isAllocNodePtr
 MM_AllocBlockHdr, 145

IsBusy
 OsclActiveObject, 307
 OsclTimerObject, 575

iSchedulerAlloc
 OsclSelect, 510

iSchedulerName
 OsclSelect, 510

iSchedulerReserve
 OsclSelect, 510

isCIEquivalentTo
 StrCSumPtrLen, 630
 StrPtrLen, 633
 WStrPtrLen, 643

isCIPrefixOf
 StrPtrLen, 633

iSelect
 OsclSocketServRequestQElem, 552

IsEmpty
 OsclDoubleListBase, 364

iSeqNum
 TReadyQueLink, 641

iServerError
 OsclSocketServIBase, 549

iServerState
 OsclSocketServIBase, 549

isFixed
 OsclFileCacheBuffer, 399

IsHead
 OsclDoubleList, 362
 OsclPriorityList, 459

IsIn
 OsclReadyQ, 475
 OsclTimerQ, 578

IsInAnyQ
 PVActiveBase, 599

IsInstalled
 OsclExecSchedulerCommonBase, 391

IsInUse
 OsclAsyncFileBuffer, 315

iSize
 Oscl_File::OsclFixedCacheParam, 185

isLetter
 StrPtrLen, 633

IsLocalData
 MediaData, 141

iSocket
 OsclIPSocketI, 409

iSocketError
 OsclDNSRequestAO, 360
 OsclSocketRequestAO, 543

iSocketFxn
 OsclSocketMethod, 537

iSocketI
 OsclSocketRequest, 539

iSocketRequest
 OsclSocketServRequestQElem, 552

iSocketRequestAO
 OsclSocketMethod, 537
 OsclSocketRequest, 539

iSocketServ
 OsclDNSIBase, 351
 OsclIPSocketI, 409
 OsclSocketIBase, 534

isOpen
 OsclSocketIBase, 532

IsReady
 OsclDNSIBase, 351

IsSameThreadContext
 PVThreadContext, 619

IsServConnected
 OsclSocketServIBase, 549

IsServerThread
 OsclSocketServI, 547

isSetFailure
 MM_Audit_Imp, 152

IsStarted
 OsclExecSchedulerCommonBase, 391

IsTail
 OsclDoubleList, 362
 OsclPriorityList, 459

iStartAddr
 OsclMemPoolResizableAllocator::Mem-PoolBufferInfo, 445

iStartTick
 OsclFileStatsItem, 402

iStatus
 PVActiveBase, 600

iStopper
 OsclExecSchedulerCommonBase, 394

iStopperCrit
 OsclExecSchedulerCommonBase, 394

IsUpdated
 OsclFileCacheBuffer, 399

iSuspended
 OsclExecSchedulerCommonBase, 394

IsValid
 OsclAsyncFileBuffer, 315
 iTAny
 OsclTrapStackItem, 587
 iterator
 Oscl_Linked_List_Base, 210
 Oscl_Map, 213
 Oscl_Rb_Tree, 238
 Oscl_Rb_Tree_Iterator, 245
 Oscl_TagTree::iterator, 272
 Oscl_Vector, 281
 OsclPriorityQueue, 461
 iThreadContext
 OsclExecSchedulerCommonBase, 394
 PVActiveBase, 600
 iTime
 OsclExecSchedulerCommonBase, 394
 iTimeCompareThreshold
 OsclExecSchedulerCommonBase, 394
 iTimeQueuedTicks
 TReadyQueLink, 641
 iTimeToRunTicks
 TReadyQueLink, 641
 iTotalPercent
 OsclExecSchedulerCommonBase, 394
 iTotalTicks
 OsclFileStatsItem, 402
 iTotalTicksTemp
 OsclExecSchedulerCommonBase, 394
 iTrapOperation
 OsclTrapStackItem, 587
 iTrapStack
 OsclErrorTrapImp, 373
 iVec
 OsclComponentRegistryData, 340
 iXferLen
 SendParam, 624
 SendToParam, 625

 Join
 OsclIPSocketI, 408
 OsclSocketI, 526
 OsclSocketIBase, 532
 OsclUDPSocket, 590
 Jump
 OsclJump, 410

 key_comp
 Oscl_Map, 216
 key_compare
 Oscl_Map, 213
 key_type
 Oscl_Map, 213
 Oscl_Rb_Tree, 238

 largeasyncfilereadwrite_test
 Oscl_File, 183
 Leave
 OsclError, 367
 LeaveIfError
 OsclError, 367
 LeaveIfNull
 OsclError, 367
 Left
 OsclPtrC, 470
 left
 Oscl_Rb_Tree_Node_Base, 249
 len
 OsclMemoryFragment, 431
 StrPtrLen, 633
 WStrPtrLen, 643
 Length
 OsclAsyncFileBuffer, 315
 OsclBuf, 336
 OsclPtr, 467
 OsclPtrC, 470
 length
 BuffFragGroup, 120
 OsclBinStream, 334
 StrPtrLen, 633
 WStrPtrLen, 643
 lineNo
 MM_AllocInfo, 147
 MM_AllocQueryInfo, 149
 link_type
 Oscl_Rb_Tree, 238
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
 Oscl_Rb_Tree_Node, 247
 LinkedListElement, 137
 LinkedListElement, 137
 LinkedListElement
 data, 137
 LinkedListElement, 137
 next, 137
 Listen
 OsclListenMethod, 411
 OsclListenRequest, 412
 OsclSocketI, 526
 OsclSocketIBase, 532
 OsclTCPSocket, 557
 OsclTCPSocketI, 560
 ListenAsync
 OsclSocketIBase, 532
 OsclTCPSocket, 557
 OsclTCPSocketI, 561
 ListenParam, 138
 ListenParam, 138
 ListenParam

iQSize, 138
 ListenParam, 138
ListenRequest
 OsclListenMethod, 411
little_endian_to_host
 osclbase, 35
localbuf
 MediaData, 141
Lock
 OsclLockBase, 413
 OsclMutex, 450
 OsclNullLock, 457
 OsclThreadLock, 566
lockAndGetInstance
 OsclSingletonRegistry, 524
Log
 OsclFileStats, 401
log_level_type
 AllPassFilter, 112
 PVLogger, 603
 PVLoggerFilter, 609
 PVLoggerRegistry, 613
LogAll
 OsclFileStats, 401
Logger
 OsclSocketI, 526
LogMsgBuffers
 PVLogger, 605
LogMsgBuffersV
 PVLogger, 605
LogMsgString
 PVLogger, 606
LogMsgStringV
 PVLogger, 606
LoopbackSocket
 OsclSocketServI, 547
lower_bound
 Oscl_Map, 216
 Oscl_Rb_Tree, 238
MakeAddr
 OsclSocketI, 527
makeValidTag
 MM_Audit_Imp, 152
map_type
 Oscl_TagTree, 265
mapit
 Oscl_TagTree::const_iterator, 269
 Oscl_TagTree::iterator, 272
mapiter
 Oscl_TagTree::const_iterator, 269
 Oscl_TagTree::iterator, 272
Match
 OsclComponentRegistryElement, 341

MAX_NUMBER_OF_BYTE_PER_UTF8
 osclutil, 67
max_size
 Oscl_Map, 216
 Oscl_Rb_Tree, 238
MAX_THRDS_REACHED_ERROR
 OsclProcStatus, 465
maximum
 Oscl_Rb_Tree_Node_Base, 249
MaxLen
 OsclNameString, 451
maxsize
 CFastRep, 126
 CHeapRep, 128
 CStackRep, 130
mbchar
 osclbase, 33
MediaData, 139
 MediaData, 140
MediaData
 ~MediaData, 140
 AddLocalFragment, 140
 available_localbuf, 141
 Clear, 140
 GetAvailableBufferSize, 140
 GetLocalBufsize, 141
 GetLocalFragment, 141
 GetMediaFragment, 141
 GetMediaSize, 141
 GetNumMediaFrags, 141
 GetTimestamp, 141
 IsLocalData, 141
 localbuf, 141
 MediaData, 140
 num_reserved_fragments, 141
 SetTimestamp, 141
 timestamp, 141
MediaStatusClass, 142
MediaTimestamp
 osclutil, 67
MEM_ALIGN_SIZE
 osclmemory, 50
MemAllocator, 143
MemAllocator
 ~MemAllocator, 143
 allocate, 143
 deallocate, 143
 pointer, 143
memoryPoolBufferMgmtOverhead
 OsclMemPoolResizableAllocator, 441
message_id_type
 AllPassFilter, 112
 PVLogger, 603
 PVLoggerAppender, 608

PVLoggerFilter, 609
 PVLoggerLayout, 611
MethodDone
 OsclDNSMethod, 354
 OsclSocketMethod, 536
MICROSECONDS
 osclbase, 34
MILLISECONDS
 osclbase, 34
MIN_FENCE_SIZE
 osclmemory, 50
minimum
 Oscl_Rb_Tree_Node_Base, 249
MM_AddTag
 MM_Audit_Imp, 152
 OsclMemAudit, 419
MM_ALLOC_MAX_QUERY_FILENAME_-LEN
 osclmemory, 50
MM_ALLOC_MAX_QUERY_TAG_LEN
 osclmemory, 50
MM_allocate
 MM_Audit_Imp, 152
 OsclMemAudit, 419
MM_AllocBlockFence, 144
 MM_AllocBlockFence, 144
MM_AllocBlockFence
 check_fence, 144
 fill_fence, 144
 MM_AllocBlockFence, 144
 pad, 144
MM_AllocBlockHdr, 145
 MM_AllocBlockHdr, 145
MM_AllocBlockHdr
 isAllocNodePtr, 145
 MM_AllocBlockHdr, 145
 pad, 145
 pNode, 145
 pRootNode, 145
 setAllocNodeFlag, 145
 size, 145
MM_AllocInfo, 146
 MM_AllocInfo, 147
MM_AllocInfo
 ~MM_AllocInfo, 147
 allocNum, 147
 bSetFailure, 147
 lineNo, 147
 MM_AllocInfo, 147
 operator delete, 147
 operator new, 147
 pFileName, 147
 pMemBlock, 147
 pStatsNode, 147
 size, 147
MM_AllocNode, 148
 MM_AllocNode, 148
 ~MM_AllocNode, 148
MM_AllocNodeAutoPtr
 osclmemory, 57
MM_AllocQueryInfo, 149
MM_AllocQueryInfo
 allocNum, 149
 fileName, 149
 lineNo, 149
 pMemBlock, 149
 size, 149
 tag, 149
MM_AUDIT_ALLOC_NODE_ENABLE_-FLAG
 osclmemory, 50
MM_AUDIT_ALLOC_NODE_SUPPORT
 osclmemory, 50
MM_AUDIT_FAILURE_SIMULATION_-SUPPORT
 osclmemory, 50
MM_AUDIT_FENCE_SUPPORT
 osclmemory, 50
MM_AUDIT_FILL_SUPPORT
 osclmemory, 50
MM_Audit_Imp, 150
 ~MM_Audit_Imp, 151
 addAllocNode, 151
 createStatsNode, 151
 getAuditRoot, 151
 getSize, 151
 getTagActualSize, 151
 isSetFailure, 152
 makeValidTag, 152
MM_AddTag, 152
MM_allocate, 152
 MM_Audit_Imp, 151
MM_CreateAllocNodeInfo, 152
MM_deallocate, 152
MM_GetAllocNo, 152
MM_GetAllocNodeInfo, 153
MM_GetExistingTag, 153
MM_GetMode, 153
MM_GetNumAllocNodes, 153
MM_GetOverheadStats, 153
MM_GetPostfillPattern, 153

MM_GetPrefillPattern, 153
 MM_GetRootNode, 154
 MM_GetStats, 154
 MM_GetStatsInDepth, 154
 MM_GetTagName, 154
 MM_GetTreeNodes, 154
 MM_ReleaseAllocNodeInfo, 154
 MM_SetFailurePoint, 155
 MM_SetMode, 155
 MM_SetPostfillPattern, 155
 MM_SetPrefillPattern, 155
 MM_SetTagLevel, 155
 MM_UnsetFailurePoint, 155
 MM_Validate, 155
 pruneSubtree, 156
 removeALLAllocNodes, 156
 removeAllocNode, 156
 retrieveParentTag, 156
 retrieveParentTagLength, 156
 updateStatsNode, 156
 updateStatsNodeInFailure, 156
 validate, 156
 validate_all_heap, 156
MM_AUDIT_INCLUDE_ALL_HEAP_-VALIDATION
 osclmemory, 50
MM_AUDIT_POSTFILL_FLAG
 osclmemory, 50
MM_AUDIT_PREFILL_FLAG
 osclmemory, 50
MM_AUDIT_SUPPRESS_FILENAME_FLAG
 osclmemory, 50
MM_AUDIT_VALIDATE_ALL_HEAP_FLAG
 osclmemory, 50
MM_AUDIT_VALIDATE_BLOCK
 osclmemory, 50
MM_AUDIT_VALIDATE_ON_FREE_FLAG
 osclmemory, 50
 MM_AuditOverheadStats, 158
 MM_AuditOverheadStats
 per_allocation_overhead, 158
 stats_overhead, 158
 MM_CreateAllocNodeInfo
 MM_Audit_Imp, 152
 OsclMemAudit, 419
 MM_deallocate
 MM_Audit_Imp, 152
 OsclMemAudit, 419
 MM_FailInsertParam, 159
 MM_FailInsertParam, 159
 MM_FailInsertParam
 MM_FailInsertParam, 159
 nAllocNum, 159
 operator delete, 159
 operator new, 159
 reset, 159
 xsubi, 159
 MM_GetAllocNo
 MM_Audit_Imp, 152
 OsclMemAudit, 419
 MM_GetAllocNodeInfo
 MM_Audit_Imp, 153
 OsclMemAudit, 419
 MM_GetExistingTag
 MM_Audit_Imp, 153
 OsclMemAudit, 420
 MM_GetMode
 MM_Audit_Imp, 153
 OsclMemAudit, 420
 MM_GetNumAllocNodes
 MM_Audit_Imp, 153
 OsclMemAudit, 420
 MM_GetOverheadStats
 MM_Audit_Imp, 153
 OsclMemAudit, 420
 MM_GetPostfillPattern
 MM_Audit_Imp, 153
 OsclMemAudit, 420
 MM_GetPrefillPattern
 MM_Audit_Imp, 153
 OsclMemAudit, 420
 MM_GetRefCount
 OsclMemAudit, 420
 MM_GetRootNode
 MM_Audit_Imp, 154
 OsclMemAudit, 421
 MM_GetStats
 MM_Audit_Imp, 154
 OsclMemAudit, 421
 MM_GetStatsInDepth
 MM_Audit_Imp, 154
 OsclMemAudit, 421
 MM_GetTagName
 MM_Audit_Imp, 154
 OsclMemAudit, 421
 MM_GetTreeNodes
 MM_Audit_Imp, 154
 OsclMemAudit, 421
 MM_ReleaseAllocNodeInfo
 MM_Audit_Imp, 154
 OsclMemAudit, 421
 MM_SetFailurePoint
 MM_Audit_Imp, 155
 OsclMemAudit, 421
 MM_SetMode
 MM_Audit_Imp, 155
 OsclMemAudit, 422
 MM_SetPostfillPattern

MM_Audit_Imp, 155
OsclMemAudit, 422
MM_SetPrefillPattern
 MM_Audit_Imp, 155
 OsclMemAudit, 422
MM_SetTagLevel
 MM_Audit_Imp, 155
 OsclMemAudit, 422
MM_Stats_CB, 160
 MM_Stats_CB, 160
 num_child_nodes, 160
 operator delete, 160
 operator new, 160
 pStats, 160
 tag, 160
MM_Stats_t, 161
 MM_Stats_t, 162
 numAllocFails, 162
 numAllocs, 162
 numBytes, 162
 operator delete, 162
 operator new, 162
 peakNumAllocs, 162
 peakNumBytes, 162
 reset, 162
 totalNumAllocs, 162
 totalNumBytes, 162
 update, 162
MM_StatsNodeTagTreeType
 osclmemory, 57
MM_UnsetFailurePoint
 MM_Audit_Imp, 155
 OsclMemAudit, 422
MM_Validate
 MM_Audit_Imp, 155
 OsclMemAudit, 422
MMAuditCharAutoPtr
 osclmemory, 57
MMAuditUint8AutoPtr
 osclmemory, 57
Mode
 OsclNativeFile, 453
mode
 oscl_stat_buf, 253
MODE_APPEND
 Oscl_File, 177
MODE_BINARY
 Oscl_File, 177
MODE_READ
 Oscl_File, 177
MODE_READ_PLUS
 Oscl_File, 177
MODE_READWRITE
 Oscl_File, 177
MODE_TEXT
 Oscl_File, 177
mode_type
 Oscl_File, 177
move_to_end
 Oscl_Linked_List, 205
 Oscl_Linked_List_Base, 209
 Oscl_MTLLinked_List, 221
move_to_front
 Oscl_Linked_List, 205
 Oscl_Linked_List_Base, 209
 Oscl_MTLLinked_List, 222
MSEC_PER_SEC
 osclbase, 44
MSEC_TO_MICROSEC
 oscl_socket_method.h, 744
MsecToTicks
 OsclTickCount, 567
MUTEX_LOCKED_ERROR
 OsclProcStatus, 466

nAllocNum
 MM_FailInsertParam, 159
New
 Oscl_DefAllocWithRefCounter, 171
NewL
 OsclAcceptMethod, 303
 OsclAsyncFile, 312
 OsclAsyncFileBuffer, 315
 OsclBindMethod, 317
 OsclBuf, 336
 OsclConnectMethod, 343
 OsclDNS, 347
 OsclDNSI, 349
 OsclGetHostByNameMethod, 403
 OsclListenMethod, 411
 OsclRecvFromMethod, 476
 OsclRecvMethod, 480
 OsclSendMethod, 513
 OsclSendToMethod, 515
 OsclShutdownMethod, 520
 OsclSocketI, 527
 OsclSocketServ, 545
 OsclSocketServI, 547
 OsclTCPSocket, 557
 OsclTCPSocketI, 561
 OsclUDPSocket, 590
 OsclUDPSocketI, 594
NewRequest
 OsclDNSRequestAO, 359
 OsclSocketRequestAO, 542
next
 BuffFragGroup, 120
 LinkedListElement, 137

nextFragPtr
 OsclBinStream, 334
NO_PERMISSION_ERROR
 OsclProcStatus, 465
Node
 Oscl_TagTree::Node, 275
node
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
node_ptr
 Oscl_TagTree, 265
node_type
 Oscl_TagTree, 265
NOT_ENOUGH_MEMORY_ERROR
 OsclProcStatus, 465
NOT_ENOUGH_RESOURCES_ERROR
 OsclProcStatus, 465
NOT_ENOUGH_SPACE
 BufFragStatusClass, 121
NOT_IMPLEMENTED
 OsclProcStatus, 466
NOT_SUSPENDED_ERROR
 OsclProcStatus, 465
notifyfreeblockavailable
 OsclMemPoolResizableAllocator, 441
notifyfreechunkavailable
 OsclMemPoolFixedChunkAllocator, 435
notifyfreememoryavailable
 OsclMemPoolResizableAllocator, 441
NTPTime, 163
 get_lower32, 165
 get_middle32, 165
 get_upper32, 165
 get_value, 165
 NTPTime, 164, 165
operator+=, 165
operator-, 165
operator=, 165, 166
 set_from_system_time, 166
 set_to_current_time, 166
 TimeValue, 639
 to_system_time, 166
NULL
 osclbase, 31
NULL_INPUT
 BufFragStatusClass, 121
NULL_TERM_CHAR
 osclbase, 31
num_child_nodes
 MM_Stats_CB, 160
num_elements
 Oscl_Linked_List_Base, 210
num_fragments
 BufFragGroup, 120
num_reserved.fragments
 MediaData, 141
numAllocFails
 MM_Stats_t, 162
numAllocs
 MM_Stats_t, 162
numBytes
 MM_Stats_t, 162
numelems
 Oscl_Queue_Base, 235
 Oscl_Vector_Base, 288
numFrags
 OsclBinStream, 334
octet
 osclbase, 33
Offset
 OsclAsyncFileBuffer, 315
Open
 Oscl_File, 179
 OsclAsyncFile, 312, 313
 OsclDNSI, 349
 OsclDNSIBase, 351
 OsclFileCache, 397
 OsclNativeFile, 453
 OsclSocketI, 527
 OsclSocketIBase, 533
 OsclSocketServRequestList, 550
OpenSession
 OsclComponentRegistry, 339
operator *
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
 Oscl_TagTree::const_iterator, 269
 Oscl_TagTree::iterator, 272
 OsclExclusiveArrayPtr, 376
 OsclExclusivePtr, 379
 OsclExclusivePtrA, 382
 OSCLMemAutoPtr, 426
 OsclSharedPtr, 518
 OsclSingleton, 522
 OsclTLS, 579
 OsclTLSEx, 581
operator **=
 TimeValue, 638
operator delete
 MM_AllocInfo, 147
 MM_AllocNode, 148
 MM_FailInsertParam, 159
 MM_Stats_CB, 160
 MM_Stats_t, 162
 oscl_mem.h, 698
 OsclErrorAllocator, 370
 osclmemory, 58

OsclMemStatsNode, 448
 operator delete[]
 osclmemory, 58
 operator new
 MM_AllocInfo, 147
 MM_AllocNode, 148
 MM_FailInsertParam, 159
 MM_Stats_CB, 160
 MM_Stats_t, 162
 oscl_mem.h, 698
 osclconfig_global_placement_new.h, 794
 OsclErrorAllocator, 370
 osclmemory, 58
 OsclMemStatsNode, 448
 operator new[]
 osclmemory, 58
 operator T *
 OsclDoubleRunner, 365
 operator TheClass *
 OsclSharedPtr, 519
 operator!=
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
 OSCL_String, 256
 Oscl_TagTree::const_iterator, 269
 Oscl_TagTree::iterator, 272
 OSCL_wString, 301
 OsclAOStatus, 310
 OsclUuid, 596
 StrCSumPtrLen, 630
 StrPtrLen, 633
 TimeValue, 639
 WStrPtrLen, 643
 operator()
 Oscl_Less, 202
 Oscl_Map::value_compare, 218
 Oscl_Select1st, 250
 Oscl_Tag_Base, 263
 operator++
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
 Oscl_TagTree::const_iterator, 269
 Oscl_TagTree::iterator, 272
 OsclDoubleRunner, 365
 operator+=
 NTPTime, 165
 OSCL_String, 256
 OSCL_wString, 301
 TimeValue, 638
 operator-
 NTPTime, 165
 osclbase, 35
 operator-
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
 Oscl_TagTree::const_iterator, 269
 Oscl_TagTree::iterator, 272
 OsclDoubleRunner, 365
 Oscl_ExclusiveArrayPtr, 376
 Oscl_ExclusivePtr, 379
 Oscl_ExclusivePtrA, 382
 OSCLMemAutoPtr, 426
 OsclSharedPtr, 519
 OsclSingleton, 522
 OsclTLS, 579
 OsclTLSEx, 581
 operator<
 OSCL_String, 257
 Oscl_Tag, 260
 OSCL_wString, 301
 OsclAOStatus, 310
 TimeValue, 639
 operator<<
 OsclBinOStreamBigEndian, 328
 OsclBinOStreamLittleEndian, 330
 operator<=

 OSCL_String, 257
 OSCL_wString, 301
 OsclAOStatus, 310
 TimeValue, 639
 operator=

 NTPTime, 165, 166
 OSCL_FastString, 174
 OSCL_HeapStringA, 198
 Oscl_Map, 216
 Oscl_Rb_Tree, 238
 OSCL_String, 257
 Oscl_TagTree, 267
 Oscl_Vector, 283
 OSCL_wFastString, 290
 OSCL_wHeapStringA, 296
 OSCL_wString, 301
 OsclAOStatus, 310
 OsclComponentRegistryElement, 341
 OsclExclusiveArrayPtr, 376
 Oscl_ExclusivePtr, 379
 Oscl_ExclusivePtrA, 382
 OSCLMemAutoPtr, 426
 OsclRefCounterMemFrag, 487
 OsclSharedPtr, 519
 osclutil, 70–72

OsclUuid, 596
 StrCSumPtrLen, 630
 StrPtrLen, 633
 TimeValue, 638
 WStrPtrLen, 643
operator==
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
 OSCL_String, 257
 Oscl_TagTree::const_iterator, 269
 Oscl_TagTree::iterator, 272
 OSCL_wString, 301
 OsclAOStatus, 310
 osclbase, 35
 OsclNetworkAddress, 456
 OsclUuid, 596
 StrCSumPtrLen, 630
 StrPtrLen, 633
 TimeValue, 639
 WStrPtrLen, 643
operator>
 OSCL_String, 257
 OSCL_wString, 301
 OsclAOStatus, 310
 TimeValue, 639
operator>=
 OSCL_String, 257
 OSCL_wString, 301
 OsclAOStatus, 310
 TimeValue, 639
operator>>
 OsclBinIStreamBigEndian, 322
 OsclBinIStreamLittleEndian, 325
operator[]
 Oscl_Map, 216
 OSCL_String, 257
 Oscl_TagTree, 267
 Oscl_Vector, 283
 OSCL_wString, 301
otype
 OSCL_FastString, 173
 OSCL_HeapString, 194
 OSCL_HeapStringA, 196
 OSCL_StackString, 252
 OSCL_wFastString, 289
 OSCL_wHeapString, 293
 OSCL_wHeapStringA, 295
 OSCL_wStackString, 298
OSCL Base, 24
OSCL config, 20
OSCL Error, 83
OSCL Init, 105
OSCL IO, 93
OSCL Memory, 45
 OSCL Proc, 101
 OSCL Util, 61
 OSCL_ABS
 osclbase, 31
 oscl_abs
 osclutil, 72
 OSCL_AF_INET
 osclconfig_io.h, 797
 Oscl_Alloc, 167
 allocate, 167
 allocate_fl, 167
 OSCL_ALLOC_DELETE
 osclmemory, 50
 OSCL_ALLOC_NEW
 osclmemory, 51
 oscl_aostatus.h, 644
 OSCL_ARRAY_DELETE
 osclmemory, 51
 OSCL_ARRAY_NEW
 osclmemory, 51
 OSCL_ASCII_CASE_MAGIC_BIT
 osclutil, 82
 oscl_asin
 osclutil, 72
 OSCL_ASSERT
 osclbase, 31
 OSCL_Assert
 osclbase, 35
 oscl_assert.h, 645
 OSCL_ASSERT_ALWAYS
 osclconfig, 21
 oscl_atan
 osclutil, 72
 OSCL_AUDIT_ARRAY_NEW
 osclmemory, 51
 OSCL_AUDIT_CALLOC
 osclmemory, 52
 OSCL_AUDIT_MALLOC
 osclmemory, 52
 OSCL_AUDIT_NEW
 osclmemory, 52
 OSCL_AUDIT_REALLOC
 osclmemory, 53
 OSCL_BAD_ALLOC_EXCEPTION_CODE
 osclerror, 86
 oscl_base.h, 646
 oscl_base_alloc.h, 647
 oscl_base_macros.h, 648
 oscl_bin_stream.h, 649
 OSCL_BYPASS_MEMMGT
 osclconfig_memory.h, 807
 oscl_byte_order.h, 650
 OSCL_BYTE_ORDER_BIG_ENDIAN
 osclconfig, 21

OSCL_BYTE_ORDER_LITTLE_ENDIAN
 osclconfig, 21

OSCL_CALLOC
 osclmemory, 53

oscl_calloc
 osclmemory, 53

OSCL_CATCH
 osclerror, 86

OSCL_CATCH_ANY
 osclerror, 87

OSCL_CHAR_IS_SIGNED
 osclconfig_limits_typedefs.h, 806

OSCL_CHAR_IS_UNSIGNED
 osclconfig_limits_typedefs.h, 806

oscl_chdir
 oscilio, 97

oscl_CIstrcmp
 osclbase, 35, 36

oscl_CIstrncmp
 osclbase, 36

OSCL_CLEANUP_BASE_CLASS
 osclmemory, 53

OSCL_CLOCK_HAS_DRIFT_CORRECTION
 osclconfig_util.h, 827

OSCL_COND_EXPORT_REF
 osclbase, 31

OSCL_COND_IMPORT_REF
 osclbase, 31

OSCL_CONST_CAST
 osclbase, 31

oscl_cos
 oscoutil, 72

Oscl_Dealloc, 168
 deallocate, 168

Oscl_DefAlloc, 169

Oscl_DefAlloc
 allocate, 169
 allocate_fl, 169
 deallocate, 169

oscl_defalloc.h, 651

Oscl_DefAllocWithRefCounter, 170

Oscl_DefAllocWithRefCounter
 addRef, 170
 Delete, 170
 getCount, 170
 New, 171
 removeRef, 171

OSCL_DEFAULT_FREE
 osclmemory, 54

OSCL_DEFAULT_MALLOC
 osclmemory, 54

OSCL_DELETE
 osclmemory, 54

Oscl_DeleteFile

Oscl_FileServer, 190, 191

OSCL_DISABLE_INLINES
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826

OSCL_DISABLE_WARNING_RETURN_-TYPE_NOT_UDT
 osclbase, 31
 osclmemory, 54

OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE
 oscl_map.h, 692
 oscl_mem.h, 698
 oscl_mem_audit.h, 701
 oscl_mem_audit_internals.h, 702
 oscl_mem_auto_ptr.h, 703
 oscl_tagtree.h, 771
 oscl_tree.h, 780
 osclbase, 31
 osclmemory, 54

oscl_dll.h, 652

OSCL_DLL_ENTRY_POINT
 osclbase, 31

OSCL_DLL_ENTRY_POINT_DEFAULT
 osclbase, 32

oscl_dns.h, 653

oscl_dns_gethostbyname.h, 654

oscl_dns_imp.h, 655

oscl_dns_imp_base.h, 656

oscl_dns_imp_pv.h, 657

oscl_dns_method.h, 658

oscl_dns_param.h, 659
 TDNSRequestParamAllocator, 659

oscl_dns_request.h, 660

oscl_dns_tuneables.h, 661
 PV_DNS_IS_THREAD, 661
 PV_DNS_SERVER, 661

oscl_double_list.h, 662

OSCL_DYNAMIC_CAST
 osclbase, 32

OSCL_ERR_NONE
 osclerror, 87

oscl_errno.h, 663

oscl_error.h, 664

oscl_error_allocator.h, 665

oscl_error_codes.h, 666

oscl_error_imp.h, 667

oscl_error_imp_cppexceptions.h, 668

oscl_error_imp_fatalerror.h, 669
 _PV_TRAP, 669
 _PV_TRAP_NO_TLS, 669

PVError_DoLeave, 669

oscl_error_imp_jumps.h, 670
 _PV_TRAP, 670
 _PV_TRAP_NO_TLS, 670

PVError_DoLeave, 671
 oscl_error_trapcleanup.h, 672
 oscl_exception.h, 673
 OSCL_EXCEPTSET_FLAG
 oscl_socket_serv_imp_pv.h, 752
 oscl_exclusive_ptr.h, 674
 oscl_exp
 osclutil, 72
 OSCL_FastString, 172
 OSCL_FastString, 173
 OSCL_FastString
 ~OSCL_FastString, 173
 chartype, 173
 get_cstr, 174
 get_maxsize, 174
 get_size, 174
 get_str, 174
 operator=, 174
 optype, 173
 OSCL_FastString, 173
 OSCL_String, 175
 other_chartype, 173
 set, 174, 175
 set_length, 175
Oscl_File
 ESymbianAccessMode_Rfile, 177
 ESymbianAccessMode_RfileBuf, 177
 MODE_APPEND, 177
 MODE_BINARY, 177
 MODE_READ, 177
 MODE_READ_PLUS, 177
 MODE_READWRITE, 177
 MODE_TEXT, 177
 SEEKCUR, 177
 SEEKEND, 177
 SEEKSET, 177
Oscl_File, 176
 ~Oscl_File, 178
 AddFixedCache, 178
 asyncfilereadcancel_test, 183
 asyncfilereadwrite_test, 183
 Close, 178
 EndOfFile, 178
 Flush, 178
 GetError, 179
 Handle, 179
 largeasyncfilereadwrite_test, 183
 mode_type, 177
 Open, 179
 Oscl_File, 177, 178
 Oscl_FileServer, 191
 OsclFileCache, 183
 OsclFileCacheBuffer, 183
 OsclFileHandle, 400
 Read, 180
 RemoveFixedCache, 180
 Seek, 180
 seek_type, 177
 SetAsyncReadBufferSize, 180
 SetCacheObserver, 180
 SetFileHandle, 181
 SetLoggingEnable, 181
 SetNativeAccessMode, 181
 SetNativeBufferSize, 181
 SetPVCacheSize, 182
 SetSummaryStatsLoggingEnable, 182
 Size, 182
 Tell, 182
 TSymbianAccessMode, 177
 Write, 182
 Oscl_File::OsclCacheObserver, 184
 Oscl_File::OsclCacheObserver
 ChooseCurCache, 184
 Oscl_File::OsclFixedCacheParam, 185
 Oscl_File::OsclFixedCacheParam
 Contains, 185
 iFilePath, 185
 iSize, 185
 oscl_file_async_read.h, 675
 OSCL_FILE_BUFFER_MAX_SIZE
 osclconfig_io.h, 797
 oscl_file_cache.h, 676
 OSCL_FILE_CHAR_PATH_DELIMITER
 osclio, 95
 oscl_file_dir_utils.h, 677
 oscl_file_find.h, 679
 oscl_file_handle.h, 680
 oscl_file_io.h, 681
 oscl_file_native.h, 682
 oscl_file_server.h, 683
 oscl_file_stats.h, 684
 OSCL_FILE_STATS_LOGGER_NODE
 osclio, 95
 oscl_file_types.h, 685
 OSCL_FILE_WCHAR_PATH_DELIMITER
 osclio, 95
Oscl_FileFind
 DIR_TYPE, 186
 E_BUFFER_TOO_SMALL, 187
 E_INVALID_ARG, 186
 E_INVALID_STATE, 186
 E_MEMORY_ERROR, 187
 E_NO_MATCH, 187
 E_NOT_IMPLEMENTED, 187
 E_OK, 186
 E_OTHER, 187
 E_PATH_NOT_FOUND, 186
 E_PATH_TOO_LONG, 186

FILE_TYPE, 186
 INVALID_TYPE, 186
Oscl_FileFind, 186
 Oscl_FileFind, 187
Oscl_FileFind
 ~Oscl_FileFind, 187
 Close, 187
 element_type, 186
 error_type, 186
 FindFirst, 187
 FindNext, 188
 GetElementType, 188
 GetLastError, 188
 Oscl_FileFind, 187
OSCL_FILEMGMT_E_ALREADY_EXISTS
 osclio, 96
OSCL_FILEMGMT_E_NO_MATCH
 osclio, 96
OSCL_FILEMGMT_E_NOT_EMPTY
 osclio, 96
OSCL_FILEMGMT_E_NOT_
 IMPLEMENTED
 osclio, 96
OSCL_FILEMGMT_E_OK
 osclio, 96
OSCL_FILEMGMT_E_PATH_NOT_FOUND
 osclio, 96
OSCL_FILEMGMT_E_PATH_TOO_LONG
 osclio, 96
OSCL_FILEMGMT_E_PERMISSION_
 DENIED
 osclio, 96
OSCL_FILEMGMT_E_SYS_SPECIFIC
 osclio, 96
OSCL_FILEMGMT_E_UNKNOWN
 osclio, 96
OSCL_FILEMGMT_ERR_TYPE
 osclio, 96
OSCL_FILEMGMT_MODE_DIR
 osclio, 96
OSCL_FILEMGMT_MODES
 osclio, 96
OSCL_FILEMGMT_PERMS
 osclio, 96
OSCL_FILEMGMT_PERMS_EXECUTE
 osclio, 96
OSCL_FILEMGMT_PERMS_READ
 osclio, 96
OSCL_FILEMGMT_PERMS_WRITE
 osclio, 96
Oscl_FileServer, 190
 Oscl_FileServer, 190
Oscl_FileServer
 ~Oscl_FileServer, 190

 Close, 190
 Connect, 190
 Oscl_DeleteFile, 190, 191
 Oscl_File, 191
 Oscl_FileServer, 190
 OsclNativeFile, 191
OSCL_FIRST_CATCH
 osclerror, 87
OSCL_FIRST_CATCH_ANY
 osclerror, 87
oscl_floor
 osclutil, 72
OSCL_FREE
 osclmemory, 54
oscl_free
 osclmemory, 54
OSCL_FSSTAT
 osclio, 95
oscl_fsstat, 192
 freebytes, 192
 totalbytes, 192
OSCL_FUNCTION_PTR
 osclconfig_compiler_warnings.h, 790
oscl_getcwd
 osclio, 97, 98
OSCL_GetLastError
 osclerror, 91
OSCL_HAS_ANDROID_FILE_IO_SUPPORT
 osclconfig.h, 787
OSCL_HAS_ANDROID_SUPPORT
 osclconfig.h, 787
OSCL_HAS_ANSI_64BIT_FILE_IO_-
 SUPPORT
 osclconfig_io.h, 797
OSCL_HAS_ANSI_FILE_IO_SUPPORT
 osclconfig_io.h, 797
OSCL_HAS_ANSI_MATH_SUPPORT
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_HAS_ANSI_MEMORY_FUNCS
 osclconfig_ansi_memory.h, 788
OSCL_HAS_ANSI_STDIO_SUPPORT
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_HAS_ANSI_STDLIB_SUPPORT
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_HAS_ANSI_STRING_SUPPORT
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_HAS_ANSI_WIDE_STRING_-
 SUPPORT
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826

OSCL_HAS_BASIC_LOCK
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826

OSCL_HAS_BERKELEY_SOCKETS
 osclconfig, 21
 osclconfig_io.h, 797

OSCL_HAS_ERRNO_H
 osclconfig_error.h, 791

OSCL_HAS_EXCEPTIONS
 osclconfig_error.h, 791

OSCL_HAS_GLOB
 osclconfig_io.h, 797

OSCL_HAS_GLOBAL_NEW_DELETE
 osclconfig_memory.h, 807
 osclmemory, 54

OSCL_HAS_GLOBAL_VARIABLE_- SUPPORT
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826

OSCL_HAS_HEAP_BASE_SUPPORT
 osclconfig_memory.h, 807

OSCL_HAS_LARGE_FILE_SUPPORT
 osclconfig_io.h, 797

OSCL_HAS_MSWIN_FILE_IO_SUPPORT
 osclconfig_io.h, 797

OSCL_HAS_MSWIN_PARTIAL_SUPPORT
 osclconfig, 21

OSCL_HAS_MSWIN_SUPPORT
 osclconfig, 21
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826

OSCL_HAS_NATIVE_FILE_CACHE_- ENABLE
 osclconfig_io.h, 797

OSCL_HAS_NATIVE_INT64_TYPE
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826

OSCL_HAS_NATIVE_UINT64_TYPE
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826

OSCL_HAS_NON_PREEMPTIVE_- THREAD_SUPPORT
 osclconfig_proc_unix_android.h, 814
 osclconfig_proc_unix_common.h, 816

OSCL_HAS_PRAGMA_PACK
 osclconfig, 21

OSCL_HAS_PTHREAD_SUPPORT
 osclconfig, 21
 osclconfig_proc_unix_android.h, 814
 osclconfig_proc_unix_common.h, 816

OSCL_HAS_PV_C_OS_API_MEMORY_- FUNCS
 osclconfig, 22

OSCL_HAS_PV_C_OS_SUPPORT
 osclconfig, 22

osclconfig, 22

OSCL_HAS_PV_C_OS_TIME_FUNCS
 osclconfig, 22

OSCL_HAS_PV_FILE_CACHE
 osclconfig_io.h, 797

OSCL_HAS_RUNTIME_LIB_LOADING_- SUPPORT
 osclconfig_lib.h, 804

OSCL_HAS_SAVAJE_IO_SUPPORT
 osclconfig, 22

OSCL_HAS_SAVAJE_SUPPORT
 osclconfig, 22

OSCL_HAS_SEM_TIMEDWAIT_SUPPORT
 osclconfig, 22
 osclconfig_proc_unix_android.h, 814
 osclconfig_proc_unix_common.h, 816

OSCL_HAS_SETJMP_H
 osclconfig_error.h, 791

OSCL_HAS_SINGLETON_SUPPORT
 osclbase, 32

OSCL_HAS_SOCKET_SUPPORT
 osclconfig_io.h, 797

OSCL_HAS_SYMBIAN_COMPATIBLE_IO_- FUNCTION
 osclconfig, 22
 osclconfig_io.h, 797

OSCL_HAS_SYMBIAN_DNS_SERVER
 osclconfig, 22
 osclconfig_io.h, 797

OSCL_HAS_SYMBIAN_ERRORTRAP
 osclconfig, 22
 osclconfig_error.h, 791

OSCL_HAS_SYMBIAN_MATH
 osclconfig, 22
 osclconfig_util.h, 827

OSCL_HAS_SYMBIAN_MEMORY_FUNCS
 osclconfig, 22
 osclconfig_memory.h, 807

OSCL_HAS_SYMBIAN_SCHEDULER
 osclconfig, 22
 osclconfig_proc_unix_android.h, 814
 osclconfig_proc_unix_common.h, 816

OSCL_HAS_SYMBIAN_SOCKET_SERVER
 osclconfig, 22
 osclconfig_io.h, 797

OSCL_HAS_SYMBIAN_SUPPORT
 osclconfig, 22
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826

OSCL_HAS_SYMBIAN_TIMERS
 osclconfig, 22
 osclconfig_util.h, 827

OSCL_HAS_THREAD_SUPPORT
 osclconfig_proc_unix_android.h, 814

osclconfig_proc_unix_common.h, 816
OSCL_HAS_TLS_SUPPORT
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_HAS_UNICODE_SUPPORT
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_HAS_UNIX_SUPPORT
 osclconfig, 22
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_HAS_UNIX_TIME_FUNCS
 osclconfig, 22
 osclconfig_time.h, 817
 oscl_heapbase.h, 686
OSCL_HeapString, 193
 osclutil, 72, 73
OSCL_HeapString
 chartype, 194
 optype, 194
 OSCL_String, 194
 other_chartype, 194
OSCL_HeapStringA, 195
 OSCL_HeapStringA, 196, 197
OSCL_HeapStringA
 ~OSCL_HeapStringA, 197
 chartype, 196
 get_cstr, 197
 get_maxsize, 197
 get_size, 198
 get_str, 198
 operator=, 198
 optype, 196
 OSCL_HeapStringA, 196, 197
 OSCL_String, 199
 other_chartype, 196
 set, 198, 199
 oscl_init.h, 687
OSCL_INLINE
 osclbase, 32
Oscl_Int64_Utils, 200
 get_int64_lower32, 201
 get_int64_middle32, 201
 get_int64_upper32, 201
 get_uint64_lower32, 201
 get_uint64_middle32, 201
 get_uint64_upper32, 201
 set_int64, 201
 set_uint64, 201
 oscl_int64_utils.h, 688
 _OsclInteger64Transport, 688
OSCL_INTEGERS_WORD_ALIGNED
 osclconfig, 22
OSCL_IO_EXTENSION_MAXLEN
 osclio, 95
OSCL_IO_FILENAME_MAXLEN
 osclio, 95
oscl_ip_socket.h, 689
OSCL IPPROTO_TCP
 osclconfig_io.h, 797
OSCL IPPROTO_UDP
 osclconfig_io.h, 797
oscl_isdigit
 osclutil, 67
OSCL_IsErrnoSupported
 osclerror, 91
oscl_isLetter
 osclbase, 36
OSCL_JUMP_MAX_JUMP_MARKS
 osclerror, 87
OSCL_LAST_CATCH
 osclerror, 87
OSCL_LEAVE
 osclerror, 88
Oscl_Less, 202
 operator(), 202
OSCL_LIB_READ_DEBUG_LIBS
 osclconfig_lib.h, 804
Oscl_Linked_List, 203
 ~Oscl_Linked_List, 203
 add_element, 204
 add_to_front, 204
 check_list, 204
 dequeue_element, 204
 get_element, 204
 get_first, 204
 get_index, 205
 get_next, 205
 get_num_elements, 205
 move_to_end, 205
 move_to_front, 205
 Oscl_Linked_List, 203
 remove_element, 206
oscl_linked_list.h, 690
Oscl_Linked_List_Base, 207
 ~Oscl_Linked_List_Base, 208
 add_element, 208
 add_to_front, 208
 check_list, 208
 construct, 208
 destroy, 208
 get_element, 208
 get_first, 209
 get_index, 209
 get_next, 209
 head, 210
 iterator, 210
 move_to_end, 209

move_to_front, 209
 num_elements, 210
 remove_element, 210
 sizeof_T, 210
 tail, 210
 oscl_lock_base.h, 691
 oscl_log
 osclutil, 73
 oscl_log10
 osclutil, 73
 OSCL_MALLOC
 osclmemory, 55
 oscl_malloc
 osclmemory, 55
 Oscl_Map, 211
 begin, 214
 clear, 214
 const_iterator, 213
 const_reference, 213
 count, 214
 empty, 214
 end, 214
 equal_range, 214
 erase, 215
 find, 215
 insert, 215
 iterator, 213
 key_comp, 216
 key_compare, 213
 key_type, 213
 lower_bound, 216
 max_size, 216
 operator=, 216
 operator[], 216
 Oscl_Map, 213
 pair_citerator_citerator, 213
 pair_iterator_bool, 213
 pair_iterator_iterator, 213
 pointer, 213
 reference, 213
 self, 213
 size, 216
 size_type, 213
 upper_bound, 216, 217
 value_comp, 217
 value_type, 213
 oscl_map.h, 692
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 692
 Oscl_Map::value_compare, 218
 comp, 218
 operator(), 218
 Oscl_Map< Key, T, Alloc, Compare >, 218
 value_compare, 218
 Oscl_Map< Key, T, Alloc, Compare >
 Oscl_Map::value_compare, 218
 oscl_math.h, 693
 OSCL_MAX
 osclbase, 32
 OSCL_MAX_TRAP_LEVELS
 osclerror, 88
 oscl_media_data.h, 694
 oscl_media_status.h, 695
 oscl_mem.h, 696
 operator delete, 698
 operator new, 698
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 698
 oscl_mem_align.h, 699
 oscl_mem_aligned_size
 osclmemory, 58
 OsclMemPoolAllocator, 432
 oscl_mem_audit.h, 700
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 701
 oscl_mem_audit_internals.h, 702
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 702
 oscl_mem_auto_ptr.h, 703
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 703
 oscl_mem_basic_functions.h, 704
 oscl_mem_inst.h, 705
 oscl_mem_mempool.h, 706
 oscl_memcmp
 osclmemory, 59
 oscl_memcpy
 osclmemory, 59
 OSCL_MEMFRAG_PTR_BEFORE_LEN
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
 oscl_memmove
 osclmemory, 59
 oscl_memmove32
 osclmemory, 59
 oscl_mempool_allocator.h, 707
 oscl_memset
 osclmemory, 60
 oscl_memsize_t
 osclconfig_ansi_memory.h, 788
 OSCL_MIN
 osclbase, 32
 oscl_mkdir

osclio, 98
Oscl_MTLinked_List, 220
 ~Oscl_MTLinked_List, 220
 add_element, 221
 add_to_front, 221
 dequeue_element, 221
 get_element, 221
 get_index, 221
 move_to_end, 221
 move_to_front, 222
Oscl_MTLinked_List, 220
 remove_element, 222
 the_list, 222
oscl_mutex.h, 708
 OsclNoYieldMutex, 708
oscl_namestring.h, 709
OSCL_NATIVE_INT64_TYPE
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_NATIVE_UINT64_TYPE
 osclconfig.h, 787
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_NATIVE_WCHAR_TYPE
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_NEW
 osclmemory, 55
oscl_opaque_type.h, 710
Oscl_Opaque_Type_Alloc, 224
 allocate, 224
 construct, 224
 deallocate, 224
 destroy, 224
Oscl_Opaque_Type_Alloc_LL, 225
 allocate, 225
 compare_data, 225
 construct, 225
 deallocate, 225
 destroy, 225
 get_data, 226
 get_next, 226
 set_next, 226
Oscl_Opaque_Type_Compare, 227
 compare_EQ, 227
 compare_LT, 227
 swap, 227
OSCL_PACKED_STRUCT_BEGIN
 osclconfig.h, 787
OSCL_PACKED_STRUCT_END
 osclconfig.h, 787
OSCL_PACKED_VAR
 osclbase, 32
 osclconfig.h, 787
Oscl_Pair, 229
 first, 229
Oscl_Pair, 229
 second, 229
OSCL_PERF_SUMMARY_LOGGING
 osclproc, 103
OSCL_PLACEMENT_NEW
 osclmemory, 55
oscl_pow
 osclutil, 73
oscl_priqueue.h, 711
oscl_priqueue_test
 OsclPriorityQueue, 463
oscl_procstatus.h, 712
Oscl_Queue, 230
 ~Oscl_Queue, 231
 back, 231
 clear, 231
 const_reference, 231
 front, 232
Oscl_Queue, 231
 pointer, 231
 pop, 232
 push, 232
 reference, 231
 size_type, 231
 value_type, 231
oscl_queue.h, 713
Oscl_Queue_Base, 233
 ~Oscl_Queue_Base, 233
 bufsize, 235
 capacity, 234
 clear, 234
 construct, 234
 destroy, 234
 elems, 235
 empty, 234
 ifront, 235
 irear, 235
 numelems, 235
 pop, 234
 push, 234
 reserve, 234
 size, 234
 sizeof_T, 235
oscl_rand.h, 714
OSCL RAND MAX
 osclconfig_util.h, 827
Oscl_Rb_Tree, 236
 ~Oscl_Rb_Tree, 238
 begin, 238
 clear, 238
 const_iterator, 238
 const_pointer, 238

const_reference, 238
count, 238
difference_type, 238
empty, 238
end, 238
equal_range, 238
erase, 238
find, 238
insert_unique, 238
iterator, 238
key_type, 238
link_type, 238
lower_bound, 238
max_size, 238
operator=, 238
Oscl_Rb_Tree, 238
pointer, 238
reference, 238
size, 238
size_type, 238
upper_bound, 238
value_type, 238
Oscl_Rb_Tree_Base, 240
base_link_type, 240
rebalance, 240
rebalance_for_erase, 240
rotate_left, 240
rotate_right, 240
Oscl_Rb_Tree_Const_Iterator, 241
base_link_type, 242
const_iterator, 242
link_type, 242
node, 242
operator *, 242
operator!=, 242
operator++, 242
operator-, 242
operator->, 242
operator==, 242
Oscl_Rb_Tree_Const_Iterator, 242
pointer, 242
reference, 242
self, 242
value_type, 242
Oscl_Rb_Tree_Iterator, 244
base_link_type, 245
iterator, 245
link_type, 245
node, 245
operator *, 245
operator!=, 245
operator++, 245
operator-, 245
operator->, 245
operator==, 245
operator==, 245
Oscl_Rb_Tree_Iterator, 245
pointer, 245
reference, 245
self, 245
value_type, 245
Oscl_Rb_Tree_Node, 247
link_type, 247
value, 247
value_type, 247
Oscl_Rb_Tree_Node_Base
black, 248
red, 248
Oscl_Rb_Tree_Node_Base, 248
base_link_type, 248
color, 249
color_type, 248
left, 249
maximum, 249
minimum, 249
parent, 249
RedBl, 248
right, 249
OSCL_READSET_FLAG
oscl_socket_serv_imp_pv.h, 752
OSCL_REALLOC
osclmemory, 55
oscl_realloc
osclmemory, 55
oscl_refcounter.h, 715
oscl_refcounter_memfrag.h, 716
oscl_registry_access_client.h, 717
oscl_registry_client.h, 718
oscl_registry_client_impl.h, 719
oscl_registry_serv_impl.h, 720
oscl_registry_serv_impl_global.h, 721
oscl_registry_serv_impl_tls.h, 722
oscl_registry_types.h, 723
OSCL_REINTERPRET_CAST
osclbase, 32
OSCL_RELEASE_BUILD
osclconfig.h, 787
oscl_rename
osclio, 98, 99
OSCL_REQUEST_ERR_CANCEL
osclproc, 104
OSCL_REQUEST_ERR_GENERAL
osclproc, 104
OSCL_REQUEST_ERR_NONE
osclproc, 104
OSCL_REQUEST_PENDING
osclproc, 104
oscl_rmdir
osclio, 99

oscl_scheduler.h, 724
 oscl_scheduler_ao.h, 725
 oscl_scheduler_aobase.h, 726
 oscl_scheduler_readyq.h, 727
 oscl_scheduler_threadcontext.h, 728
 oscl_scheduler_tuneables.h, 729
 oscl_scheduler_types.h, 730
OSCL_SD_BOTH
 osclconfig_io.h, 797
OSCL_SD_RECEIVE
 osclconfig_io.h, 797
OSCL_SD_SEND
 osclconfig_io.h, 797
Oscl_Select1st, 250
 operator(), 250
 oscl_semaphore.h, 731
OSCL_SetLastError
 osclerror, 91
 oscl_shared_ptr.h, 732
 oscl_sin
 osclutil, 74
 oscl_singleton.h, 733
 OSCL_SINGLETON_ID_CPM_PLUGIN,
 734
 OSCL_SINGLETON_ID_LAST, 734
 OSCL_SINGLETON_ID_OMX, 734
 OSCL_SINGLETON_ID_-
 OMXMASTERCORE, 734
 OSCL_SINGLETON_ID_OSCLMEM,
 734
 OSCL_SINGLETON_ID_-
 OSCLREGISTRY, 734
 OSCL_SINGLETON_ID_-
 PAYLOADPARSER, 734
 OSCL_SINGLETON_ID_-
 PVERRORTRAP, 734
 OSCL_SINGLETON_ID_PVLOGGER,
 734
 OSCL_SINGLETON_ID_-
 PVMFRECOGNIZER, 734
 OSCL_SINGLETON_ID_-
 PVSCHEDULER, 734
 OSCL_SINGLETON_ID_-
 SDPMEDIAPARSER, 734
 OSCL_SINGLETON_ID_TEST, 734
 OSCL_SINGLETON_ID_TICKCOUNT,
 734
 OSCL_SINGLETON_ID_-
 WMDRMLOCK, 734
OSCL_SINGLETON_ID_CPM_PLUGIN
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_LAST
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_OMX

oscl_singleton.h, 734
OSCL_SINGLETON_ID_-
 OMXMASTERCORE
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_OSCLMEM
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_OSCLREGISTRY
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_PAYLOADPARSER
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_PVERRORTRAP
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_PVLOGGER
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_-
 PVMFRECOGNIZER
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_PVSCHEDULER
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_-
 SDPMEDIAPARSER
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_TEST
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_TICKCOUNT
 oscl_singleton.h, 734
OSCL_SINGLETON_ID_WMDRMLOCK
 oscl_singleton.h, 734
 oscl_snprintf
 osclutil, 74
 oscl_snprintf.h, 735
OSCL_SOCKET_DATAGRAM
 osclconfig_io.h, 797
OSCL_SOCKET_STREAM
 osclconfig_io.h, 797
 oscl_socket.h, 736
 oscl_socket_accept.h, 737
 oscl_socket_bind.h, 738
 oscl_socket_connect.h, 739
 oscl_socket_imp.h, 740
 oscl_socket_imp_base.h, 741
 oscl_socket_imp_pv.h, 742
 PVSOCK_ERR_BAD_PARAM, 742
 PVSOCK_ERR_NOT_IMPLEMENTED,
 742
 PVSOCK_ERR_SERV_NOT_-
 CONNECTED, 742
 PVSOCK_ERR SOCK_NO_SERV, 742
 PVSOCK_ERR SOCK_NOT_-
 CONNECTED, 742
 PVSOCK_ERR SOCK_NOT_OPEN, 742
 oscl_socket_listen.h, 743
 OSCL_SOCKET_LISTEN_H_-
 INCLUDEDd, 743

OSCL_SOCKET_LISTEN_H_INCLUDEDd
oscl_socket_listen.h, 743

oscl_socket_method.h, 744
MSEC_TO_MICROSEC, 744

oscl_socket_recv.h, 745

oscl_socket_recv_from.h, 746

oscl_socket_request.h, 747

oscl_socket_send.h, 748

oscl_socket_send_to.h, 749

oscl_socket_serv_imp.h, 750

oscl_socket_serv_imp_base.h, 751

oscl_socket_serv_imp_pv.h, 752
OSCL_EXCEPTSET_FLAG, 752
OSCL_READSET_FLAG, 752
OSCL_WRITESET_FLAG, 752

oscl_socket_serv_imp_reqlist.h, 753

oscl_socket_shutdown.h, 754

oscl_socket_stats.h
EOsclSocket_DataRecv, 756
EOsclSocket_DataSent, 756
EOsclSocket_Except, 755
EOsclSocket_OS, 755
EOsclSocket_Readable, 755
EOsclSocket_RequestAO_Canceled, 755
EOsclSocket_RequestAO_Error, 755
EOsclSocket_RequestAO_Success, 755
EOsclSocket_RequestAO_Timeout, 755
EOsclSocket_ServPoll, 755
EOsclSocket_ServRequestCancelIssued, 756
EOsclSocket_ServRequestComplete, 756
EOsclSocket_ServRequestIssued, 755
EOsclSocket_Writable, 755
EOsclSocketServ_LastEvent, 755
EOsclSocketServ_LoopsckError, 756
EOsclSocketServ_LoopsckOk, 756
EOsclSocketServ_SelectActivity, 755
EOsclSocketServ_SelectNoActivity, 755
EOsclSocketServ_SelectRescheduleAsap, 755
EOsclSocketServ_SelectReschedulePoll, 755

oscl_socket_stats.h, 755

TOsclSocketServStatEvent, 755

TOsclSocketStatEvent, 755

oscl_socket_tuneables.h, 757

PV_OSCL_SOCKET_1MB_RECV_BUF,
757

PV_OSCL_SOCKET_SERVER_-
LOGGER_OUTPUT, 757

PV_OSCL_SOCKET_STATS_LOGGING,
757

PV_SOCKET_REQUEST_AO_-
PRIORITY, 757

PV_SOCKET_SERVER, 757

PV_SOCKET_SERVER_AO_-
INTERVAL_MSEC, 758

PV_SOCKET_SERVER_AO_PRIORITY,
758

PV_SOCKET_SERVER_IS_THREAD,
758

PV_SOCKET_SERVER_SELECT, 758

PV_SOCKET_SERVER_SELECT_-
LOOPBACK_SOCKET, 758

PV_SOCKET_SERVER_SELECT_-
TIMEOUT_MSEC, 758

PV_SOCKET_SERVER_THREAD_-
PRIORITY, 758

PV_SOCKET_SERVI_STATS, 758

oscl_socket_types.h

EPVSocket_Last, 760

EPVSocketAccept, 760

EPVSocketBind, 760

EPVSocketBothShutdown, 760

EPVSocketCancel, 759

EPVSocketConnect, 760

EPVSocketFailure, 759

EPVSocketListen, 760

EPVSocketPending, 759

EPVSocketRecv, 760

EPVSocketRecvFrom, 760

EPVSocketRecvShutdown, 760

EPVSocketSend, 760

EPVSocketSendShutdown, 760

EPVSocketSendTo, 760

EPVSocketShutdown, 760

EPVSocketSuccess, 759

EPVSocketTimeout, 759

oscl_socket_types.h, 759

PVNETWORKADDRESS_LEN, 759

TPVSocketEvent, 759

TPVSocketFxn, 759

TPVSocketShutdown, 760

oscl_sqrt

osclutil, 74

OSCL_StackString, 251

osclutil, 74, 75

OSCL_StackString

chartype, 252

otype, 252

OSCL_String, 252

other_chartype, 252

oscl_stat

osclio, 99, 100

OSCL_STAT_BUF

osclio, 95

oscl_stat_buf, 253

mode, 253

perms, 253
 oscl_statfs
 osclio, 100
 OSCL_STATIC_CAST
 osclbase, 32
 oscl_stdstring.h, 761
 oscl_str_escape_xml
 osclutil, 75
 oscl_str_is_valid_utf8
 osclutil, 75
 oscl_str_need_escape_xml
 osclutil, 76
 oscl_str_ptr_len.h, 763
 oscl_str_truncate_utf8
 osclutil, 76
 oscl_str_unescape_uri
 osclutil, 76, 77
 oscl_strcat
 osclbase, 37
 oscl_strchr
 osclbase, 37, 38
 oscl_strcmp
 osclbase, 38
 OSCL_StrError
 osclerror, 91
 OSCL_String, 254
 ~OSCL_String, 255
 append_rep, 255
 chartype, 255
 get_cstr, 255
 get_maxsize, 255
 get_size, 256
 get_str, 256
 hash, 256
 is_writable, 256
 operator!=, 256
 operator+=, 256
 operator<, 257
 operator<=, 257
 operator=, 257
 operator==, 257
 operator>, 257
 operator>=, 257
 operator[], 257
 OSCL_FastString, 175
 OSCL_HeapString, 194
 OSCL_HeapStringA, 199
 OSCL_StackString, 252
 OSCL_String, 255
 read, 257
 set_len, 258
 set_rep, 258
 setrep_to_char, 258
 write, 258
 oscl_string.h, 764
 oscl_string_containers.h, 765
 oscl_string_rep.h, 766
 oscl_string_uri.h, 767
 oscl_string_utf8.h, 768
 oscl_string_utils.h, 769
 oscl_string_xml.h, 770
 oscl_strlen
 osclbase, 38
 oscl_strncat
 osclbase, 39
 oscl_strncmp
 osclbase, 39, 40
 oscl_strncpy
 osclbase, 40
 oscl strrchr
 osclbase, 41
 oscl_strset
 osclbase, 41
 oscl strstr
 osclbase, 41, 42
 Oscl_Tag, 260
 ~Oscl_Tag, 260
 operator<, 260
 Oscl_Tag, 260
 tag, 260
 tagAllocator, 260
 Oscl_Tag_Base, 262
 operator(), 263
 size_type, 263
 tag_ancestor, 263
 tag_base_type, 263
 tag_base_unit, 263
 tag_cmp, 263
 tag_copy, 263
 tag_depth, 263
 tag_len, 263
 Oscl_TagTree, 264
 Oscl_TagTree, 265
 Oscl_TagTree
 ~Oscl_TagTree, 265
 begin, 265
 children_type, 265
 clear, 266
 count, 266
 empty, 266
 end, 266
 erase, 266
 find, 266
 insert, 267
 map_type, 265
 node_ptr, 265
 node_type, 265
 operator=, 267

operator[], 267
 Oscl_TagTree, 265
 pair_iterator_bool, 265
 size, 267
 size_type, 265
 tag_base_type, 265
 tag_type, 265
 value_type, 265
 oscl_tagtree.h, 771
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 771
 Oscl_TagTree::const_iterator, 268
 Oscl_TagTree::const_iterator
 const_iterator, 269
 mapit, 269
 mapiter, 269
 operator *, 269
 operator!=, 269
 operator++, 269
 operator-, 269
 operator->, 269
 operator==, 269
 pointer, 269
 reference, 269
 self, 269
 Oscl_TagTree::iterator, 271
 Oscl_TagTree::iterator
 iterator, 272
 mapit, 272
 mapiter, 272
 operator *, 272
 operator!=, 272
 operator++, 272
 operator-, 272
 operator->, 272
 operator==, 272
 pointer, 272
 reference, 272
 self, 272
 Oscl_TagTree::Node, 274
 Oscl_TagTree::Node
 children, 275
 children_type, 275
 depth, 275
 Node, 275
 parent, 275
 sort_children, 275
 tag, 275
 value, 275
 Oscl_TAlloc, 276
 ~Oscl_TAlloc, 277
 address, 277
 alloc_and_construct, 277
 alloc_and_construct_fl, 277
 allocate, 277
 allocate_fl, 277
 const_pointer, 277
 const_reference, 277
 construct, 277
 deallocate, 277
 destroy, 277
 destruct_and_dealloc, 277
 pointer, 277
 reference, 277
 size_type, 277
 value_type, 277
 Oscl_TAlloc::rebind, 279
 other, 279
 oscl_tan
 osclutil, 77
 OSCL_TCHAR
 osclbase, 33
 oscl_tcp_socket.h, 772
 OSCL_TEMPLATED_DESTRUCTOR_CALL
 osclbase, 32
 osclconfig.h, 787
 oscl_thread.h
 Start_on_creation, 773
 Suspend_on_creation, 773
 ThreadPriorityAboveNormal, 774
 ThreadPriorityBelowNormal, 773
 ThreadPriorityHighest, 774
 ThreadPriorityLow, 773
 ThreadPriorityLowest, 773
 ThreadPriorityNormal, 773
 ThreadPriorityTimeCritical, 774
 oscl_thread.h, 773
 OsclThread_State, 773
 OsclThreadPriority, 773
 TOsclThreadFuncPtr, 773
 OSCL_THREAD DECL
 osclconfig_proc_unix_android.h, 814
 osclconfig_proc_unix_common.h, 816
 oscl_tickcount.h, 775
 oscl_time.h, 776
 oscl_timer.h, 778
 oscl_tls.h, 779
 OSCL_TLS_BASE_SLOTS
 osclbase, 32
 OSCL_TLS_EXTERNAL_SLOTS
 osclbase, 32
 OSCL_TLS_GET_FUNC
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
 OSCL_TLS_ID_BASE_LAST
 osclbase, 44
 OSCL_TLS_ID_ERRORHOOK

osclbase, 44
OSCL_TLS_ID_MAGICNUM
 osclbase, 44
OSCL_TLS_ID_OSCLREGISTRY
 osclbase, 44
OSCL_TLS_ID_PAYLOADPARSER
 osclbase, 44
OSCL_TLS_ID_PVERRORTRAP
 osclbase, 44
OSCL_TLS_ID_PVLOGGER
 osclbase, 44
OSCL_TLS_ID_PVMFRECOGNIZER
 osclbase, 44
OSCL_TLS_ID_PVSCHEDULER
 osclbase, 44
OSCL_TLS_ID_SDPMEDIAPARSER
 osclbase, 44
OSCL_TLS_ID_SQLITE3
 osclbase, 44
OSCL_TLS_ID_TEST
 osclbase, 44
OSCL_TLS_ID_WMDRM
 osclbase, 44
OSCL_TLS_IS_KEYED
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_TLS_KEY_CREATE_FUNC
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_TLS_KEY_DELETE_FUNC
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
OSCL_TLS_MAX_SLOTS
 osclbase, 32
OSCL_TLS_STORE_FUNC
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
oscl_tolower
 osclbase, 42
OSCL_TRAP_ALLOC_NEW
 osclmemory, 55
OSCL_TRAP_AUDIT_NEW
 osclmemory, 56
OSCL_TRAP_NEW
 osclmemory, 56
OSCL_TRAPSTACK_POP
 osclerror, 88
OSCL_TRAPSTACK_POPDEALLOC
 osclerror, 88
OSCL_TRAPSTACK_PUSH
 osclerror, 88
oscl_tree.h, 780
OSCL_DISABLE_WARNING_-
TRUNCATE_DEBUG_MESSAGE, 780
OSCL_TRY
 osclerror, 88
OSCL_TRY_NO_TLS
 osclerror, 88
OSCL_TStrPtrLen
 osclutil, 67
oscl_types.h, 781
oscl_udp_socket.h, 782
oscl_UnicodeToUTF8
 osclutil, 77
OSCL_UNSIGNED_CONST
 osclbase, 32
 osclconfig.h, 787
OSCL_UNUSED_ARG
 osclbase, 32
OSCL_UNUSED_RETURN
 osclbase, 33
oscl_utf8conv.h, 783
oscl_UTF8ToUnicode
 osclutil, 78
oscl_uuid.h, 784
 BYTES_IN_UUID_ARRAY, 784
OsclUid32, 784
 PV_CHAR_CLOSE_BRACKET, 784
 PV_CHAR_COMMA, 784
Oscl_Vector, 280
 ~Oscl_Vector, 281
 back, 282
 begin, 282
 clear, 282
 const_iterator, 281
 const_reference, 281
 destroy, 282
 end, 282
 erase, 282
 front, 283
 insert, 283
 iterator, 281
 operator=, 283
 operator[], 283
Oscl_Vector, 281
 pointer, 281
 pop_back, 283
 push_back, 284
 push_front, 284
 reference, 281
 value_type, 281
oscl_vector.h, 785
Oscl_Vector_Base, 285
 ~Oscl_Vector_Base, 286
 assign_vector, 286
 bufsize, 288
 capacity, 286

construct, 286
 destroy, 286
 elems, 288
 empty, 286
 erase, 286, 287
 insert, 287
 numelems, 288
 OsclPriorityQueueBase, 288
 pop_back, 287
 push_back, 287
 push_front, 288
 reserve, 288
 size, 288
 sizeof_T, 288
 oscl_vsnprintf
 osclutil, 78, 80
 oscl_wchar
 osclbase, 33
 OSCL_wFastString, 289
 OSCL_wFastString, 290
 OSCL_wFastString
 ~OSCL_wFastString, 290
 chartype, 289
 get_cstr, 290
 get_maxsize, 290
 get_size, 290
 get_str, 290
 operator=, 290
 optype, 289
 OSCL_wFastString, 290
 OSCL_wString, 291
 other_chartype, 290
 set, 291
 set_length, 291
 OSCL_wHeapString, 292
 osclutil, 80
 OSCL_wHeapString
 chartype, 293
 optype, 293
 OSCL_wString, 293
 other_chartype, 293
 OSCL_wHeapStringA, 294
 OSCL_wHeapStringA, 295
 OSCL_wHeapStringA
 ~OSCL_wHeapStringA, 295
 chartype, 295
 get_cstr, 295
 get_maxsize, 295
 get_size, 295
 get_str, 296
 operator=, 296
 optype, 295
 OSCL_wHeapStringA, 295
 OSCL_wString, 296

other_chartype, 295
 set, 296
 OSCL_WRITESET_FLAG
 oscl_socket_serv_imp_pv.h, 752
 OSCL_wStackString, 297
 osclutil, 80
 OSCL_wStackString
 chartype, 298
 optype, 298
 OSCL_wString, 298
 other_chartype, 298
 OSCL_wString, 299
 OSCL_wFastString, 291
 OSCL_wHeapString, 293
 OSCL_wHeapStringA, 296
 OSCL_wStackString, 298
 OSCL_wString, 300
 OSCL_wString
 ~OSCL_wString, 300
 append_rep, 300
 chartype, 300
 get_cstr, 300
 get_maxsize, 300
 get_size, 300
 get_str, 300
 hash, 300
 is_writable, 301
 operator!=, 301
 operator+=, 301
 operator<, 301
 operator<=, 301
 operator=, 301
 operator==, 301
 operator>, 301
 operator>=, 301
 operator[], 301
 OSCL_wString, 300
 read, 301
 set_len, 302
 set_rep, 302
 setrep_to_wide_char, 302
 write, 302
 OSCL_ZEROIZE
 osclproc, 103
 OsclAccept
 osclconfig_io.h, 797
 OsclAcceptMethod, 303
 OsclAcceptMethod
 ~OsclAcceptMethod, 303
 Accept, 303
 AcceptRequest, 303
 DiscardAcceptedSocket, 303
 GetAcceptedSocket, 303
 NewL, 303

OsclAcceptRequest, 304
 OsclAcceptRequest, 304
 OsclSocketI, 528
 OsclAcceptRequest
 Accept, 304
 OsclAcceptRequest, 304
 OsclActiveObject, 305
 EPriorityHigh, 306
 EPriorityHighest, 306
 EPriorityIdle, 306
 EPriorityLow, 306
 EPriorityNominal, 306
 OsclActiveObject, 306
 OsclExecSchedulerCommonBase, 392
 PVActiveBase, 600
 PVActiveStats, 601
 PVThreadContext, 620
 OsclActiveObject
 ~OsclActiveObject, 306
 AddToScheduler, 306
 Cancel, 306
 DoCancel, 307
 IsBusy, 307
 OsclActiveObject, 306
 OsclActivePriority, 306
 PendComplete, 307
 PendForExec, 307
 Priority, 307
 RemoveFromScheduler, 307
 RunError, 307
 RunIfNotReady, 308
 SetBusy, 308
 SetStatus, 308
 Status, 308
 StatusRef, 308
 OsclActivePriority
 OsclActiveObject, 306
 OsclAllocDestructDealloc, 309
 OsclAllocDestructDealloc
 ~OsclAllocDestructDealloc, 309
 OsclAny
 osclbase, 33
 OsclAOStatus, 310
 OsclAOStatus, 310
 OsclAOStatus
 operator!=, 310
 operator<, 310
 operator<=, 310
 operator=, 310
 operator==, 310
 operator>, 310
 operator>=, 310
 OsclAOStatus, 310
 Value, 310

OsclAsyncFile, 311
 OsclAsyncFile
 ~OsclAsyncFile, 312
 Close, 312
 Delete, 312
 EndOfFile, 312
 Flush, 312
 iNumOfRun, 313
 iNumOfRunErr, 313
 NewL, 312
 Open, 312, 313
 Read, 313
 Seek, 313
 Size, 313
 Tell, 313
 Write, 313
 OsclAsyncFileBuffer, 314
 OsclAsyncFileBuffer
 ~OsclAsyncFileBuffer, 315
 Buffer, 315
 CleanInUse, 315
 HasThisOffset, 315
 Id, 315
 IsInUse, 315
 IsValid, 315
 Length, 315
 NewL, 315
 Offset, 315
 SetInUse, 315
 SetOffset, 315
 StartAsyncRead, 315
 UpdateData, 315
 OsclAuditCB, 316
 OsclAuditCB, 316
 OsclAuditCB
 OsclAuditCB, 316
 pAudit, 316
 pStatsNode, 316
 OsclBase
 OsclSingletonRegistry, 524
 OsclTLSRegistry, 583
 osclbase
 _OSCL_Abort, 34
 ALLOC_AND_CONSTRUCT, 31
 ALLOCATE, 31
 big_endian_to_host, 34
 Bind, 34
 c_bool, 33
 CTIME_BUFFER_SIZE, 44
 CtimeStrBuf, 33
 host_to_big_endian, 34
 host_to_little_endian, 34
 int64, 33
 little_endian_to_host, 35

mbchar, 33
 MICROSECONDS, 34
 MILLISECONDS, 34
 MSEC_PER_SEC, 44
 NULL, 31
 NULL_TERM_CHAR, 31
 octet, 33
 operator-, 35
 operator==, 35
 OSCL_ABS, 31
 OSCL_ASSERT, 31
 OSCL Assert, 35
 oscl_CIstrcmp, 35, 36
 oscl_CIstrncmp, 36
 OSCL_COND_EXPORT_REF, 31
 OSCL_COND_IMPORT_REF, 31
 OSCL_CONST_CAST, 31
 OSCL_DISABLE_WARNING_-
 RETURN_TYPE_NOT_UDT, 31
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 31
 OSCL_DLL_ENTRY_POINT, 31
 OSCL_DLL_ENTRY_POINT_DEFAULT,
 32
 OSCL_DYNAMIC_CAST, 32
 OSCL_HAS_SINGLETON_SUPPORT, 32
 OSCL_INLINE, 32
 oscl_isLetter, 36
 OSCL_MAX, 32
 OSCL_MIN, 32
 OSCL_PACKED_VAR, 32
 OSCL_REINTERPRET_CAST, 32
 OSCL_STATIC_CAST, 32
 oscl_strcat, 37
 oscl strchr, 37, 38
 oscl strcmp, 38
 oscl strlen, 38
 oscl_strncat, 39
 oscl strncmp, 39, 40
 oscl_strncpy, 40
 oscl strrchr, 41
 oscl_strset, 41
 oscl strstr, 41, 42
 OSCL_TCHAR, 33
 OSCL_TEMPLATED_DESTRUCTOR_-
 CALL, 32
 OSCL_TLS_BASE_SLOTS, 32
 OSCL_TLS_EXTERNAL_SLOTS, 32
 OSCL_TLS_ID_BASE_LAST, 44
 OSCL_TLS_ID_ERRORHOOK, 44
 OSCL_TLS_ID_MAGICNUM, 44
 OSCL_TLS_ID_OSCLREGISTRY, 44
 OSCL_TLS_ID_PAYLOADPARSER, 44
 OSCL_TLS_ID_PVERRORTRAP, 44
 OSCL_TLS_ID_PVLOGGER, 44
 OSCL_TLS_ID_PVMFRECOGNIZER, 44
 OSCL_TLS_ID_PVSCHEDULER, 44
 OSCL_TLS_ID_SDPMEDIAPARSER, 44
 OSCL_TLS_ID_SQLITE3, 44
 OSCL_TLS_ID_TEST, 44
 OSCL_TLS_ID_WMDRM, 44
 OSCL_TLS_MAX_SLOTS, 32
 oscl_tolower, 42
 OSCL_UNSIGNED_CONST, 32
 OSCL_UNUSED_ARG, 32
 OSCL_UNUSED_RETURN, 33
 oscl_wchar, 33
 OsclAny, 33
 OsclFloat, 33
 PV8601TIME_BUFFER_SIZE, 44
 PV8601timeStrBuf, 33
 PV8601ToRFC822, 42
 PVMEM_INST_LEVEL, 33
 PVOscBase_Cleanup, 43
 PVOscBase_Init, 43
 RFC822ToPV8601, 43
 SECONDS, 34
 TimeUnits, 34
 TOscITlsKey, 33
 uint, 33
 uint64, 33
 unix_ntp_offset, 44
 USEC_PER_SEC, 44
 OsclBasicDateStruct
 osclconfig_time.h, 817
 OsclBasicTimeStruct
 osclconfig_time.h, 817
 OsclBind
 osclconfig_io.h, 797
 OsclBindMethod, 317
 OsclBindMethod
 ~OsclBindMethod, 317
 Bind, 317
 BindRequest, 317
 NewL, 317
 OsclBindRequest, 318
 OsclBindRequest, 318
 OsclBindRequest
 Bind, 318
 OsclBindRequest, 318
 OsclBinIStream, 319
 OsclBinIStream, 319
 OsclBinIStream
 ~OsclBinIStream, 319
 get, 319
 OsclBinIStream, 319
 Read_uint8, 319

OsclBinIStreamBigEndian, 321
 OsclBinIStreamBigEndian, 322
 OsclBinIStreamBigEndian
 operator>>, 322
 OsclBinIStreamBigEndian, 322
 Read, 322
 Read_uint16, 322
 Read_uint32, 322
 OsclBinIStreamLittleEndian, 324
 OsclBinIStreamLittleEndian, 325
 OsclBinIStreamLittleEndian
 operator>>, 325
 OsclBinIStreamLittleEndian, 325
 Read_uint16, 325
 Read_uint32, 325
 OsclBinOStream, 326
 OsclBinOStream, 326
 OsclBinOStream
 ~OsclBinOStream, 326
 OsclBinOStream, 326
 write, 326
 OsclBinOStreamBigEndian, 327
 OsclBinOStreamBigEndian, 328
 OsclBinOStreamBigEndian
 operator<<, 328
 OsclBinOStreamBigEndian, 328
 WriteUnsignedLong, 328
 WriteUnsignedShort, 328
 OsclBinOStreamLittleEndian, 329
 OsclBinOStreamLittleEndian, 330
 OsclBinOStreamLittleEndian
 operator<<, 330
 OsclBinOStreamLittleEndian, 330
 WriteUnsignedLong, 330
 WriteUnsignedShort, 330
 OsclBinStream, 331
 EOF_STATE, 332
 FAIL_STATE, 332
 GOOD_STATE, 332
 OsclBinStream, 332
 OsclBinStream
 Attach, 332
 eof, 332
 fail, 333
 firstFragPtr, 334
 fragsLeft, 334
 good, 333
 HaveRoomInCurrentBlock, 333
 length, 334
 nextFragPtr, 334
 numFrags, 334
 OsclBinStream, 332
 pBasePosition, 334
 PositionInBlock, 333
 pPosition, 334
 ReserveSpace, 333
 Seek, 333
 seekFromCurrentPosition, 333
 specialFragBuffer, 334
 state, 334
 state_t, 332
 tellg, 333
 OsclBuf, 335
 OsclBuf, 336
 OsclBuf
 Delete, 336
 Des, 336
 DesC, 336
 iBuffer, 336
 iLength, 336
 iMaxLength, 336
 Length, 336
 NewL, 336
 OsclBuf, 336
 OsclCloseSocket
 osclconfig_io.h, 798
 OsclCoeActiveScheduler
 OsclExecSchedulerBase, 386
 OsclExecSchedulerCommonBase, 392
 PVThreadContext, 620
 OsclCoeActiveSchedulerBase
 PVThreadContext, 620
 OsclCompareLess, 337
 OsclCompareLess
 compare, 337
 OsclComponentFactory
 osclutil, 67
 OsclComponentRegistry, 338
 OsclComponentRegistry, 339
 OsclComponentRegistry
 ~OsclComponentRegistry, 339
 CloseSession, 339
 FindExact, 339
 FindHierarchical, 339
 iComponentIdCounter, 339
 iData, 339
 iMutex, 339
 iNumSessions, 339
 OpenSession, 339
 OsclComponentRegistry, 339
 Register, 339
 Unregister, 339
 OsclComponentRegistryData, 340
 OsclComponentRegistryData
 Find, 340
 iVec, 340
 OsclComponentRegistryElement, 341
 OsclComponentRegistryElement, 341

OsclComponentRegistryElement
 ~OsclComponentRegistryElement, 341
 iComponentId, 341
 iFactory, 341
 iId, 341
 Match, 341
 operator=, 341
 OsclComponentRegistryElement, 341
 osclconfig
 __int16_check__, 23
 __int32_check__, 23
 __int8_check__, 23
 __uint16_check__, 23
 __uint32_check__, 23
 __uint8_check__, 23
 OSCL_ASSERT_ALWAYS, 21
 OSCL_BYTE_ORDER_BIG_ENDIAN,
 21
 OSCL_BYTE_ORDER_LITTLE_-
 ENDIAN, 21
 OSCL_HAS_BERKELEY_SOCKETS, 21
 OSCL_HAS_MSWIN_PARTIAL_-
 SUPPORT, 21
 OSCL_HAS_MSWIN_SUPPORT, 21
 OSCL_HAS_PRAGMA_PACK, 21
 OSCL_HAS_PTHREAD_SUPPORT, 21
 OSCL_HAS_PV_C_OS_API_-
 MEMORY_FUNCS, 22
 OSCL_HAS_PV_C_OS_SUPPORT, 22
 OSCL_HAS_PV_C_OS_TIME_FUNCS,
 22
 OSCL_HAS_SAVAJE_IO_SUPPORT, 22
 OSCL_HAS_SAVAJE_SUPPORT, 22
 OSCL_HAS_SEM_TIMEDWAIT_-
 SUPPORT, 22
 OSCL_HAS_SYMBIAN_-
 COMPATIBLE_IO_FUNCTION,
 22
 OSCL_HAS_SYMBIAN_DNS_SERVER,
 22
 OSCL_HAS_SYMBIAN_ERRORTRAP,
 22
 OSCL_HAS_SYMBIAN_MATH, 22
 OSCL_HAS_SYMBIAN_MEMORY_-
 FUNCS, 22
 OSCL_HAS_SYMBIAN_SCHEDULER,
 22
 OSCL_HAS_SYMBIAN_SOCKET_-
 SERVER, 22
 OSCL_HAS_SYMBIAN_SUPPORT, 22
 OSCL_HAS_SYMBIAN_TIMERS, 22
 OSCL_HAS_UNIX_SUPPORT, 22
 OSCL_HAS_UNIX_TIME_FUNCS, 22
 OSCL_INTEGERS_WORD_ALIGNED,
 22
 osclconfig.h, 786
 __TFS__, 787
 OSCL_HAS_ANDROID_FILE_IO_-
 SUPPORT, 787
 OSCL_HAS_ANDROID_SUPPORT, 787
 OSCL_NATIVE_UINT64_TYPE, 787
 OSCL_PACKED_STRUCT_BEGIN, 787
 OSCL_PACKED_STRUCT_END, 787
 OSCL_PACKED_VAR, 787
 OSCL_RELEASE_BUILD, 787
 OSCL_TEMPLATED_DESTRUCTOR_-
 CALL, 787
 OSCL_UNSIGNED_CONST, 787
 osclconfig_ansi_memory.h, 788
 OSCL_HAS_ANSI_MEMORY_FUNCS,
 788
 oscl_memsize_t, 788
 osclconfig_check.h, 789
 osclconfig_compiler_warnings.h, 790
 OSCL_FUNCTION_PTR, 790
 osclconfig_error.h, 791
 OSCL_HAS_ERRNO_H, 791
 OSCL_HAS_EXCEPTIONS, 791
 OSCL_HAS_SETJMP_H, 791
 OSCL_HAS_SYMBIAN_ERRORTRAP,
 791
 osclconfig_error_check.h, 792
 osclconfig_global_new_delete.h, 793
 osclconfig_global_placement_new.h, 794
 operator new, 794
 osclconfig_io.h, 795
 __FILE_OFFSET_BITS, 797
 OSCL_AF_INET, 797
 OSCL_FILE_BUFFER_MAX_SIZE, 797
 OSCL_HAS_ANSI_64BIT_FILE_IO_-
 SUPPORT, 797
 OSCL_HAS_ANSI_FILE_IO_SUPPORT,
 797
 OSCL_HAS_BERKELEY_SOCKETS,
 797
 OSCL_HAS_GLOB, 797
 OSCL_HAS_LARGE_FILE_SUPPORT,
 797
 OSCL_HAS_MSWIN_FILE_IO_-
 SUPPORT, 797
 OSCL_HAS_NATIVE_FILE_CACHE_-
 ENABLE, 797
 OSCL_HAS_PV_FILE_CACHE, 797
 OSCL_HAS_SOCKET_SUPPORT, 797
 OSCL_HAS_SYMBIAN_-
 COMPATIBLE_IO_FUNCTION,
 797

OSCL_HAS_SYMBIAN_DNS_SERVER,
 797
 OSCL_HAS_SYMBIAN_SOCKET_-
 SERVER, 797
 OSCL_IPPROTO_TCP, 797
 OSCL_IPPROTO_UDP, 797
 OSCL_SD_BOTH, 797
 OSCL_SD_RECEIVE, 797
 OSCL_SD_SEND, 797
 OSCL SOCK_DGRAM, 797
 OSCL SOCK_STREAM, 797
 OsclAccept, 797
 OsclBind, 797
 OsclCloseSocket, 798
 OsclConnect, 798
 OsclConnectComplete, 798
 OsclGetAsyncSockErr, 798
 OsclGetDottedAddr, 798
 OsclGethostbyname, 798
 OsclJoin, 799
 OsclListen, 799
 OsclMakeSockAddr, 799
 OsclRecv, 799
 OsclRecvFrom, 799
 OsclSend, 800
 OsclSendTo, 800
 OsclSetNonBlocking, 800
 OsclSetRecvBufferSize, 800
 OsclShutdown, 800
 OsclSocket, 800
 OsclSocketCleanup, 801
 OsclSocketSelect, 801
 OsclSocketStartup, 801
 OsclUnMakeSockAddr, 801
 OsclValidInetAddr, 801
 TOsclFileOffset, 801
 TOsclHostent, 801
 TOsclSockAddr, 801
 TOsclSockAddrLen, 801
 TOsclSocket, 801
 osclconfig_io_check.h, 802
 __verify__TOsclFileOffset__defined__,
 802
 osclconfig_ix86.h, 803
 osclconfig_lib.h, 804
 OSCL HAS_RUNTIME_LIB_-
 LOADING_SUPPORT, 804
 OSCL_LIB_READ_DEBUG_LIBS, 804
 PV_DYNAMIC_LOADING_CONFIG_-
 FILE_PATH, 804
 PV_RUNTIME_LIB_FILENAME_-
 EXTENSION, 804
 osclconfig_lib_check.h, 805
 osclconfig_limits_typedefs.h, 806
 OSCL_CHAR_IS_SIGNED, 806
 OSCL_CHAR_IS_UNSIGNED, 806
 osclconfig_memory.h, 807
 OSCL_BYPASS_MEMMGT, 807
 OSCL_HAS_GLOBAL_NEW_DELETE,
 807
 OSCL_HAS_HEAP_BASE_SUPPORT,
 807
 OSCL_HAS_SYMBIAN_MEMORY_-
 FUNCS, 807
 PVMEM_INST_LEVEL, 807
 osclconfig_memory_check.h, 808
 osclconfig_no_os.h, 809
 osclconfig_proc.h, 810
 osclconfig_proc_check.h, 811
 __verify__TOsclConditionObject__-
 defined__, 811
 __verify__TOsclMutexObject__defined__,
 811
 __verify__TOsclSemaphoreObject__-
 defined__, 811
 __verify__TOsclThreadFuncArg__-
 defined__, 811
 __verify__TOsclThreadFuncRet__-
 defined__, 811
 __verify__TOsclThreadId__defined__, 811
 __verify__TOsclThreadObject__defined__-,
 811
 osclconfig_proc_unix_android.h, 813
 OSCL HAS_NON_PREEMPTIVE_-
 THREAD_SUPPORT, 814
 OSCL HAS_PTHREAD_SUPPORT, 814
 OSCL HAS_SEM_TIMEDWAIT_-
 SUPPORT, 814
 OSCL HAS_SYMBIAN_SCHEDULER,
 814
 OSCL HAS_THREAD_SUPPORT, 814
 OSCL THREAD_DECL, 814
 TOsclConditionObject, 814
 TOsclMutexObject, 814
 TOsclSemaphoreObject, 814
 TOsclThreadFuncArg, 814
 TOsclThreadFuncRet, 814
 TOsclThreadId, 814
 TOsclThreadObject, 814
 osclconfig_proc_unix_common.h, 815
 OSCL HAS_NON_PREEMPTIVE_-
 THREAD_SUPPORT, 816
 OSCL HAS_PTHREAD_SUPPORT, 816
 OSCL HAS_SEM_TIMEDWAIT_-
 SUPPORT, 816
 OSCL HAS_SYMBIAN_SCHEDULER,
 816
 OSCL HAS_THREAD_SUPPORT, 816

OSCL_THREAD_DECL, 816
 TOsclConditionObject, 816
 TOsclMutexObject, 816
 TOsclSemaphoreObject, 816
 TOsclThreadFuncArg, 816
 TOsclThreadFuncRet, 816
 TOsclThreadId, 816
 TOsclThreadObject, 816
 osclconfig_time.h, 817
 OSCL_HAS_UNIX_TIME_FUNCS, 817
 OsclBasicDateStruct, 817
 OsclBasicTimeStruct, 817
 osclconfig_time_check.h, 818
 __Validate__BasicTimeDateStruct__, 818
 __Validate__BasicTimeStruct__, 818
 osclconfig_unix_android.h, 819
 __STRLIT, 822
 __STRLIT_CHAR, 822
 __STRLIT_WCHAR, 822
 INT64, 822
 INT64_HILO, 822
 OSCL_DISABLE_INLINES, 822
 OSCL_HAS_ANSI_MATH_SUPPORT,
 822
 OSCL_HAS_ANSI_STDIO_SUPPORT,
 822
 OSCL_HAS_ANSI_STDLIB_SUPPORT,
 822
 OSCL_HAS_ANSI_STRING_SUPPORT,
 822
 OSCL_HAS_ANSI_WIDE_STRING_-
 SUPPORT, 822
 OSCL_HAS_BASIC_LOCK, 822
 OSCL_HAS_GLOBAL_VARIABLE_-
 SUPPORT, 822
 OSCL_HAS_MSWIN_SUPPORT, 822
 OSCL_HAS_NATIVE_INT64_TYPE, 822
 OSCL_HAS_NATIVE_UINT64_TYPE,
 822
 OSCL_HAS_SYMBIAN_SUPPORT, 822
 OSCL_HAS_TLS_SUPPORT, 822
 OSCL_HAS_UNICODE_SUPPORT, 822
 OSCL_HAS_UNIX_SUPPORT, 822
 OSCL_MEMFRAG_PTR_BEFORE_LEN,
 822
 OSCL_NATIVE_INT64_TYPE, 822
 OSCL_NATIVE_UINT64_TYPE, 822
 OSCL_NATIVE_WCHAR_TYPE, 822
 OSCL_TLS_GET_FUNC, 822
 OSCL_TLS_IS_KEYED, 822
 OSCL_TLS_KEY_CREATE_FUNC, 822
 OSCL_TLS_KEY_DELETE_FUNC, 822
 OSCL_TLS_STORE_FUNC, 822
 TOsclBasicLockObject, 822

 TOsclTlsKey, 822
 UINT64, 822
 UINT64_HILO, 822
 osclconfig_unix_common.h, 823
 __STRLIT, 826
 __STRLIT_CHAR, 826
 __STRLIT_WCHAR, 826
 INT64, 826
 INT64_HILO, 826
 OSCL_DISABLE_INLINES, 826
 OSCL_HAS_ANSI_MATH_SUPPORT,
 826
 OSCL_HAS_ANSI_STDIO_SUPPORT,
 826
 OSCL_HAS_ANSI_STDLIB_SUPPORT,
 826
 OSCL_HAS_ANSI_STRING_SUPPORT,
 826
 OSCL_HAS_ANSI_WIDE_STRING_-
 SUPPORT, 826
 OSCL_HAS_BASIC_LOCK, 826
 OSCL_HAS_GLOBAL_VARIABLE_-
 SUPPORT, 826
 OSCL_HAS_MSWIN_SUPPORT, 826
 OSCL_HAS_NATIVE_INT64_TYPE, 826
 OSCL_HAS_NATIVE_UINT64_TYPE,
 826
 OSCL_HAS_SYMBIAN_SUPPORT, 826
 OSCL_HAS_TLS_SUPPORT, 826
 OSCL_HAS_UNICODE_SUPPORT, 826
 OSCL_HAS_UNIX_SUPPORT, 826
 OSCL_MEMFRAG_PTR_BEFORE_LEN,
 826
 OSCL_NATIVE_INT64_TYPE, 826
 OSCL_NATIVE_UINT64_TYPE, 826
 OSCL_NATIVE_WCHAR_TYPE, 826
 OSCL_TLS_GET_FUNC, 826
 OSCL_TLS_IS_KEYED, 826
 OSCL_TLS_KEY_CREATE_FUNC, 826
 OSCL_TLS_KEY_DELETE_FUNC, 826
 OSCL_TLS_STORE_FUNC, 826
 TOsclBasicLockObject, 826
 TOsclTlsKey, 826
 UINT64, 826
 UINT64_HILO, 826
 osclconfig_util.h, 827
 OSCL_CLOCK_HAS_DRIFT_-
 CORRECTION, 827
 OSCL_HAS_SYMBIAN_MATH, 827
 OSCL_HAS_SYMBIAN_TIMERS, 827
 OSCL RAND_MAX, 827
 SLEEP_ONE_SEC, 827
 osclconfig_util_check.h, 828
 OsclConnect

osclconfig_io.h, 798
OsclConnectComplete
 osclconfig_io.h, 798
OsclConnectMethod, 343
OsclConnectMethod
 ~OsclConnectMethod, 343
 Connect, 343
 ConnectRequest, 343
 NewL, 343
OsclConnectRequest, 344
 OsclConnectRequest, 344
 OsclSocketI, 528
OsclConnectRequest
 Connect, 344
 OsclConnectRequest, 344
OsclDestructDealloc, 345
OsclDestructDealloc
 destruct_and_dealloc, 345
OsclDNS, 346
 OsclSocketServ, 545
OsclDNS
 ~OsclDNS, 346
 CancelGetHostByName, 346
 GetHostByName, 347
 NewL, 347
 OsclDNSRequestAO, 347
OsclDNSI, 348
 OsclDNSRequestAO, 360
 OsclSocketServI, 547
OsclDNSI
 ~OsclDNSI, 348
 Close, 348
 DNSRequestParam, 349
 GetHostByName, 348
 GetHostByNameSuccess, 348
 NewL, 349
 Open, 349
 OsclDNSRequest, 349
OsclDNSIBase, 350
 OsclDNSIBase, 351
OsclDNSIBase
 ~OsclDNSIBase, 351
 CancelFxn, 351
 CancelGetHostByName, 351
 Close, 351
 GetHostByName, 351
 GetHostByNameSuccess, 351
 iAlloc, 351
 iSocketServ, 351
 IsReady, 351
 Open, 351
 OsclDNSIBase, 351
 OsclDNSRequest, 351
 OsclGetHostByNameRequest, 351

OsclDNSMethod, 353
OsclDNSMethod, 354
OsclDNSRequestAO, 360
OsclDNSMethod
 Abort, 354
 AbortAll, 354
 CancelMethod, 354
 ConstructL, 354
 iAlloc, 355
 iDNSFxn, 355
 iDNSObserver, 355
 iDNSRequestAO, 355
 iId, 355
 iLogger, 355
 MethodDone, 354
 OsclDNSMethod, 354
 Run, 354
 StartMethod, 354
OsclDNSObserver, 356
OsclDNSObserver
 ~OsclDNSObserver, 356
 HandleDNSEvent, 356
OsclDNSRequest, 357
 OsclDNSI, 349
 OsclDNSIBase, 351
 OsclDNSRequest, 357
 OsclDNSRequestAO, 360
OsclDNSRequest
 ~OsclDNSRequest, 357
 Activate, 357
 CancelRequest, 357
 Complete, 357
 iActive, 357
 iDNSRequestAO, 357
 iDNSRequestParam, 357
 OsclDNSRequest, 357
OsclDNSRequestAO, 358
 OsclDNS, 347
 OsclDNSRequestAO, 359
OsclDNSRequestAO
 Abort, 359
 ConstructL, 359
 DNSRequestParam, 360
 DoCancel, 359
 GetSocketError, 359
 iDNSI, 360
 iDNSMethod, 360
 iLogger, 360
 iSocketError, 360
 NewRequest, 359
 OsclDNSI, 360
 OsclDNSMethod, 360
 OsclDNSRequest, 360
 OsclDNSRequestAO, 359

RequestDone, 359
 Run, 359
 Serv, 359
 Success, 360
OsclDoubleLink, 361
 OsclDoubleLink, 361
OsclDoubleLink
 iNext, 361
 InsertAfter, 361
 InsertBefore, 361
 iPrev, 361
 OsclDoubleLink, 361
 Remove, 361
OsclDoubleList, 362
 OsclDoubleList, 362
OsclDoubleList
 Head, 362
 InsertHead, 362
 InsertTail, 362
 IsHead, 362
 IsTail, 362
 OsclDoubleList, 362
 Tail, 362
OsclDoubleListBase, 363
 OsclDoubleListBase, 364
OsclDoubleListBase
 getHead, 364
 getOffset, 364
 iHead, 364
 Insert, 364
 InsertHead, 364
 InsertTail, 364
 iOffset, 364
 IsEmpty, 364
 OsclDoubleListBase, 364
 Reset, 364
 SetOffset, 364
OsclDoubleRunner, 365
 OsclDoubleRunner, 365
OsclDoubleRunner
 iHead, 365
 iNext, 365
 iOffset, 365
 operator T *, 365
 operator++, 365
 operator-, 365
 OsclDoubleRunner, 365
 Set, 365
 SetToHead, 365
 SetToTail, 365
OsclErrAlreadyExists
 osclerror, 90
OsclErrAlreadyInstalled
 osclerror, 90

 OsclErrArgument
 osclerror, 90
OsclErrBadHandle
 osclerror, 90
OsclErrBusy
 osclerror, 90
OsclErrCancelled
 osclerror, 90
OsclErrCorrupt
 osclerror, 90
OsclErrGeneral
 osclerror, 90
OsclErrInvalidState
 osclerror, 90
OsclErrNoHandler
 osclerror, 90
OsclErrNoMemory
 osclerror, 90
OsclErrNone
 osclerror, 90
OsclErrNoResources
 osclerror, 90
OsclErrNotInstalled
 osclerror, 90
OsclErrNotReady
 osclerror, 90
OsclErrNotSupported
 osclerror, 90
OsclError, 367
 OsclErrorTrapImp, 373
 OsclExecSchedulerCommonBase, 392
 OsclTrapStack, 586
OsclError
 Leave, 367
 LeaveIfError, 367
 LeaveIfNull, 367
 Pop, 367
 PopDealloc, 367, 368
 PushL, 368
osclerror
 _PV_TRAP, 86
 _PV_TRAP_NO_TLS, 86
 internalLeave, 86
 OSCL_BAD_ALLOC_EXCEPTION_-
 CODE, 86
 OSCL_CATCH, 86
 OSCL_CATCH_ANY, 87
 OSCL_ERR_NONE, 87
 OSCL_FIRST_CATCH, 87
 OSCL_FIRST_CATCH_ANY, 87
 OSCL_GetLastError, 91
 OSCL_IsErrnoSupported, 91
 OSCL_JUMP_MAX_JUMP_MARKS, 87
 OSCL_LAST_CATCH, 87

OSCL_LEAVE, 88
 OSCL_MAX_TRAP_LEVELS, 88
 OSCL_SetLastError, 91
 OSCL_StrError, 91
 OSCL_TRAPSTACK_POP, 88
 OSCL_TRAPSTACK_POPDEALLOC, 88
 OSCL_TRAPSTACK_PUSH, 88
 OSCL_TRY, 88
 OSCL_TRY_NO_TLS, 88
 OsclErrAlreadyExists, 90
 OsclErrAlreadyInstalled, 90
 OsclErrArgument, 90
 OsclErrBadHandle, 90
 OsclErrBusy, 90
 OsclErrCancelled, 90
 OsclErrCorrupt, 90
 OsclErrGeneral, 90
 OsclErrInvalidState, 90
 OsclErrNoHandler, 90
 OsclErrNoMemory, 90
 OsclErrNone, 90
 OsclErrNoResources, 90
 OsclErrNotInstalled, 90
 OsclErrNotReady, 90
 OsclErrNotSupported, 90
 OsclErrOverflow, 90
 OsclErrSystemCallFailed, 90
 OsclErrThreadContextIncorrect, 90
 OsclErrTimeout, 90
 OsclErrUnderflow, 90
 OsclFailure, 90
 OsclLeaveCode, 91
 OsclPending, 90
 OsclReturnCode, 91
 OsclSuccess, 90
 OsclTrapOperation, 91
 PVERROR_DoLeave, 90
 PVERROR_IMP_JUMPS, 90
 PVERRORTRAP_REGISTRY, 90
 PVERRORTRAP_REGISTRY_ID, 91
 OsclErrorAllocator, 369
 OsclErrorAllocator, 369
 OsclErrorAllocator
 allocate, 369
 deallocate, 369
 operator delete, 370
 operator new, 370
 OsclErrorAllocator, 369
 OsclErrorTrap, 371
 OsclErrorTrapImp, 373
 OsclTrapStack, 586
 OsclErrorTrap
 Cleanup, 371
 GetErrorTrapImp, 371
 Init, 371
 OsclErrorTrapImp, 372
 OsclJump, 410
 OsclTrapStack, 586
 OsclErrorTrapImp
 CPVInterfaceProxy, 373
 iJumpData, 373
 iLeave, 373
 iTrapStack, 373
 OsclError, 373
 OsclErrorTrap, 373
 OsclExecScheduler, 373
 OsclExecSchedulerCommonBase, 373
 OsclJump, 373
 OsclJumpMark, 373
 OsclScheduler, 373
 OsclTrapStack, 373
 Trap, 372
 TrapNoTls, 372
 UnTrap, 372
 OsclErrOverflow
 osclerror, 90
 OsclErrSystemCallFailed
 osclerror, 90
 OsclErrThreadContextIncorrect
 osclerror, 90
 OsclErrTimeout
 osclerror, 90
 OsclErrUnderflow
 osclerror, 90
 OsclException, 374
 OsclException, 374
 OsclException
 getLeaveCode, 374
 OsclException, 374
 OsclExclusiveArrayPtr, 375
 OsclExclusiveArrayPtr, 376
 OsclExclusiveArrayPtr
 ~OsclExclusiveArrayPtr, 376
 _Ptr, 377
 get, 376
 operator *, 376
 operator->, 376
 operator=, 376
 OsclExclusiveArrayPtr, 376
 release, 377
 set, 377
 OsclExclusivePtr, 378
 OsclExclusivePtr, 379
 OsclExclusivePtr
 ~OsclExclusivePtr, 379
 _Ptr, 380
 get, 379
 operator *, 379

operator->, 379
 operator=, 379
 OsclExclusivePtr, 379
 release, 380
 set, 380
 OsclExclusivePtrA, 381
 OsclExclusivePtrA, 382
 OsclExclusivePtrA
 ~OsclExclusivePtrA, 382
 _Ptr, 383
 get, 382
 operator *, 382
 operator->, 382
 operator=, 382
 OsclExclusivePtrA, 382
 release, 383
 set, 383
 OsclExecScheduler, 384
 OsclErrorTrapImp, 373
 OsclExecSchedulerBase, 386
 OsclExecSchedulerCommonBase, 392
 PVActiveBase, 600
 PVActiveStats, 601
 PVThreadContext, 620
 OsclExecScheduler
 Current, 384
 OsclScheduler, 385
 RegisterForCallback, 384
 RunSchedulerNonBlocking, 384
 OsclExecSchedulerBase, 386
 PVThreadContext, 620
 OsclExecSchedulerBase
 OsclCoeActiveScheduler, 386
 OsclExecScheduler, 386
 PVActiveBase, 386
 OsclExecSchedulerCommonBase, 387
 EOtherExecStats_Last, 389
 EOtherExecStats_NativeOS, 389
 EOtherExecStats_QueueTime, 389
 EOtherExecStats_ReleaseTime, 389
 EOtherExecStats_WaitTime, 389
 OsclErrorTrapImp, 373
 OsclExecSchedulerCommonBase, 390
 PVActiveStats, 601
 PVThreadContext, 620
 OsclExecSchedulerCommonBase
 ~OsclExecSchedulerCommonBase, 390
 AddToExecTimerQ, 390
 BeginScheduling, 390
 BeginStats, 390
 BlockingLoopL, 390
 CallRunExec, 390
 CleanupExecQ, 390
 CleanupStatQ, 390
 ConstructL, 390
 ConstructStatQ, 390
 EndScheduling, 390
 EndStats, 390
 Error, 390
 FindPVBase, 390
 GetId, 390
 GetName, 390
 GetScheduler, 390
 iAlloc, 394
 iBlockingMode, 394
 iDebugLogger, 394
 iDefAlloc, 394
 iDelta, 394
 iDoStop, 394
 iDoSuspend, 394
 iErrorTrapImp, 394
 iExecTimerQ, 394
 iGrandTotalTicks, 394
 iLogger, 394
 iLogPerfIndentStr, 394
 iLogPerfIndentStrLen, 394
 iLogPerfTotal, 394
 iName, 394
 iNativeMode, 394
 IncLogPerf, 391
 InitExecQ, 391
 InstallScheduler, 391
 iNumAOAdded, 394
 iOtherExecStats, 394
 iPVStatQ, 394
 iPVStats, 394
 iReadyQ, 394
 iResumeSem, 394
 IsInstalled, 391
 IsStarted, 391
 iStopper, 394
 iStopperCrit, 394
 iSuspended, 394
 iThreadContext, 394
 iTime, 394
 iTimeCompareThreshold, 394
 iTotalPercent, 394
 iTotalTicksTemp, 394
 OsclActiveObject, 392
 OsclCoeActiveScheduler, 392
 OsclError, 392
 OsclExecScheduler, 392
 OsclExecSchedulerCommonBase, 390
 OsclReadyQ, 392
 OsclScheduler, 392
 OsclTimerCompare, 392
 OsclTimerObject, 394
 PendComplete, 391

PVActiveBase, 394
 PVActiveStats, 394
 PVSchedulerStopper, 394
 PVThreadContext, 394
 RequestCanceled, 391
 ResetLogPerf, 391
 ResumeScheduler, 391
 SetScheduler, 391
 ShowStats, 391
 ShowSummaryStats, 391
 StartNativeScheduler, 391
 StartScheduler, 391
 StopScheduler, 391
 SuspendScheduler, 392
 TOtherExecStats, 389
 UninstallScheduler, 392
 UpdateTimers, 392
 UpdateTimersMsec, 392
 WaitForReadyAO, 392

OsclFailure
 osclerror, 90

OsclFileCache, 396
 Oscl_File, 183
 OsclFileCache, 397

OsclFileCache
 ~OsclFileCache, 397
 _fixedCaches, 397
 _movableCache, 397
 AddFixedCache, 397
 Close, 397
 EndOfFile, 397
 FileSize, 397
 Flush, 397
 Open, 397
 OsclFileCache, 397
 OsclFileCacheBuffer, 397
 Read, 397
 Seek, 397
 Tell, 397
 Write, 397

OsclFileCacheBuffer, 398
 Oscl_File, 183
 OsclFileCache, 397
 OsclFileCacheBuffer, 399

OsclFileCacheBuffer
 capacity, 399
 Contains, 399
 currentPos, 399
 endPos, 399
 filePosition, 399
 FillFromFile, 399
 iContainer, 399
 isFixed, 399
 IsUpdated, 399

 OsclFileCacheBuffer, 399
 pBuffer, 399
 Preceeds, 399
 PrepRead, 399
 PrepWrite, 399
 SetPosition, 399
 updateEnd, 399
 updateStart, 399
 usableSize, 399
 WriteUpdatesToFile, 399

OsclFileHandle, 400
 OsclFileHandle, 400

OsclFileHandle
 Handle, 400
 Oscl_File, 400
 OsclFileHandle, 400

OsclFileStats, 401
 OsclFileStats, 401

OsclFileStats
 End, 401
 Log, 401
 LogAll, 401
 OsclFileStats, 401
 Start, 401

OsclFileStatsItem, 402

OsclFileStatsItem
 iOpCount, 402
 iParam, 402
 iParam2, 402
 iStartTick, 402
 iTotTicks, 402

OsclFloat
 osclbase, 33

OsclGetAsyncSockErr
 osclconfig_io.h, 798

OsclGetDottedAddr
 osclconfig_io.h, 798

OsclGethostbyname
 osclconfig_io.h, 798

OsclGetHostNameMethod, 403
 OsclGetHostNameRequest, 404

OsclGetHostNameMethod
 ~OsclGetHostNameMethod, 403
 GetHostName, 403
 NewL, 403

OsclGetHostNameRequest, 404
 OsclDNSIBase, 351

OsclGetHostNameRequest
 OsclGetHostNameMethod, 404

OsclInit, 405

OsclInit
 Cleanup, 405
 Init, 405

OsclInteger64Transport, 406

OsclInteger64Transport
 iHigh, [406](#)
 iLow, [406](#)
 osclo
 EOsclFileOp_Close, [96](#)
 EOsclFileOp_EndOfFile, [96](#)
 EOsclFileOp_Flush, [96](#)
 EOsclFileOp_Last, [97](#)
 EOsclFileOp_NativeClose, [96](#)
 EOsclFileOp_NativeEndOfFile, [97](#)
 EOsclFileOp_NativeFlush, [97](#)
 EOsclFileOp_NativeOpen, [96](#)
 EOsclFileOp_NativeRead, [96](#)
 EOsclFileOp_NativeSeek, [97](#)
 EOsclFileOp_NativeSize, [97](#)
 EOsclFileOp_NativeTell, [97](#)
 EOsclFileOp_NativeWrite, [96](#)
 EOsclFileOp_Open, [96](#)
 EOsclFileOp_Read, [96](#)
 EOsclFileOp_Seek, [96](#)
 EOsclFileOp_Size, [96](#)
 EOsclFileOp_Tell, [96](#)
 EOsclFileOp_Write, [96](#)
 EPVDNSCancel, [97](#)
 EPVDNSFailure, [97](#)
 EPVDNSGetHostByName, [97](#)
 EPVDNSPending, [97](#)
 EPVDNSSuccess, [97](#)
 EPVDNSTimeout, [97](#)
 oscl_chdir, [97](#)
 OSCL_FILE_CHAR_PATH_-
 DELIMITER, [95](#)
 OSCL_FILE_STATS_LOGGER_NODE,
 [95](#)
 OSCL_FILE_WCHAR_PATH_-
 DELIMITER, [95](#)
 OSCL_FILEMGMT_E_ALREADY_-
 EXISTS, [96](#)
 OSCL_FILEMGMT_E_NO_MATCH, [96](#)
 OSCL_FILEMGMT_E_NOT_EMPTY, [96](#)
 OSCL_FILEMGMT_E_NOT_-
 IMPLEMENTED, [96](#)
 OSCL_FILEMGMT_E_OK, [96](#)
 OSCL_FILEMGMT_E_PATH_NOT_-
 FOUND, [96](#)
 OSCL_FILEMGMT_E_PATH_TOO_-
 LONG, [96](#)
 OSCL_FILEMGMT_E_PERMISSION_-
 DENIED, [96](#)
 OSCL_FILEMGMT_E_SYS_SPECIFIC,
 [96](#)
 OSCL_FILEMGMT_E_UNKNOWN, [96](#)
 OSCL_FILEMGMT_ERR_TYPE, [96](#)
 OSCL_FILEMGMT_MODE_DIR, [96](#)
 OSCL_FILEMGMT_MODES, [96](#)
 OSCL_FILEMGMT_PERMS, [96](#)
 OSCL_FILEMGMT_PERMS_EXECUTE,
 [96](#)
 OSCL_FILEMGMT_PERMS_READ, [96](#)
 OSCL_FILEMGMT_PERMS_WRITE, [96](#)
 OSCL_FSSTAT, [95](#)
 oscl_getcwd, [97](#), [98](#)
 OSCL_IO_EXTENSION_MAXLEN, [95](#)
 OSCL_IO_FILENAME_MAXLEN, [95](#)
 oscl_mkdir, [98](#)
 oscl_rename, [98](#), [99](#)
 oscl_rmdir, [99](#)
 oscl_stat, [99](#), [100](#)
 OSCL_STAT_BUF, [95](#)
 oscl_statfs, [100](#)
 TOsclFileHandle, [95](#)
 TOsclFileOffsetInt32, [95](#)
 TOsclFileOp, [96](#)
 TPVDNSEvent, [97](#)
 TPVDNSFx, [97](#)
 OsclIPSocketI, [407](#)
 OsclIPSocketI, [408](#)
 OsclIPSocketI
 ~OsclIPSocketI, [408](#)
 Alloc, [408](#)
 Bind, [408](#)
 Close, [408](#)
 ConstructL, [408](#)
 GetRecvData, [408](#)
 GetSendData, [408](#)
 iAddress, [409](#)
 iAlloc, [409](#)
 iId, [409](#)
 iLogger, [409](#)
 iObserver, [409](#)
 iSocket, [409](#)
 iSocketServ, [409](#)
 Join, [408](#)
 OsclIPSocketI, [408](#)
 OsclSocketMethod, [409](#)
 OsclSocketRequestAO, [409](#)
 SetRecvBufferSize, [409](#)
 SocketServ, [409](#)
 OsclJoin
 osclconfig_io.h, [799](#)
 OsclJump, [410](#)
 OsclErrorTrapImp, [373](#)
 OsclJump
 ~OsclJump, [410](#)
 Jump, [410](#)
 OsclErrorTrapImp, [410](#)
 StaticJump, [410](#)
 Top, [410](#)

OsclJumpMark
 OsclErrorTrapImp, 373
 OsclLeaveCode
 osclerror, 91
 OsclListen
 osclconfig_io.h, 799
 OsclListenMethod, 411
 OsclListenMethod
 ~OsclListenMethod, 411
 Listen, 411
 ListenRequest, 411
 NewL, 411
 OsclListenRequest, 412
 OsclListenRequest, 412
 OsclListenRequest
 Listen, 412
 OsclListenRequest, 412
 OsclLockBase, 413
 OsclLockBase
 ~OsclLockBase, 413
 Lock, 413
 Unlock, 413
 OsclMakeSockAddr
 osclconfig_io.h, 799
 OsclMem, 414
 OsclMemGlobalAuditObject, 430
 OsclMem
 Cleanup, 414
 Init, 414
 OsclMemAllocator, 415
 OsclMemAllocator
 allocate, 415
 allocate_fl, 415
 deallocate, 415
 OsclMemAllocDestructDealloc, 416
 OsclMemAllocDestructDealloc
 allocate, 416
 allocate_fl, 416
 deallocate, 416
 destruct_and_dealloc, 416
 OsclMemAudit, 418
 OsclMemAudit, 418
 OsclMemAudit
 ~OsclMemAudit, 418
 GetLock, 419
 MM_AddTag, 419
 MM_allocate, 419
 MM_CreateAllocNodeInfo, 419
 MM_deallocate, 419
 MM_GetAllocNo, 419
 MM_GetAllocNodeInfo, 419
 MM_GetExistingTag, 420
 MM_GetMode, 420
 MM_GetNumAllocNodes, 420
 MM_GetOverheadStats, 420
 MM_GetPostfillPattern, 420
 MM_GetPrefillPattern, 420
 MM_GetRefCount, 420
 MM_GetRootNode, 421
 MM_GetStats, 421
 MM_GetStatsInDepth, 421
 MM_GetTagName, 421
 MM_GetTreeNodes, 421
 MM_ReleaseAllocNodeInfo, 421
 MM_SetFailurePoint, 421
 MM_SetMode, 422
 MM_SetPostfillPattern, 422
 MM_SetPrefillPattern, 422
 MM_SetTagLevel, 422
 MM_UnsetFailurePoint, 422
 MM_Validate, 422
 OsclMemAudit, 418
 OsclMemGlobalAuditObject, 423
 OSCLMemAutoPtr, 424
 OSCLMemAutoPtr, 425
 OSCLMemAutoPtr
 ~OSCLMemAutoPtr, 425
 _Ownership, 427
 allocate, 426
 deallocate, 426
 get, 426
 operator *, 426
 operator->, 426
 operator=, 426
 OSCLMemAutoPtr, 425
 release, 426
 setWithoutOwnership, 426
 takeOwnership, 427
 OsclMemBasicAllocator, 428
 OsclMemBasicAllocator
 allocate, 428
 deallocate, 428
 OsclMemBasicAllocDestructDealloc, 429
 OsclMemBasicAllocDestructDealloc
 allocate, 429
 deallocate, 429
 destruct_and_dealloc, 429
 OsclMemGlobalAuditObject, 430
 OsclMemAudit, 423
 OsclMemGlobalAuditObject
 audit_type, 430
 getGlobalMemAuditObject, 430
 OsclMem, 430
 OsclMemInit
 osclmemory, 60
 osclmemory
 _OSCL_CLEANUP_BASE_CLASS, 48
 _OSCL_TRAP_NEW, 48

_oscl_audit_calloc, 57
 _oscl_audit_free, 57
 _oscl_audit_malloc, 57
 _oscl_audit_new, 57
 _oscl_audit_realloc, 58
 _oscl_calloc, 58
 _oscl_default_audit_calloc, 58
 _oscl_default_audit_malloc, 58
 _oscl_default_audit_new, 58
 _oscl_default_audit_realloc, 58
 _oscl_free, 58
 _oscl_malloc, 58
 _oscl_realloc, 58
 ALLOC_NODE_FLAG, 60
 COMPUTE_MEM_ALIGN_SIZE, 49
 DEFAULT_MM_AUDIT_MODE, 50
 DEFAULT_POSTFILL_PATTERN, 50
 DEFAULT_PREFILL_PATTERN, 50
 FENCE_PATTERN, 50
 MEM_ALIGN_SIZE, 50
 MIN_FENCE_SIZE, 50
 MM_ALLOC_MAX_QUERY_-
 FILENAME_LEN, 50
 MM_ALLOC_MAX_QUERY_TAG_LEN,
 50
 MM_AllocNodeAutoPtr, 57
 MM_AUDIT_ALLOC_NODE_-
 ENABLE_FLAG, 50
 MM_AUDIT_ALLOC_NODE_-
 SUPPORT, 50
 MM_AUDIT_FAILURE_SIMULATION_-
 SUPPORT, 50
 MM_AUDIT_FENCE_SUPPORT, 50
 MM_AUDIT_FILL_SUPPORT, 50
 MM_AUDIT_INCLUDE_ALL_HEAP_-
 VALIDATION, 50
 MM_AUDIT_POSTFILL_FLAG, 50
 MM_AUDIT_PREFILL_FLAG, 50
 MM_AUDIT_SUPPRESS_FILENAME_-
 FLAG, 50
 MM_AUDIT_VALIDATE_ALL_HEAP_-
 FLAG, 50
 MM_AUDIT_VALIDATE_BLOCK, 50
 MM_AUDIT_VALIDATE_ON_FREE_-
 FLAG, 50
 MM_StatsNodeTagTreeType, 57
 MMAuditCharAutoPtr, 57
 MMAuditUint8AutoPtr, 57
 operator delete, 58
 operator delete[], 58
 operator new, 58
 operator new[], 58
 OSCL_ALLOC_DELETE, 50
 OSCL_ALLOC_NEW, 51
 OSCL_ARRAY_DELETE, 51
 OSCL_ARRAY_NEW, 51
 OSCL_AUDIT_ARRAY_NEW, 51
 OSCL_AUDIT_CALLOC, 52
 OSCL_AUDIT_MALLOC, 52
 OSCL_AUDIT_NEW, 52
 OSCL_AUDIT_REALLOC, 53
 OSCL_CALLOC, 53
 oscl_calloc, 53
 OSCL_CLEANUP_BASE_CLASS, 53
 OSCL_DEFAULT_FREE, 54
 OSCL_DEFAULT_MALLOC, 54
 OSCL_DELETE, 54
 OSCL_DISABLE_WARNING_-
 RETURN_TYPE_NOT_UDT, 54
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 54
 OSCL_FREE, 54
 oscl_free, 54
 OSCL_HAS_GLOBAL_NEW_DELETE,
 54
 OSCL_MALLOC, 55
 oscl_malloc, 55
 oscl_mem_aligned_size, 58
 oscl_memcmp, 59
 oscl_memcpy, 59
 oscl_memmove, 59
 oscl_memmove32, 59
 oscl_memset, 60
 OSCL_NEW, 55
 OSCL_PLACEMENT_NEW, 55
 OSCL_REALLOC, 55
 oscl_realloc, 55
 OSCL_TRAP_ALLOC_NEW, 55
 OSCL_TRAP_AUDIT_NEW, 56
 OSCL_TRAP_NEW, 56
 OsclMemInit, 60
 OsclMemStatsNodeAutoPtr, 57
 OsclTagTreeType, 57
 TagTreeAllocator, 57
 OsclMemoryFragment, 431
 OsclMemoryFragment
 len, 431
 ptr, 431
 OsclMemPoolAllocator, 432
 OsclMemPoolAllocator, 432
 OsclMemPoolAllocator
 ~OsclMemPoolAllocator, 432
 CreateMemPool, 432
 DestroyMemPool, 432
 oscl_mem_aligned_size, 432
 OsclMemPoolAllocator, 432
 OsclMemPoolFixedChunkAllocator, 433

OsclMemPoolFixedChunkAllocator, 434
 OsclMemPoolFixedChunkAllocator
 ~OsclMemPoolFixedChunkAllocator, 434
 addRef, 434
 allocate, 434
 CancelFreeChunkAvailableCallback, 434
 createmempool, 434
 deallocate, 435
 destroymempool, 435
 enablenullpointerreturn, 435
 iCheckNextAvailableFreeChunk, 436
 iChunkAlignment, 436
 iChunkSize, 436
 iChunkSizeMemAligned, 436
 iEnableNullPtrReturn, 436
 iFreeMemChunkList, 436
 iMemPool, 436
 iMemPoolAligned, 436
 iMemPoolAllocator, 436
 iNextAvailableContextData, 436
 iNumChunk, 436
 iObserver, 436
 iRefCount, 436
 notifyfreechunkavailable, 435
 OsclMemPoolFixedChunkAllocator, 434
 removeRef, 435
 OsclMemPoolFixedChunkAllocatorObserver,
 437
 OsclMemPoolFixedChunkAllocatorObserver
 ~OsclMemPoolFixedChunkAllocatorObserver,
 437
 freechunkavailable, 437
 OsclMemPoolResizableAllocator, 438
 OsclMemPoolResizableAllocator, 439
 OsclMemPoolResizableAllocator
 ~OsclMemPoolResizableAllocator, 439
 addnewmempoolbuffer, 439
 addRef, 439
 allocate, 440
 allocateblock, 440
 CancelFreeChunkAvailableCallback, 440
 CancelFreeMemoryAvailableCallback, 440
 deallocate, 440
 deallocateblock, 440
 destroyallmempoolbuffers, 440
 enablenullpointerreturn, 440
 findfreeblock, 441
 getAllocatedSize, 441
 getAvailableSize, 441
 getBufferSize, 441
 getLargestContiguousFreeBlockSize, 441
 getMemPoolBufferAllocatedSize, 441
 getMemPoolBufferSize, 441
 iBlockInfoAlignedSize, 443
 iBufferInfoAlignedSize, 443
 iCheckFreeMemoryAvailable, 443
 iCheckNextAvailable, 443
 iEnableNullPtrReturn, 443
 iExpectedNumBlocksPerBuffer, 443
 iFreeMemContextData, 443
 iFreeMemPoolObserver, 443
 iMaxNewMemPoolBufferSz, 443
 iMemPoolBufferAllocator, 443
 iMemPoolBufferList, 443
 iMemPoolBufferNumLimit, 443
 iMemPoolBufferSize, 443
 iNextAvailableContextData, 443
 iObserver, 443
 iRefCount, 443
 iRequestedAvailableFreeMemSize, 443
 iRequestedNextAvailableSize, 443
 memoryPoolBufferMgmtOverhead, 441
 notifyfreeblockavailable, 441
 notifyfreememoryavailable, 441
 OsclMemPoolResizableAllocator, 439
 removeRef, 442
 setMaxSzForNewMemPoolBuffer, 442
 trim, 442
 validateblock, 442
 OsclMemPoolResizableAllocator::MemPoolBlockInfo,
 444
 OsclMemPoolResizableAllocator::MemPool-
 BlockInfo
 iBlockBuffer, 444
 iBlockPostFence, 444
 iBlockPreFence, 444
 iBlockSize, 444
 iNextFreeBlock, 444
 iParentBuffer, 444
 iPrevFreeBlock, 444
 OsclMemPoolResizableAllocator::MemPoolBufferInfo,
 445
 OsclMemPoolResizableAllocator::MemPool-
 BufferInfo
 iAllocatedSz, 445
 iBufferPostFence, 445
 iBufferPreFence, 445
 iBufferSize, 445
 iEndAddr, 445
 iNextFreeBlock, 445
 iNumOutstanding, 445
 iStartAddr, 445
 OsclMemPoolResizableAllocatorMemoryObserver,
 446
 OsclMemPoolResizableAllocatorMemory-
 Observer
 ~OsclMemPoolResizableAllocatorMemoryObserver,
 446

freememoryavailable, 446
OsclMemPoolResizableAllocatorObserver, 447
OsclMemPoolResizableAllocatorObserver
 ~OsclMemPoolResizableAllocatorObserver,
 447
 freeblockavailable, 447
OsclMemStatsNode, 448
 OsclMemStatsNode, 448
OsclMemStatsNode
 ~OsclMemStatsNode, 448
 operator delete, 448
 operator new, 448
 OsclMemStatsNode, 448
 pMMFIParam, 448
 pMMStats, 448
 reset, 448
 tag, 448
OsclMemStatsNodeAutoPtr
 osclmemory, 57
OsclMutex, 449
 OsclMutex, 449
OsclMutex
 ~OsclMutex, 449
 Close, 449
 Create, 449
 Lock, 450
 OsclMutex, 449
 TryLock, 450
 Unlock, 450
OsclNameString, 451
 OsclNameString, 451
OsclNameString
 MaxLen, 451
 OsclNameString, 451
 Set, 451
 Str, 451
OsclNativeFile, 452
 Oscl_FileServer, 191
 OsclNativeFile, 453
OsclNativeFile
 ~OsclNativeFile, 453
 Close, 453
 EndOfFile, 453
 Flush, 453
 GetError, 453
 GetReadAsyncNumElements, 453
 HasAsyncRead, 453
 Mode, 453
 Open, 453
 OsclNativeFile, 453
 Read, 453
 ReadAsync, 453
 ReadAsyncCancel, 453
 Seek, 454
 Size, 454
 Tell, 454
 Write, 454
OsclNativeFileParams, 455
 OsclNativeFileParams, 455
OsclNativeFileParams
 iAsyncReadBufferSize, 455
 iNativeAccessMode, 455
 iNativeBufferSize, 455
 OsclNativeFileParams, 455
OsclNetworkAddress, 456
 OsclNetworkAddress, 456
OsclNetworkAddress
 ipAddr, 456
 operator==, 456
 OsclNetworkAddress, 456
 port, 456
OsclNoYieldMutex
 oscl_mutex.h, 708
OsclNullLock, 457
OsclNullLock
 ~OsclNullLock, 457
 Lock, 457
 Unlock, 457
OsclPending
 osclerror, 90
OsclPriorityLink, 458
OsclPriorityLink
 iPriority, 458
OsclPriorityList, 459
 OsclPriorityList, 459
OsclPriorityList
 Head, 459
 Insert, 459
 IsHead, 459
 IsTail, 459
 OsclPriorityList, 459
 Tail, 459
OsclPriorityQueue, 460
 OsclPriorityQueue, 461
OsclPriorityQueue
 ~OsclPriorityQueue, 461
 c, 463
 comp, 463
 compare_EQ, 461
 compare_LT, 461
 const_reference, 461
 container_type, 461
 empty, 462
 find_heap, 462
 iterator, 461
 oscl_priqueue_test, 463
 OsclPriorityQueue, 461
 pop, 462

pop_heap, 462
 push, 462
 push_heap, 462
 remove, 462
 reserve, 462
 size, 462
 swap, 462
 top, 462
 validate, 463
 value_type, 461
 vec, 463
OsclPriorityQueueBase, 464
 Oscl_Vector_Base, 288
OsclPriorityQueueBase
 ~OsclPriorityQueueBase, 464
 construct, 464
 find_heap, 464
 pop_heap, 464
 push_heap, 464
 remove, 464
osclproc
 EPVThreadContext_InThread, 104
 EPVThreadContext_NonOsclThread, 104
 EPVThreadContext_OsclThread, 104
 EPVThreadContext_Undetermined, 104
 OSCL_PERF_SUMMARY_LOGGING, 103
 OSCL_REQUEST_ERR_CANCEL, 104
 OSCL_REQUEST_ERR_GENERAL, 104
 OSCL_REQUEST_ERR_NONE, 104
 OSCL_REQUEST_PENDING, 104
 OSCL_ZEROIZE, 103
 OsclPtrAdd, 104
 OsclPtrSub, 104
 PV_SCHED_CHECK_Q, 103
 PV_SCHED_ENABLE_AO_STATS, 103
 PV_SCHED_ENABLE_LOOP_STATS, 103
 PV_SCHED_ENABLE_PERF_LOGGING, 103
 PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS, 103
 PV_SCHED_FAIR_SCHEDULING, 103
 PV_SCHED_LOG_Q, 103
 PVEEXECNAMELEN, 103
 PVSCHEDNAMELEN, 103
 QUE_ITER_BEGIN, 103
 QUE_ITER_END, 103
 TOsclReady, 104
 TPVThreadContext, 104
OsclProcStatus, 465
 ALREADY_SUSPENDED_ERROR, 465
 BAD_THREADID_ADDR_ERROR, 465
EXCEED_MAX_COUNT_VARIABLE_ERROR, 466
EXCEED_MAX_SEM_COUNT_ERROR, 466
INVALID_ACCESS_ERROR, 466
INVALID_ARGUMENT_ERROR, 466
INVALID_FUNCTION_ERROR, 466
INVALID_HANDLE_ERROR, 466
INVALID_OPERATION_ERROR, 466
INVALID_PARAM_ERROR, 465
INVALID_POINTER_ERROR, 466
INVALID_PRIORITY_ERROR, 465
INVALID_THREAD_ERROR, 465
INVALID_THREAD_ID_ERROR, 465
MAX_THRDS_REACHED_ERROR, 465
MUTEX_LOCKED_ERROR, 466
NO_PERMISSION_ERROR, 465
NOT_ENOUGH_MEMORY_ERROR, 465
NOT_ENOUGH_RESOURCES_ERROR, 465
NOT_IMPLEMENTED, 466
NOT_SUSPENDED_ERROR, 465
OTHER_ERROR, 465
OUTOFMEMORY_ERROR, 465
PSHARED_ATTRIBUTE_SETTING_ERROR, 466
PSHARED_NOT_ZERO_ERROR, 466
RELOCK_MUTEX_ERROR, 466
SEM_NOT_SIGNALLED_ERROR, 466
SUCCESS_ERROR, 465
SYSTEM_RESOURCES_UNAVAILABLE_ERROR, 466
THREAD_1_INACTIVE_ERROR, 465
THREAD_BLOCK_ERROR, 466
THREAD_NOT_OWN_MUTEX_ERROR, 466
TOO_MANY_THREADS_ERROR, 465
WAIT_ABANDONED_ERROR, 466
WAIT_TIMEOUT_ERROR, 466
OsclProcStatus
 eOsclProcError, 465
OsclPtr, 467
 OsclPtr, 467
OsclPtr
 Append, 467
 Length, 467
 OsclPtr, 467
 Ptr, 467
 Set, 467
 SetLength, 467
 Zero, 467
OsclPtrAdd
 osclproc, 104
OsclPtrC, 469

OsclPtrC, 470
 OsclPtrC
 Left, 470
 Length, 470
 OsclPtrC, 470
 Ptr, 470
 Right, 470
 Set, 470
 SetLength, 470
 Zero, 470
 OsclPtrSub
 osclproc, 104
 OsclRand, 471
 OsclRand
 Rand, 471
 Seed, 471
 OsclReadyAlloc, 472
 OsclReadyAlloc
 allocate, 472
 allocate_fl, 472
 deallocate, 472
 OsclReadyCompare, 473
 PVActiveBase, 600
 OsclReadyCompare
 compare, 473
 OsclReadyQ, 474
 OsclExecSchedulerCommonBase, 392
 PVActiveBase, 600
 PVActiveStats, 601
 OsclReadyQ
 Callback, 475
 Construct, 475
 Depth, 475
 IsIn, 475
 PendComplete, 475
 PopTop, 475
 RegisterForCallback, 475
 Remove, 475
 ThreadLogoff, 475
 ThreadLogon, 475
 TimerCallback, 475
 Top, 475
 WaitAndPopTop, 475
 WaitForRequestComplete, 475
 OsclReadySetPosition
 PVActiveBase, 600
 OsclRecv
 osclconfig_io.h, 799
 OsclRecvFrom
 osclconfig_io.h, 799
 OsclRecvFromMethod, 476
 OsclRecvFromMethod
 ~OsclRecvFromMethod, 476
 GetRecvData, 476
 NewL, 476
 RecvFrom, 476
 RecvFromRequest, 476
 OsclRecvFromRequest, 478
 OsclRecvFromRequest, 478
 OsclSocketI, 528
 OsclRecvFromRequest
 GetRecvData, 478
 OsclRecvFromRequest, 478
 RecvFrom, 478
 Success, 478
 OsclRecvMethod, 480
 OsclRecvMethod
 ~OsclRecvMethod, 480
 GetRecvData, 480
 NewL, 480
 Recv, 480
 RecvRequest, 480
 OsclRecvRequest, 481
 OsclRecvRequest, 481
 OsclSocketI, 528
 OsclRecvRequest
 GetRecvData, 481
 OsclRecvRequest, 481
 Recv, 481
 Success, 481
 OsclRefCounter, 482
 OsclRefCounter
 ~OsclRefCounter, 482
 addRef, 482
 getCount, 482
 removeRef, 482
 OsclRefCounterDA, 484
 OsclRefCounterDA, 484
 OsclRefCounterDA
 ~OsclRefCounterDA, 484
 addRef, 485
 getCount, 485
 OsclRefCounterDA, 484
 removeRef, 485
 OsclRefCounterMemFrag, 486
 OsclRefCounterMemFrag, 486
 OsclRefCounterMemFrag
 ~OsclRefCounterMemFrag, 486
 getCapacity, 487
 getCount, 487
 getMemFrag, 487
 getMemFragPtr, 487
 getMemFragSize, 487
 getRefCounter, 487
 operator=, 487
 OsclRefCounterMemFrag, 486
 OsclRefCounterMTDA, 488
 OsclRefCounterMTDA, 488

OsclRefCounterMTDA
 ~OsclRefCounterMTDA, 488
 addRef, 489
 getCount, 489
 OsclRefCounterMTDA, 488
 removeRef, 489
OsclRefCounterMTSA, 490
 OsclRefCounterMTSA, 490
OsclRefCounterMTSA
 ~OsclRefCounterMTSA, 490
 addRef, 491
 getCount, 491
 OsclRefCounterMTSA, 490
 removeRef, 491
OsclRefCounterSA, 492
 OsclRefCounterSA, 492
OsclRefCounterSA
 ~OsclRefCounterSA, 492
 addRef, 493
 getCount, 493
 OsclRefCounterSA, 492
 removeRef, 493
OsclRegistryAccessClient, 494
 OsclRegistryAccessClient, 494
 OsclRegistryClientImpl, 502
 OsclRegistryServTlsImpl, 505
OsclRegistryAccessClient
 ~OsclRegistryAccessClient, 494
 Close, 494
 Connect, 494
 GetFactories, 494
 GetFactory, 494
 OsclRegistryAccessClient, 494
OsclRegistryAccessClientImpl, 496
OsclRegistryAccessClientTlsImpl, 497
OsclRegistryAccessElement, 498
OsclRegistryAccessElement
 iFactory, 498
 iMimeType, 498
OsclRegistryClient, 499
 OsclRegistryClient, 499
 OsclRegistryClientImpl, 502
 OsclRegistryServTlsImpl, 505
OsclRegistryClient
 ~OsclRegistryClient, 499
 Close, 499
 Connect, 499
 OsclRegistryClient, 499
 Register, 499
 UnRegister, 500
OsclRegistryClientImpl, 501
OsclRegistryClientImpl
 Close, 502
 Connect, 502
 GetFactories, 502
 GetFactory, 502
 OsclRegistryAccessClient, 502
 OsclRegistryClient, 502
 Register, 502
 UnRegister, 502
OsclRegistryClientTlsImpl, 503
OsclRegistryServTlsImpl, 504
 OsclRegistryServTlsImpl, 505
OsclRegistryServTlsImpl
 ~OsclRegistryServTlsImpl, 505
 Close, 505
 Connect, 505
 GetFactories, 505
 GetFactory, 505
 OsclRegistryAccessClient, 505
 OsclRegistryClient, 505
 OsclRegistryServTlsImpl, 505
 Register, 505
 UnRegister, 505
OsclReturnCode
 osclerror, 91
OsclScheduler, 506
 OsclErrorTrapImpl, 373
 OsclExecScheduler, 385
 OsclExecSchedulerCommonBase, 392
OsclScheduler
 Cleanup, 506
 Init, 506
OsclSchedulerCommonBase
 PVActiveBase, 600
OsclSchedulerObserver, 507
OsclSchedulerObserver
 ~OsclSchedulerObserver, 507
 OsclSchedulerReadyCallback, 507
 OsclSchedulerTimerCallback, 507
OsclSchedulerReadyCallback
 OsclSchedulerObserver, 507
OsclSchedulerTimerCallback
 OsclSchedulerObserver, 507
OsclScopedLock, 508
 OsclScopedLock, 508
OsclScopedLock
 ~OsclScopedLock, 508
 OsclScopedLock, 508
OsclSelect, 509
 OsclSelect, 510
OsclSelect
 iErrAlloc, 510
 iHeapCheck, 510
 iOsclBase, 510
 iOsclErrorTrap, 510
 iOsclLogger, 510
 iOsclMemory, 510

iOsclScheduler, 510
 iOutputFile, 510
 iSchedulerAlloc, 510
 iSchedulerName, 510
 iSchedulerReserve, 510
 OsclSelect, 510
OsclSemaphore, 511
 OsclSemaphore, 511
OsclSemaphore
 ~OsclSemaphore, 511
 Close, 511
 Create, 511
 OsclSemaphore, 511
 Signal, 512
 TryWait, 512
 Wait, 512
OsclSend
 osclconfig_io.h, 800
OsclSendMethod, 513
OsclSendMethod
 ~OsclSendMethod, 513
 GetSendData, 513
 NewL, 513
 Send, 513
 SendRequest, 513
OsclSendRequest, 514
 OsclSendRequest, 514
 OsclSocketI, 528
OsclSendRequest
 GetSendData, 514
 OsclSendRequest, 514
 Send, 514
 Success, 514
OsclSendTo
 osclconfig_io.h, 800
OsclSendToMethod, 515
OsclSendToMethod
 ~OsclSendToMethod, 515
 GetSendData, 515
 NewL, 515
 SendTo, 515
 SendToRequest, 515
OsclSendToRequest, 516
 OsclSendToRequest, 516
 OsclSocketI, 528
OsclSendToRequest
 GetSendData, 516
 OsclSendToRequest, 516
 SendTo, 516
 Success, 516
OsclSetNonBlocking
 osclconfig_io.h, 800
OsclSetRecvBufferSize
 osclconfig_io.h, 800
OsclSharedPtr, 517
 OsclSharedPtr, 518
OsclSharedPtr
 ~OsclSharedPtr, 518
 get_count, 518
 GetRefCounter, 518
 GetRep, 518
 operator *, 518
 operator TheClass *, 519
 operator->, 519
 operator=, 519
 OsclSharedPtr, 518
 Unbind, 519
OsclShutdown
 osclconfig_io.h, 800
OsclShutdownMethod, 520
OsclShutdownMethod
 ~OsclShutdownMethod, 520
 NewL, 520
 Shutdown, 520
 ShutdownRequest, 520
OsclShutdownRequest, 521
 OsclShutdownRequest, 521
 OsclSocketI, 528
OsclShutdownRequest
 OsclShutdownRequest, 521
 Shutdown, 521
OsclSingleton, 522
 OsclSingleton, 522
OsclSingleton
 ~OsclSingleton, 522
 _Ptr, 523
 operator *, 522
 operator->, 522
 OsclSingleton, 522
 set, 522
OsclSingletonRegistry, 524
OsclSingletonRegistry
 getInstance, 524
 lockAndGetInstance, 524
 OsclBase, 524
 registerInstance, 524
 registerInstanceAndUnlock, 524
OsclSocket
 osclconfig_io.h, 800
OsclSocketCleanup
 osclconfig_io.h, 801
OsclSocketI, 525
 OsclSocketRequestAO, 543
 OsclSocketServI, 547
OsclSocketI
 ~OsclSocketI, 526
 Accept, 526
 Bind, 526

Close, 526
 Connect, 526
 Join, 526
 Listen, 526
 Logger, 526
 MakeAddr, 527
 NewL, 527
 Open, 527
 OsclAcceptRequest, 528
 OsclConnectRequest, 528
 OsclRecvFromRequest, 528
 OsclRecvRequest, 528
 OsclSendRequest, 528
 OsclSendToRequest, 528
 OsclShutdownRequest, 528
 OsclTCPSocket, 528
 OsclUDPSocket, 528
 ProcessAccept, 527
 ProcessConnect, 527
 ProcessRecv, 527
 ProcessRecvFrom, 527
 ProcessSend, 527
 ProcessSendTo, 527
 ProcessShutdown, 527
 Recv, 527
 RecvFrom, 527
 RecvFromSuccess, 527
 RecvSuccess, 527
 Send, 527
 SendSuccess, 528
 SendTo, 528
 SendToSuccess, 528
 SetRecvBufferSize, 528
 Shutdown, 528
 Socket, 528
 OsclSocketIBase, 530
 OsclSocketIBase, 531
 OsclSocketIBase
 ~OsclSocketIBase, 531
 Accept, 531
 Bind, 531
 BindAsync, 531
 CancelAccept, 532
 CancelBind, 532
 CancelConnect, 532
 CancelFxn, 532
 CancelListen, 532
 CancelRecv, 532
 CancelRecvFrom, 532
 CancelSend, 532
 CancelSendTo, 532
 CancelShutdown, 532
 Close, 532
 Connect, 532
 GetShutdown, 532
 HasAsyncBind, 532
 HasAsyncListen, 532
 iAlloc, 534
 iSocketServ, 534
 IsOpen, 532
 Join, 532
 Listen, 532
 ListenAsync, 532
 Open, 533
 OsclSocketIBase, 531
 OsclSocketMethod, 534
 OsclSocketRequest, 534
 OsclSocketRequestAO, 534
 OsclTCPSocket, 534
 OsclUDPSocket, 534
 Recv, 533
 RecvFrom, 533
 RecvFromSuccess, 533
 RecvSuccess, 533
 Send, 533
 SendSuccess, 533
 SendTo, 533
 SendToSuccess, 533
 Shutdown, 534
 OsclSocketMethod, 535
 OsclIPSocketI, 409
 OsclSocketIBase, 534
 OsclSocketMethod, 536
 OsclSocketRequestAO, 543
 OsclSocketMethod
 ~OsclSocketMethod, 536
 Abort, 536
 AbortAll, 536
 Alloc, 536
 CancelMethod, 536
 ConstructL, 536
 iContainer, 537
 iSocketFxn, 537
 iSocketRequestAO, 537
 MethodDone, 536
 OsclSocketMethod, 536
 Run, 536
 StartMethod, 537
 OsclSocketObserver, 538
 OsclSocketObserver
 ~OsclSocketObserver, 538
 HandleSocketEvent, 538
 OsclSocketRequest, 539
 OsclSocketIBase, 534
 OsclSocketRequest, 539
 OsclSocketRequestAO, 543
 OsclSocketServI, 547
 OsclSocketRequest

Activate, 539
 CancelRequest, 539
 Complete, 539
 Fxn, 539
 iParam, 539
 iSocketI, 539
 iSocketRequestAO, 539
 OsclSocketRequest, 539
 OsclSocketRequestAO, 540
 OsclIPSocketI, 409
 OsclSocketIBase, 534
 OsclSocketRequestAO, 541
 OsclSocketRequestAO
 ~OsclSocketRequestAO, 541
 Abort, 541
 Alloc, 541
 CleanupParam, 541
 ConstructL, 541
 DoCancel, 541
 GetSocketError, 541
 iContainer, 543
 Id, 542
 iParam, 543
 iParamSize, 543
 iSocketError, 543
 NewRequest, 542
 OsclSocketI, 543
 OsclSocketMethod, 543
 OsclSocketRequest, 543
 OsclSocketRequestAO, 541
 RequestDone, 542
 Run, 542
 SocketI, 542
 SocketObserver, 542
 Success, 542
 OsclSocketSelect
 osclconfig_io.h, 801
 OsclSocketServ, 544
 OsclSocketServI, 547
 OsclSocketServ
 ~OsclSocketServ, 544
 Close, 544
 Connect, 544
 NewL, 545
 OsclDNS, 545
 OsclTCPSocket, 545
 OsclUDPSocket, 545
 OsclSocketServI, 546
 OsclSocketServRequestList, 550
 OsclSocketServI
 Close, 546
 Connect, 546
 IsServerThread, 547
 LoopbackSocket, 547
 NewL, 547
 OsclDNSI, 547
 OsclSocketI, 547
 OsclSocketRequest, 547
 OsclSocketServ, 547
 OsclSocketServRequestList, 547
 OsclTCPSocketI, 547
 OsclUDPSocketI, 547
 OsclSocketServIBase, 548
 ESocketServ_Connected, 548
 ESocketServ_Error, 549
 ESocketServ_Idle, 548
 OsclSocketServIBase, 549
 OsclSocketServIBase
 ~OsclSocketServIBase, 549
 Close, 549
 Connect, 549
 iAlloc, 549
 iLogger, 549
 iServerError, 549
 iServState, 549
 IsServConnected, 549
 OsclSocketServIBase, 549
 State, 549
 TSocketServState, 548
 OsclSocketServRequestList, 550
 OsclSocketServI, 547
 OsclSocketServRequestList, 550
 OsclSocketServRequestList
 Add, 550
 Close, 550
 Open, 550
 OsclSocketServI, 550
 OsclSocketServRequestList, 550
 Remove, 550
 StartCancel, 550
 WaitOnRequests, 550
 Wakeup, 550
 OsclSocketServRequestQElem, 552
 OsclSocketServRequestQElem, 552
 OsclSocketServRequestQElem
 iCancel, 552
 iSelect, 552
 iSocketRequest, 552
 OsclSocketServRequestQElem, 552
 OsclSocketStartup
 osclconfig_io.h, 801
 OsclSuccess
 osclerror, 90
 OsclTagTreeType
 osclmemory, 57
 OsclTCPSocket, 553
 OsclSocketI, 528
 OsclSocketIBase, 534

OsclSocketServ, 545
 OsclTCPSocket
 ~OsclTCPSocket, 554
 Accept, 554
 Bind, 554
 BindAsync, 554
 CancelAccept, 554
 CancelBind, 555
 CancelConnect, 555
 CancelListen, 555
 CancelRecv, 555
 CancelSend, 555
 CancelShutdown, 555
 Close, 555
 Connect, 556
 GetAcceptedSocketL, 556
 GetRecvData, 556
 GetSendData, 556
 Listen, 557
 ListenAsync, 557
 NewL, 557
 Recv, 557
 Send, 558
 Shutdown, 558
 OsclTCPSocketI, 559
 OsclSocketServI, 547
 OsclTCPSocketI
 ~OsclTCPSocketI, 560
 Accept, 560
 BindAsync, 560
 CancelAccept, 560
 CancelBind, 560
 CancelConnect, 560
 CancelListen, 560
 CancelRecv, 560
 CancelSend, 560
 CancelShutdown, 560
 Close, 560
 Connect, 560
 GetAcceptedSocketL, 560
 GetRecvData, 560
 GetSendData, 560
 Listen, 560
 ListenAsync, 561
 NewL, 561
 Recv, 561
 Send, 561
 Shutdown, 561
 OsclThread, 562
 OsclThread, 562
 OsclThread
 ~OsclThread, 562
 CompareId, 562
 Create, 563
 EnableKill, 563
 Exit, 563
 GetId, 563
 GetPriority, 564
 OsclThread, 562
 Resume, 564
 SetPriority, 564
 SleepMillisec, 564
 Suspend, 565
 Terminate, 565
 OsclThread_State
 oscl_thread.h, 773
 OsclThreadLock, 566
 OsclThreadLock, 566
 OsclThreadLock
 ~OsclThreadLock, 566
 Lock, 566
 OsclThreadLock, 566
 Unlock, 566
 OsclThreadPriority
 oscl_thread.h, 773
 OsclTickCount, 567
 OsclTickCount
 MsecToTicks, 567
 TickCount, 567
 TickCountFrequency, 567
 TickCountPeriod, 567
 TicksToMsec, 567
 OSCLTICKCOUNT_MAX_TICKS
 osclutil, 67
 OsclTimer, 569
 OsclTimer, 570
 OsclTimer
 ~OsclTimer, 570
 callback_timer_type, 570
 CallbackTimer< Alloc >, 571
 Cancel, 570
 Clear, 570
 OsclTimer, 570
 Request, 570
 SetExactFrequency, 570
 SetFrequency, 571
 SetObserver, 571
 TimerBaseElapsed, 571
 OsclTimerCompare, 572
 OsclExecSchedulerCommonBase, 392
 OsclTimerCompare
 compare, 572
 OsclTimerObject, 573
 OsclExecSchedulerCommonBase, 394
 OsclTimerObject, 574
 PVActiveBase, 600
 PVActiveStats, 601
 PVThreadContext, 620

OsclTimerObject
 ~OsclTimerObject, 574
 AddToScheduler, 574
 After, 574
 Cancel, 574
 DoCancel, 574
 IsBusy, 575
 OsclTimerObject, 574
 Priority, 575
 RemoveFromScheduler, 575
 RunError, 575
 RunIfNotReady, 575
 SetBusy, 575
 SetStatus, 575
 Status, 576
 StatusRef, 576
OsclTimerObserver, 577
OsclTimerObserver
 ~OsclTimerObserver, 577
 TimeoutOccurred, 577
OsclTimerQ, 578
OsclTimerQ
 Add, 578
 Construct, 578
 IsIn, 578
 Pop, 578
 PopTop, 578
 Remove, 578
 Top, 578
OsclTLS, 579
 OsclTLS, 579
OsclTLS
 ~OsclTLS, 579
 _Ptr, 580
 operator *, 579
 operator->, 579
 OsclTLS, 579
 set, 579
OsclTLSEEx, 581
 OsclTLSEEx, 581
OsclTLSEEx
 ~OsclTLSEEx, 581
 _Ptr, 582
 operator *, 581
 operator->, 581
 OsclTLSEEx, 581
 set, 581
OsclTLSRegistry, 583
OsclTLSRegistry
 getInstance, 583
 OsclBase, 583
 registerInstance, 583
OsclTLSRegistryEx, 584
OsclTLSRegistryEx
 getInstance, 584
 registerInstance, 584
OsclTrapItem, 585
 OsclTrapItem, 585
OsclTrapItem
 OsclTrapItem, 585
 OsclTrapStack, 585
 OsclTrapStackItem, 585
OsclTrapOperation
 osclerror, 91
OsclTrapStack, 586
 OsclErrorTrapImp, 373
 OsclTrapItem, 585
OsclTrapStack
 OsclError, 586
 OsclErrorTrap, 586
 OsclErrorTrapImp, 586
OsclTrapStackItem, 587
 OsclTrapItem, 585
 OsclTrapStackItem, 587
OsclTrapStackItem
 iCBase, 587
 iNext, 587
 iTAny, 587
 iTrapOperation, 587
 OsclTrapStackItem, 587
OsclUDPSocket, 588
 OsclSocketI, 528
 OsclSocketIBase, 534
 OsclSocketServ, 545
OsclUDPSocket
 ~OsclUDPSocket, 588
 Bind, 589
 BindAsync, 589
 CancelBind, 589
 CancelRecvFrom, 589
 CancelSendTo, 589
 Close, 589
 GetRecvData, 590
 GetSendData, 590
 Join, 590
 NewL, 590
 RecvFrom, 591
 SendTo, 591
 SetRecvBufferSize, 591
OsclUDPSocketI, 593
 OsclSocketServI, 547
OsclUDPSocketI
 ~OsclUDPSocketI, 594
 BindAsync, 594
 CancelBind, 594
 CancelRecvFrom, 594
 CancelSendTo, 594
 Close, 594

GetRecvData, 594
 GetSendData, 594
 NewL, 594
 RecvFrom, 594
 SendTo, 594
OsclUid32
 oscl_uuid.h, 784
OsclUnMakeSockAddr
 osclconfig_io.h, 801
osclutil
 ~OSCL_HeapString, 82
 ~OSCL_StackString, 82
 ~OSCL_wHeapString, 82
 ~OSCL_wStackString, 82
 APPEND_MEDIA_AT_END, 82
 BufferFreeFuncPtr, 67
 EOSCL_StringOp_CompressASCII, 68
 EOSCL_StringOp_UTF16ToUTF8, 68
 EOSCL_wStringOp_ExpandASCII, 68
 EOSCL_wStringOp_UTF8ToUTF16, 68
 extract_string, 68
 get_cstr, 68
 get_maxsize, 69
 get_size, 69
 get_str, 70
 GetBufferState, 70
 GetFragment, 70
 MAX_NUMBER_OF_BYTE_PER_UTF8,
 67
 MediaTimestamp, 67
 operator=, 70–72
 oscl_abs, 72
 OSCL_ASCII_CASE_MAGIC_BIT, 82
 oscl_asin, 72
 oscl_atan, 72
 oscl_cos, 72
 oscl_exp, 72
 oscl_floor, 72
 OSCL_HeapString, 72, 73
 oscl_isdigit, 67
 oscl_log, 73
 oscl_log10, 73
 oscl_pow, 73
 oscl_sin, 74
 oscl_snprintf, 74
 oscl_sqrt, 74
 OSCL_StackString, 74, 75
 oscl_str_escape_xml, 75
 oscl_str_is_valid_utf8, 75
 oscl_str_need_escape_xml, 76
 oscl_str_truncate_utf8, 76
 oscl_str_unescape_uri, 76, 77
 oscl_tan, 77
 OSCL_TStrPtrLen, 67
 oscl_UnicodeToUTF8, 77
 oscl_UTF8ToUnicode, 78
 oscl_vsnprintf, 78, 80
 OSCL_wHeapString, 80
 OSCL_wStackString, 80
 OsclComponentFactory, 67
 OSCLTICKCOUNT_MAX_TICKS, 67
 PV_atof, 80
 PV_atoi, 80
 set, 80–82
 skip_to_line_term, 82
 skip_to_whitespace, 82
 skip_whitespace, 82
 skip_whitespace_and_line_term, 82
 StrCSumPtrLen, 67
 StrPtrLen, 67
 TOSCL_StringOp, 68
 TOSCL_wStringOp, 68
 WStrPtrLen, 67
OsclUuid, 595
 OsclUuid, 596
OsclUuid
 data1, 596
 data2, 596
 data3, 596
 data4, 596
 operator!=, 596
 operator=, 596
 operator==, 596
 OsclUuid, 596
OsclValidInetAddr
 osclconfig_io.h, 801
other
 Oscl_TAlloc::rebind, 279
other_chartype
 OSCL_FastString, 173
 OSCL_HeapString, 194
 OSCL_HeapStringA, 196
 OSCL_StackString, 252
 OSCL_wFastString, 290
 OSCL_wHeapString, 293
 OSCL_wHeapStringA, 295
 OSCL_wStackString, 298
OTHER_ERROR
 OsclProcStatus, 465
OUTOFMEMORY_ERROR
 OsclProcStatus, 465
overwrite
 CFastRep, 126
pad
 MM_AllocBlockFence, 144
 MM_AllocBlockHdr, 145
pair_citerator_citerator

Oscl_Map, 213
 pair_iterator_bool
 Oscl_Map, 213
 Oscl_TagTree, 265
 pair_iterator_iterator
 Oscl_Map, 213
 pAllocInfo
 MM_AllocNode, 148
 parent
 Oscl_Rb_Tree_Node_Base, 249
 Oscl_TagTree::Node, 275
 pAudit
 OsclAuditCB, 316
 pBasePosition
 OsclBinStream, 334
 pBuffer
 OsclFileCacheBuffer, 399
 peakNumAllocs
 MM_Stats_t, 162
 peakNumBytes
 MM_Stats_t, 162
 PendComplete
 OsclActiveObject, 307
 OsclExecSchedulerCommonBase, 391
 OsclReadyQ, 475
 PendForExec
 OsclActiveObject, 307
 per_allocation_overhead
 MM_AuditOverheadStats, 158
 perms
 oscl_stat_buf, 253
 pFileName
 MM_AllocInfo, 147
 pMemBlock
 MM_AllocInfo, 147
 MM_AllocQueryInfo, 149
 pMMFIParam
 OsclMemStatsNode, 448
 pMMStats
 OsclMemStatsNode, 448
 pNext
 MM_AllocNode, 148
 pNode
 MM_AllocBlockHdr, 145
 pointer
 MemAllocator, 143
 Oscl_Map, 213
 Oscl_Queue, 231
 Oscl_Rb_Tree, 238
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
 Oscl_TagTree::const_iterator, 269
 Oscl_TagTree::iterator, 272
 Oscl_TAlloc, 277

 Oscl_Vector, 281
 Pop
 OsclError, 367
 OsclTimerQ, 578
 pop
 Oscl_Queue, 232
 Oscl_Queue_Base, 234
 OsclPriorityQueue, 462
 pop_back
 Oscl_Vector, 283
 Oscl_Vector_Base, 287
 pop_heap
 OsclPriorityQueue, 462
 OsclPriorityQueueBase, 464
 PopDealloc
 OsclError, 367, 368
 PopTop
 OsclReadyQ, 475
 OsclTimerQ, 578
 port
 OsclNetworkAddress, 456
 PositionInBlock
 OsclBinStream, 333
 pPosition
 OsclBinStream, 334
 pPrev
 MM_AllocNode, 148
 Preceeds
 OsclFileCacheBuffer, 399
 PrepRead
 OsclFileCacheBuffer, 399
 PrepWrite
 OsclFileCacheBuffer, 399
 Priority
 OsclActiveObject, 307
 OsclTimerObject, 575
 ProcessAccept
 OsclSocketI, 527
 ProcessConnect
 OsclSocketI, 527
 ProcessRecv
 OsclSocketI, 527
 ProcessRecvFrom
 OsclSocketI, 527
 ProcessSend
 OsclSocketI, 527
 ProcessSendTo
 OsclSocketI, 527
 ProcessShutdown
 OsclSocketI, 527
 pRootNode
 MM_AllocBlockHdr, 145
 pruneSubtree
 MM_Audit_Imp, 156

PSHARED_ATTRIBUTE_SETTING_ERROR
 OsclProcStatus, [466](#)
PSHARED_NOT_ZERO_ERROR
 OsclProcStatus, [466](#)
pStats
 MM_Stats_CB, [160](#)
pStatsNode
 MM_AllocInfo, [147](#)
 OsclAuditCB, [316](#)
Ptr
 OsclPtr, [467](#)
 OsclPtrC, [470](#)
ptr
 OsclMemoryFragment, [431](#)
 StrPtrLen, [633](#)
 WStrPtrLen, [643](#)
push
 Oscl_Queue, [232](#)
 Oscl_Queue_Base, [234](#)
 OsclPriorityQueue, [462](#)
push_back
 Oscl_Vector, [284](#)
 Oscl_Vector_Base, [287](#)
push_front
 Oscl_Vector, [284](#)
 Oscl_Vector_Base, [288](#)
push_heap
 OsclPriorityQueue, [462](#)
 OsclPriorityQueueBase, [464](#)
PushL
 OsclError, [368](#)
PV8601TIME_BUFFER_SIZE
 osclbase, [44](#)
PV8601timeStrBuf
 osclbase, [33](#)
PV8601ToRFC822
 osclbase, [42](#)
PV_atof
 osclutil, [80](#)
PV_atoi
 osclutil, [80](#)
PV_CHAR_CLOSE_BRACKET
 oscl_uuid.h, [784](#)
PV_CHAR_COMMA
 oscl_uuid.h, [784](#)
PV_DNS_IS_THREAD
 oscl_dns_tuneables.h, [661](#)
PV_DNS_SERVER
 oscl_dns_tuneables.h, [661](#)
PV_DYNAMIC_LOADING_CONFIG_FILE_PATH
 osclconfig_lib.h, [804](#)
PV_OSCL_SOCKET_1MB_RECV_BUF
 oscl_socket_tuneables.h, [757](#)
PV_OSCL_SOCKET_SERVER_LOGGER_OUTPUT
 oscl_socket_tuneables.h, [757](#)
PV_OSCL_SOCKET_STATS_LOGGING
 oscl_socket_tuneables.h, [757](#)
PV_RUNTIME_LIB_FILENAME_EXTENSION
 osclconfig_lib.h, [804](#)
PV_SCHED_CHECK_Q
 osclproc, [103](#)
PV_SCHED_ENABLE_AO_STATS
 osclproc, [103](#)
PV_SCHED_ENABLE_LOOP_STATS
 osclproc, [103](#)
PV_SCHED_ENABLE_PERF_LOGGING
 osclproc, [103](#)
PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS
 osclproc, [103](#)
PV_SCHED_FAIR_SCHEDULING
 osclproc, [103](#)
PV_SCHED_LOG_Q
 osclproc, [103](#)
PV_SOCKET_REQUEST_AO_PRIORITY
 oscl_socket_tuneables.h, [757](#)
PV_SOCKET_SERVER
 oscl_socket_tuneables.h, [757](#)
PV_SOCKET_SERVER_AO_INTERVAL_MSEC
 oscl_socket_tuneables.h, [758](#)
PV_SOCKET_SERVER_AO_PRIORITY
 oscl_socket_tuneables.h, [758](#)
PV_SOCKET_SERVER_IS_THREAD
 oscl_socket_tuneables.h, [758](#)
PV_SOCKET_SERVER_SELECT
 oscl_socket_tuneables.h, [758](#)
PV_SOCKET_SERVER_SELECT_LOOPBACK_SOCKET
 oscl_socket_tuneables.h, [758](#)
PV_SOCKET_SERVER_SELECT_TIMEOUT_MSEC
 oscl_socket_tuneables.h, [758](#)
PV_SOCKET_SERVER_THREAD_PRIORITY
 oscl_socket_tuneables.h, [758](#)
PV_SOCKET_SERVI_STATS
 oscl_socket_tuneables.h, [758](#)
PVActiveBase, [597](#)
 OsclExecSchedulerBase, [386](#)
 OsclExecSchedulerCommonBase, [394](#)
PVActiveBase, [598](#)
 PVActiveStats, [601](#)
 PVThreadContext, [620](#)
PVActiveBase

~PVActiveBase, 598
 Activate, 598
 AddToScheduler, 598
 Cancel, 598
 Destroy, 598
 DoCancel, 598
 iAddedNum, 600
 iBusy, 600
 iName, 600
 iPVActiveStats, 600
 iPVReadyQLink, 600
 IsAdded, 598
 IsInAnyQ, 599
 iStatus, 600
 iThreadContext, 600
 OsclActiveObject, 600
 OsclExecScheduler, 600
 OsclReadyCompare, 600
 OsclReadyQ, 600
 OsclReadySetPosition, 600
 OsclSchedulerCommonBase, 600
 OsclTimerObject, 600
 PVActiveBase, 598
 PVActiveStats, 600
 RemoveFromScheduler, 599
 Run, 599
 RunError, 599
 PVActiveStats, 601
 OsclExecSchedulerCommonBase, 394
 PVActiveBase, 600
 PVActiveStats
 OsclActiveObject, 601
 OsclExecScheduler, 601
 OsclExecSchedulerCommonBase, 601
 OsclReadyQ, 601
 OsclTimerObject, 601
 PVActiveBase, 601
 PVCleanupStack
 _OsclHeapBase, 109
 PVERROR_DoLeave
 oscl_error_imp_fatalerror.h, 669
 oscl_error_imp_jumps.h, 671
 osclerror, 90
 PVERROR_IMP_JUMPS
 osclerror, 90
 PVERRORTRAP_REGISTRY
 osclerror, 90
 PVERRORTRAP_REGISTRY_ID
 osclerror, 91
 PVEXECNAMELEN
 osclproc, 103
 PVLogger, 602
 ~PVLogger, 603
 AddAppender, 603
 AddFilter, 603
 alloc_type, 603
 Cleanup, 604
 DisableAppenderInheritance, 604
 filter_status_type, 603
 GetLoggerObject, 604
 GetLogLevel, 604
 GetNumAppenders, 604
 GetParent, 605
 Init, 605
 IsActive, 605
 log_level_type, 603
 LogMsgBuffers, 605
 LogMsgBuffersV, 605
 LogMsgString, 606
 LogMsgStringV, 606
 message_id_type, 603
 PVLogger, 603
 PVLoggerRegistry, 607
 RemoveAppender, 606
 SetLogLevel, 606
 SetLogLevelAndPropagate, 607
 SetParent, 607
 pvlogger.h, 829
 _PVLOGGER_LOGBIN, 831
 _PVLOGGER_LOGBIN_V, 831
 _PVLOGGER_LOGMSG, 831
 _PVLOGGER_LOGMSG_V, 831
 PVLOGGER_ENABLE, 831
 PVLOGGER_INST_LEVEL, 832
 PVLOGGER_INST_LEVEL_SUPPORT, 832
 PVLOGGER_LEVEL_UNINITIALIZED, 835
 PVLOGGER_LOG_USE_ONLY, 832
 PVLOGGER_LOGBIN, 832
 PVLOGGER_LOGBIN_PVLOGMSG_-
 INST_HLDBG, 832
 PVLOGGER_LOGBIN_PVLOGMSG_-
 INST_LLDBG, 833
 PVLOGGER_LOGBIN_PVLOGMSG_-
 INST_MLDBG, 833
 PVLOGGER_LOGBIN_PVLOGMSG_-
 INST_PROF, 833
 PVLOGGER_LOGBIN_PVLOGMSG_-
 INST_REL, 833
 PVLOGGER_LOGBIN_V, 833
 PVLOGGER_LOGBIN_V_-
 PVLOGMSG_INST_HLDBG, 833
 PVLOGGER_LOGBIN_V_-
 PVLOGMSG_INST_LLDBG, 833
 PVLOGGER_LOGBIN_V_-
 PVLOGMSG_INST_PROF, 833

PVLOGGER_LOGBIN_V_-
 PVLOGMSG_INST_REL, 833
 PVLOGGER_LOGBIN_V_-
 PVLOGMSG_V_INST_MLDBG,
 833
 PVLOGGER_LOGMSG, 833
 PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_HLDBG, 833
 PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_LLDBG, 834
 PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_MLDBG, 834
 PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_PROF, 834
 PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_REL, 834
 PVLOGGER_LOGMSG_V, 834
 PVLOGGER_LOGMSG_V_-
 PVLOGMSG_INST_HLDBG, 834
 PVLOGGER_LOGMSG_V_-
 PVLOGMSG_INST_LLDBG, 834
 PVLOGGER_LOGMSG_V_-
 PVLOGMSG_INST_MLDBG,
 834
 PVLOGGER_LOGMSG_V_-
 PVLOGMSG_INST_PROF, 834
 PVLOGGER_LOGMSG_V_-
 PVLOGMSG_INST_REL, 834
 PVLOGMSG_ALERT, 835
 PVLOGMSG_CRIT, 835
 PVLOGMSG_DEBUG, 835
 PVLOGMSG_EMERG, 835
 PVLOGMSG_ERR, 835
 PVLOGMSG_FATAL_ERROR, 835
 PVLOGMSG_INFO, 836
 PVLOGMSG_INST_HLDBG, 834
 PVLOGMSG_INST_LLDBG, 834
 PVLOGMSG_INST_MLDBG, 834
 PVLOGMSG_INST_PROF, 835
 PVLOGMSG_INST_REL, 835
 PVLOGMSG_NONFATAL_ERROR, 836
 PVLOGMSG_NOTICE, 836
 PVLOGMSG_STACK_TRACE, 836
 PVLOGMSG_STATISTIC, 836
 PVLOGMSG_VERBOSE, 836
 PVLOGMSG_WARNING, 836
 pvlogger_accessories.h, 837
 PVLOGGER_FILTER_ACCEPT, 837
 PVLOGGER_FILTER_NEUTRAL, 837
 PVLOGGER_FILTER_REJECT, 837
 pvlogger_c.h, 838
 PVLOGGER_C_INST_LEVEL, 839
 pvLogger_GetLoggerObject, 839
 pvLogger_IsActive, 839
 pvLogger_LogMsgString, 839
 PVLOGMSG_C_ALERT, 839
 PVLOGMSG_C_CRIT, 839
 PVLOGMSG_C_EMERG, 839
 PVLOGMSG_C_ERR, 839
 PVLOGMSG_C_INFO, 839
 PVLOGMSG_C_INST_HLDBG, 839
 PVLOGMSG_C_INST_LLDBG, 839
 PVLOGMSG_C_INST_MLDBG, 839
 PVLOGMSG_C_INST_PROF, 839
 PVLOGMSG_C_INST_REL, 839
 PVLOGMSG_C_NOTICE, 839
 PVLOGMSG_C_STACK_DEBUG, 839
 PVLOGMSG_C_STACK_TRACE, 839
 PVLOGMSG_C_WARNING, 839
 PVLOGGER_C_INST_LEVEL
 pvlogger_c.h, 839
 PVLOGGER_ENABLE
 pvlogger.h, 831
 PVLOGGER_FILTER_ACCEPT
 pvlogger_accessories.h, 837
 PVLOGGER_FILTER_NEUTRAL
 pvlogger_accessories.h, 837
 PVLOGGER_FILTER_REJECT
 pvlogger_accessories.h, 837
 pvLogger_GetLoggerObject
 pvlogger_c.h, 839
 PVLOGGER_INST_LEVEL
 pvlogger.h, 832
 PVLOGGER_INST_LEVEL_SUPPORT
 pvlogger.h, 832
 pvLogger_IsActive
 pvlogger_c.h, 839
 PVLOGGER_LEVEL_UNINITIALIZED
 pvlogger.h, 835
 PVLOGGER_LOG_USE_ONLY
 pvlogger.h, 832
 PVLOGGER_LOGBIN
 pvlogger.h, 832
 PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 HLDBG
 pvlogger.h, 832
 PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 LLDBG
 pvlogger.h, 833
 PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 MLDBG
 pvlogger.h, 833
 PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 PROF
 pvlogger.h, 833
 PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 REL
 pvlogger.h, 833

PVLOGGER_LOGBIN_V
 pvlogger.h, 833

PVLOGGER_LOGBIN_V_PVLOGMSG_-
 INST_HLDBG
 pvlogger.h, 833

PVLOGGER_LOGBIN_V_PVLOGMSG_-
 INST_LLDBG
 pvlogger.h, 833

PVLOGGER_LOGBIN_V_PVLOGMSG_-
 INST_PROF
 pvlogger.h, 833

PVLOGGER_LOGBIN_V_PVLOGMSG_-
 INST_REL
 pvlogger.h, 833

PVLOGGER_LOGBIN_V_PVLOGMSG_V_-
 INST_MLDBG
 pvlogger.h, 833

PVLOGGER_LOGMSG
 pvlogger.h, 833

PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_HLDBG
 pvlogger.h, 833

PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_LLDBG
 pvlogger.h, 834

PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_MLDBG
 pvlogger.h, 834

PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_PROF
 pvlogger.h, 834

PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_REL
 pvlogger.h, 834

PVLOGGER_LOGMSG_V
 pvlogger.h, 834

PVLOGGER_LOGMSG_V_PVLOGMSG_-
 INST_HLDBG
 pvlogger.h, 834

PVLOGGER_LOGMSG_V_PVLOGMSG_-
 INST_LLDBG
 pvlogger.h, 834

PVLOGGER_LOGMSG_V_PVLOGMSG_-
 INST_MLDBG
 pvlogger.h, 834

PVLOGGER_LOGMSG_V_PVLOGMSG_-
 INST_PROF
 pvlogger.h, 834

PVLOGGER_LOGMSG_V_PVLOGMSG_-
 INST_REL
 pvlogger.h, 834

pvLogger_LogMsgString
 pvlogger_c.h, 839

pvlogger_registry.h, 840

PVLoggerAppender, 608

PVLoggerAppender
 ~PVLoggerAppender, 608

AppendBuffers, 608

AppendString, 608

message_id_type, 608

PVLoggerFilter, 609

PVLoggerFilter
 ~PVLoggerFilter, 610

filter_status_type, 609

FilterOpaqueMessge, 610

FilterString, 610

log_level_type, 609

message_id_type, 609

PVLoggerLayout, 611

PVLoggerLayout
 ~PVLoggerLayout, 611

FormatOpaqueMessage, 611

FormatString, 611

message_id_type, 611

PVLoggerRegistry, 613

PVLogger, 607

PVLoggerRegistry, 613

PVLoggerRegistry
 ~PVLoggerRegistry, 613

alloc_type, 613

CreatePVLogger, 614

GetPVLoggerObject, 614

GetPVLoggerRegistry, 614

log_level_type, 613

PVLoggerRegistry, 613

SetNodeLogLevelExplicit, 614

PVLOGMSG_ALERT
 pvlogger.h, 835

PVLOGMSG_C_ALERT
 pvlogger_c.h, 839

PVLOGMSG_C_CRIT
 pvlogger_c.h, 839

PVLOGMSG_C_EMERG
 pvlogger_c.h, 839

PVLOGMSG_C_ERR
 pvlogger_c.h, 839

PVLOGMSG_C_INFO
 pvlogger_c.h, 839

PVLOGMSG_C_INST_HLDBG
 pvlogger_c.h, 839

PVLOGMSG_C_INST_LLDBG
 pvlogger_c.h, 839

PVLOGMSG_C_INST_MLDBG
 pvlogger_c.h, 839

PVLOGMSG_C_INST_PROF
 pvlogger_c.h, 839

PVLOGMSG_C_INST_REL
 pvlogger_c.h, 839

PVLOGMSG_C_NOTICE
pvlogger_c.h, 839

PVLOGMSG_C_STACK_DEBUG
pvlogger_c.h, 839

PVLOGMSG_C_STACK_TRACE
pvlogger_c.h, 839

PVLOGMSG_C_WARNING
pvlogger_c.h, 839

PVLOGMSG_CRIT
pvlogger.h, 835

PVLOGMSG_DEBUG
pvlogger.h, 835

PVLOGMSG_EMERG
pvlogger.h, 835

PVLOGMSG_ERR
pvlogger.h, 835

PVLOGMSG_FATAL_ERROR
pvlogger.h, 835

PVLOGMSG_INFO
pvlogger.h, 836

PVLOGMSG_INST_HLDBG
pvlogger.h, 834

PVLOGMSG_INST_LLDBG
pvlogger.h, 834

PVLOGMSG_INST_MLDBG
pvlogger.h, 834

PVLOGMSG_INST_PROF
pvlogger.h, 835

PVLOGMSG_INST_REL
pvlogger.h, 835

PVLOGMSG_NONFATAL_ERROR
pvlogger.h, 836

PVLOGMSG_NOTICE
pvlogger.h, 836

PVLOGMSG_STACK_TRACE
pvlogger.h, 836

PVLOGMSG_STATISTIC
pvlogger.h, 836

PVLOGMSG_VERBOSE
pvlogger.h, 836

PVLOGMSG_WARNING
pvlogger.h, 836

PVMEM_INST_LEVEL
osclbase, 33
osclconfig_memory.h, 807

PVNETWORKADDRESS_LEN
oscl_socket_types.h, 759

PVOsclBase_Cleanup
osclbase, 43

PVOsclBase_Init
osclbase, 43

PVSCHEDNAMELEN
osclproc, 103

PVSchedulerStopper
616

OsclExecSchedulerCommonBase
394

PVSchedulerStopper
616

PVSchedulerStopper
~PVSchedulerStopper, 616

PVSchedulerStopper
PVSchedulerStopper, 616

PVSOCK_ERR_BAD_PARAM
oscl_socket_imp_pv.h, 742

PVSOCK_ERR_NOT_IMPLEMENTED
oscl_socket_imp_pv.h, 742

PVSOCK_ERR_SERV_NOT_CONNECTED
oscl_socket_imp_pv.h, 742

PVSOCK_ERR_SOCK_NO_SERV
oscl_socket_imp_pv.h, 742

PVSOCK_ERR_SOCK_NOT_CONNECTED
oscl_socket_imp_pv.h, 742

PVSOCK_ERR_SOCK_NOT_OPEN
oscl_socket_imp_pv.h, 742

PVSockBufRecv
617

PVSockBufRecv
PVSockBufRecv, 617

PVSockBufRecv
iLen, 617

iMaxLen, 617

iPtr, 617

PVSockBufRecv
617

PVSockBufSend
618

PVSockBufSend
PVSockBufSend, 618

PVSockBufSend
iLen, 618

iPtr, 618

PVSockBufSend
618

PVThreadContext
619

OsclExecSchedulerCommonBase
394

PVThreadContext
619

PVThreadContext
~PVThreadContext, 619

EnterThreadContext
619

ExitThreadContext
619

Id
619

IsSameThreadContext
619

OsclActiveObject
620

OsclCoeActiveScheduler
620

OsclCoeActiveSchedulerBase
620

OsclExecScheduler
620

OsclExecSchedulerBase
620

OsclExecSchedulerCommonBase
620

OsclTimerObject
620

PVActiveBase
620

PVThreadContext
619

ThreadHasScheduler
620

QUE_ITER_BEGIN
osclproc, 103

QUE_ITER_END
osclproc, 103

Rand
 OsclRand, [471](#)
 Read
 Oscl_File, [180](#)
 OsclAsyncFile, [313](#)
 OsclBinIStreamBigEndian, [322](#)
 OsclFileCache, [397](#)
 OsclNativeFile, [453](#)
 read
 OSCL_String, [257](#)
 OSCL_wString, [301](#)
 Read_uint16
 OsclBinIStreamBigEndian, [322](#)
 OsclBinIStreamLittleEndian, [325](#)
 Read_uint32
 OsclBinIStreamBigEndian, [322](#)
 OsclBinIStreamLittleEndian, [325](#)
 Read_uint8
 OsclBinIStream, [319](#)
 ReadAsync
 OsclNativeFile, [453](#)
 ReadAsyncCancel
 OsclNativeFile, [453](#)
 rebalance
 Oscl_Rb_Tree_Base, [240](#)
 rebalance_for_erase
 Oscl_Rb_Tree_Base, [240](#)
 Recv
 OsclRecvMethod, [480](#)
 OsclRecvRequest, [481](#)
 OsclSocketI, [527](#)
 OsclSocketIBase, [533](#)
 OsclTCPSocket, [557](#)
 OsclTCPSocketI, [561](#)
 RecvFrom
 OsclRecvFromMethod, [476](#)
 OsclRecvFromRequest, [478](#)
 OsclSocketI, [527](#)
 OsclSocketIBase, [533](#)
 OsclUDPSocket, [591](#)
 OsclUDPSocketI, [594](#)
 RecvFromParam, [621](#)
 RecvFromParam, [621](#)
 RecvFromParam
 iAddr, [621](#)
 iBufRecv, [621](#)
 iFlags, [621](#)
 iMultiMaxLen, [621](#)
 iPacketLen, [621](#)
 iPacketSource, [621](#)
 RecvFromParam, [621](#)
 RecvFromRequest
 OsclRecvFromMethod, [476](#)
 RecvFromSuccess

OsclSocketI, [527](#)
 OsclSocketIBase, [533](#)
 RecvParam, [623](#)
 RecvParam, [623](#)
 RecvParam
 iBufRecv, [623](#)
 iFlags, [623](#)
 RecvParam, [623](#)
 RecvRequest
 OsclRecvMethod, [480](#)
 RecvSuccess
 OsclSocketI, [527](#)
 OsclSocketIBase, [533](#)
 red
 Oscl_Rb_Tree_Node_Base, [248](#)
 RedBl
 Oscl_Rb_Tree_Node_Base, [248](#)
 refcount
 CHearRep, [128](#)
 reference
 Oscl_Map, [213](#)
 Oscl_Queue, [231](#)
 Oscl_Rb_Tree, [238](#)
 Oscl_Rb_Tree_Const_Iterator, [242](#)
 Oscl_Rb_Tree_Iterator, [245](#)
 Oscl_TagTree::const_iterator, [269](#)
 Oscl_TagTree::iterator, [272](#)
 Oscl_TAlloc, [277](#)
 Oscl_Vector, [281](#)
 Register
 OsclComponentRegistry, [339](#)
 OsclRegistryClient, [499](#)
 OsclRegistryClientImpl, [502](#)
 OsclRegistryServTlsImpl, [505](#)
 RegisterForCallback
 OsclExecScheduler, [384](#)
 OsclReadyQ, [475](#)
 registerInstance
 OsclSingletonRegistry, [524](#)
 OsclTLSRegistry, [583](#)
 OsclTLSRegistryEx, [584](#)
 registerInstanceAndUnlock
 OsclSingletonRegistry, [524](#)
 release
 OsclExclusiveArrayPtr, [377](#)
 OsclExclusivePtr, [380](#)
 OsclExclusivePtrA, [383](#)
 OSCLMemAutoPtr, [426](#)
 RELOCK_MUTEX_ERROR
 OsclProcStatus, [466](#)
 Remove
 OsclDoubleLink, [361](#)
 OsclReadyQ, [475](#)
 OsclSocketServRequestList, [550](#)

OsclTimerQ, 578
 remove
 OsclPriorityQueue, 462
 OsclPriorityQueueBase, 464
 remove_element
 Oscl_Linked_List, 206
 Oscl_Linked_List_Base, 210
 Oscl_MTLLinked_List, 222
 remove_ref
 CHheapRep, 128
 removeAllAllocNodes
 MM_Audit_Imp, 156
 removeAllocNode
 MM_Audit_Imp, 156
 RemoveAppender
 PVLogger, 606
 RemoveFixedCache
 Oscl_File, 180
 RemoveFromScheduler
 OsclActiveObject, 307
 OsclTimerObject, 575
 PVActiveBase, 599
 RemoveRef
 DNSRequestParam, 132
 removeRef
 Oscl_DefAllocWithRefCounter, 171
 OsclMemPoolFixedChunkAllocator, 435
 OsclMemPoolResizableAllocator, 442
 OsclRefCounter, 482
 OsclRefCounterDA, 485
 OsclRefCounterMTDA, 489
 OsclRefCounterMTSA, 491
 OsclRefCounterSA, 493
 Request
 OsclTimer, 570
 RequestCanceled
 OsclExecSchedulerCommonBase, 391
 RequestDone
 OsclDNSRequestAO, 359
 OsclSocketRequestAO, 542
 reserve
 Oscl_Queue_Base, 234
 Oscl_Vector_Base, 288
 OsclPriorityQueue, 462
 ReserveSpace
 OsclBinStream, 333
 Reset
 OsclDoubleListBase, 364
 reset
 BufferState, 117
 MM_FailInsertParam, 159
 MM_Stats_t, 162
 OsclMemStatsNode, 448
 ResetLogPerf

OsclExecSchedulerCommonBase, 391
 Resume
 OsclThread, 564
 ResumeScheduler
 OsclExecSchedulerCommonBase, 391
 retrieveParentTag
 MM_Audit_Imp, 156
 retrieveParentTagLength
 MM_Audit_Imp, 156
 RFC822ToPV8601
 osclbase, 43
 Right
 OsclPtrC, 470
 right
 Oscl_Rb_Tree_Node_Base, 249
 rotate_left
 Oscl_Rb_Tree_Base, 240
 rotate_right
 Oscl_Rb_Tree_Base, 240
 Run
 CallbackTimer, 122
 OsclDNSMethod, 354
 OsclDNSRequestAO, 359
 OsclSocketMethod, 536
 OsclSocketRequestAO, 542
 PVActiveBase, 599
 RunError
 OsclActiveObject, 307
 OsclTimerObject, 575
 PVActiveBase, 599
 RunIfNotReady
 OsclActiveObject, 308
 OsclTimerObject, 575
 RunSchedulerNonBlocking
 OsclExecScheduler, 384
 save_registry
 TLSStorageOps, 640
 second
 Oscl_Pair, 229
 SECONDS
 osclbase, 34
 Seed
 OsclRand, 471
 Seek
 Oscl_File, 180
 OsclAsyncFile, 313
 OsclBinStream, 333
 OsclFileCache, 397
 OsclNativeFile, 454
 seek_type
 Oscl_File, 177
 SEEKCUR
 Oscl_File, 177

SEEKEND
 Oscl_File, 177
 seekFromCurrentPosition
 OsclBinStream, 333
 SEEKSET
 Oscl_File, 177
 self
 Oscl_Map, 213
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
 Oscl_TagTree::const_iterator, 269
 Oscl_TagTree::iterator, 272
 SEM_NOT_SIGNALLED_ERROR
 OsclProcStatus, 466
 Send
 OsclSendMethod, 513
 OsclSendRequest, 514
 OsclSocketI, 527
 OsclSocketIBase, 533
 OsclTCPSocket, 558
 OsclTCPSocketI, 561
 SendParam, 624
 SendParam, 624
 SendParam
 iBufSend, 624
 iFlags, 624
 iXferLen, 624
 SendParam, 624
 SendRequest
 OsclSendMethod, 513
 SendSuccess
 OsclSocketI, 528
 OsclSocketIBase, 533
 SendTo
 OsclSendToMethod, 515
 OsclSendToRequest, 516
 OsclSocketI, 528
 OsclSocketIBase, 533
 OsclUDPSocket, 591
 OsclUDPSocketI, 594
 SendToParam, 625
 SendToParam, 625
 SendToParam
 ~SendToParam, 625
 iAddr, 625
 iBufSend, 625
 iFlags, 625
 iXferLen, 625
 SendToParam, 625
 SendToRequest
 OsclSendToMethod, 515
 SendToSuccess
 OsclSocketI, 528
 OsclSocketIBase, 533

Serv
 OsclDNSRequestAO, 359
 Set
 OsclDoubleRunner, 365
 OsclNameString, 451
 OsclPtr, 467
 OsclPtrC, 470
 set
 CHHeapRep, 128
 CStackRep, 130
 OSCL_FastString, 174, 175
 OSCL_HeapStringA, 198, 199
 OSCL_wFastString, 291
 OSCL_wHeapStringA, 296
 OsclExclusiveArrayPtr, 377
 OsclExclusivePtr, 380
 OsclExclusivePtrA, 383
 OsclSingleton, 522
 OsclTLS, 579
 OsclTLSEx, 581
 osclutil, 80–82
 set_from_ntp_time
 TimeValue, 638
 set_from_system_time
 NTPTime, 166
 set_int64
 Oscl_Int64_Utils, 201
 set_len
 OSCL_String, 258
 OSCL_wString, 302
 set_length
 OSCL_FastString, 175
 OSCL_wFastString, 291
 set_next
 Oscl_Opaque_Type_Alloc_LL, 226
 set_r
 CFastRep, 126
 set_rep
 CHHeapRep, 128
 OSCL_String, 258
 OSCL_wString, 302
 set_to_current_time
 NTPTime, 166
 TimeValue, 638
 set_to_zero
 TimeValue, 638
 set_uint64
 Oscl_Int64_Utils, 201
 set_w
 CFastRep, 126
 setAllocNodeFlag
 MM_AllocBlockHdr, 145
 SetAsyncReadBufferSize
 Oscl_File, 180

SetBusy
 OsclActiveObject, 308
 OsclTimerObject, 575

SetCacheObserver
 Oscl_File, 180

setCheckSum
 StrCSumPtrLen, 630

SetExactFrequency
 OsclTimer, 570

SetFileHandle
 Oscl_File, 181

SetFrequency
 OsclTimer, 571

SetInUse
 OsclAsyncFileBuffer, 315

SetLength
 OsclPtr, 467
 OsclPtrC, 470

SetLoggingEnable
 Oscl_File, 181

SetLogLevel
 PVLogger, 606

SetLogLevelAndPropagate
 PVLogger, 607

setMaxSzForNewMemPoolBuffer
 OsclMemPoolResizableAllocator, 442

SetNativeAccessMode
 Oscl_File, 181

SetNativeBufferSize
 Oscl_File, 181

SetNodeLogLevelExplicit
 PVLoggerRegistry, 614

SetObserver
 OsclTimer, 571

SetOffset
 OsclAsyncFileBuffer, 315
 OsclDoubleListBase, 364

SetParent
 PVLogger, 607

SetPosition
 OsclFileCacheBuffer, 399

SetPriority
 OsclThread, 564

setPtrLen
 StrCSumPtrLen, 630
 StrPtrLen, 633
 WStrPtrLen, 643

SetPVCacheSize
 Oscl_File, 182

SetRecvBufferSize
 OsclIPSocketI, 409
 OsclSocketI, 528
 OsclUDPSocket, 591

setrep_to_char
 OSCL_String, 258
 setrep_to_wide_char
 OSCL_wString, 302

SetScheduler
 OsclExecSchedulerCommonBase, 391

SetStatus
 OsclActiveObject, 308
 OsclTimerObject, 575

SetSummaryStatsLoggingEnable
 Oscl_File, 182

SetTimestamp
 MediaData, 141

SetToHead
 OsclDoubleRunner, 365

SetToTail
 OsclDoubleRunner, 365

setWithoutOwnership
 OSCLMemAutoPtr, 426

ShowStats
 OsclExecSchedulerCommonBase, 391

ShowSummaryStats
 OsclExecSchedulerCommonBase, 391

Shutdown
 OsclShutdownMethod, 520
 OsclShutdownRequest, 521
 OsclSocketI, 528
 OsclSocketIBase, 534
 OsclTCPSocket, 558
 OsclTCPSocketI, 561

ShutdownParam, 626
 ShutdownParam, 626

ShutdownParam
 iHow, 626
 ShutdownParam, 626

ShutdownRequest
 OsclShutdownMethod, 520

Signal
 OsclSemaphore, 512

Size
 Oscl_File, 182
 OsclAsyncFile, 313
 OsclNativeFile, 454

size
 CFastRep, 126
 CHeapRep, 128
 CStackRep, 130
 MM_AllocBlockHdr, 145
 MM_AllocInfo, 147
 MM_AllocQueryInfo, 149
 Oscl_Map, 216
 Oscl_Queue_Base, 234
 Oscl_Rb_Tree, 238
 Oscl_TagTree, 267
 Oscl_Vector_Base, 288

OsclPriorityQueue, [462](#)
 StrPtrLen, [633](#)
 WStrPtrLen, [643](#)
 size_type
 Oscl_Map, [213](#)
 Oscl_Queue, [231](#)
 Oscl_Rb_Tree, [238](#)
 Oscl_Tag_Base, [263](#)
 Oscl_TagTree, [265](#)
 Oscl_TAlloc, [277](#)
 sizeof_T
 Oscl_Linked_List_Base, [210](#)
 Oscl_Queue_Base, [235](#)
 Oscl_Vector_Base, [288](#)
 skip_to_line_term
 osclutil, [82](#)
 skip_to_whitespace
 osclutil, [82](#)
 skip_whitespace
 osclutil, [82](#)
 skip_whitespace_and_line_term
 osclutil, [82](#)
 SLEEP_ONE_SEC
 osclconfig_util.h, [827](#)
 SleepMillisec
 OsclThread, [564](#)
 Socket
 OsclSocketI, [528](#)
 SocketI
 OsclSocketRequestAO, [542](#)
 SocketObserver
 OsclSocketRequestAO, [542](#)
 SocketRequestParam, [627](#)
 SocketRequestParam, [628](#)
 SocketRequestParam
 iFxn, [628](#)
 SocketRequestParam, [628](#)
 SocketServ
 OsclIPSocketI, [409](#)
 sort_children
 Oscl_TagTree::Node, [275](#)
 specialFragBuffer
 OsclBinStream, [334](#)
 Start
 OsclFileStats, [401](#)
 Start_on_creation
 oscl_thread.h, [773](#)
 StartAsyncRead
 OsclAsyncFileBuffer, [315](#)
 StartCancel
 OsclSocketServRequestList, [550](#)
 StartMethod
 OsclIDNSMethod, [354](#)
 OsclSocketMethod, [537](#)
 StartNativeScheduler
 OsclExecSchedulerCommonBase, [391](#)
 StartScheduler
 OsclExecSchedulerCommonBase, [391](#)
 State
 OsclSocketServIBase, [549](#)
 state
 OsclBinStream, [334](#)
 state_t
 OsclBinStream, [332](#)
 StaticJump
 OsclJump, [410](#)
 stats_overhead
 MM_AuditOverheadStats, [158](#)
 Status
 OsclActiveObject, [308](#)
 OsclTimerObject, [576](#)
 status_t
 BuffFragStatusClass, [121](#)
 StatusRef
 OsclActiveObject, [308](#)
 OsclTimerObject, [576](#)
 StopScheduler
 OsclExecSchedulerCommonBase, [391](#)
 Str
 OsclNameString, [451](#)
 StrCSumPtrLen, [629](#)
 osclutil, [67](#)
 StrCSumPtrLen, [630](#)
 StrCSumPtrLen
 checkSum, [630](#)
 CheckSumType, [630](#)
 getCheckSum, [630](#)
 isCIEquivalentTo, [630](#)
 operator!=, [630](#)
 operator=, [630](#)
 operator==, [630](#)
 setCheckSum, [630](#)
 setPtrLen, [630](#)
 StrCSumPtrLen, [630](#)
 StrPtrLen, [632](#)
 osclutil, [67](#)
 StrPtrLen, [633](#)
 StrPtrLen
 c_str, [633](#)
 isCIEquivalentTo, [633](#)
 isCIPrefixOf, [633](#)
 isLetter, [633](#)
 len, [633](#)
 length, [633](#)
 operator!=, [633](#)
 operator=, [633](#)
 operator==, [633](#)
 ptr, [633](#)

setPtrLen, 633
 size, 633
 StrPtrLen, 633
Success
 OsclIDNSRequestAO, 360
 OsclRecvFromRequest, 478
 OsclRecvRequest, 481
 OsclSendRequest, 514
 OsclSendToRequest, 516
 OsclSocketRequestAO, 542
SUCCESS_ERROR
 OsclProcStatus, 465
Suspend
 OsclThread, 565
Suspend_on_creation
 oscl_thread.h, 773
SuspendScheduler
 OsclExecSchedulerCommonBase, 392
swap
 Oscl_Opaque_Type_Compare, 227
 OsclPriorityQueue, 462
SYSTEM_RESOURCES_UNAVAILABLE_-ERROR
 OsclProcStatus, 466

tag
 MM_AllocQueryInfo, 149
 MM_Stats_CB, 160
 Oscl_Tag, 260
 Oscl_TagTree::Node, 275
 OsclMemStatsNode, 448
tag_ancestor
 Oscl_Tag_Base, 263
tag_base_type
 Oscl_Tag_Base, 263
 Oscl_TagTree, 265
tag_base_unit
 Oscl_Tag_Base, 263
tag_cmp
 Oscl_Tag_Base, 263
tag_copy
 Oscl_Tag_Base, 263
tag_depth
 Oscl_Tag_Base, 263
tag_len
 Oscl_Tag_Base, 263
tag_type
 Oscl_TagTree, 265
tagAllocator
 Oscl_Tag, 260
TagTree_Allocator
 osclmemory, 57
Tail
 OsclDoubleList, 362

 OsclPriorityList, 459
tail
 Oscl_Linked_List_Base, 210
takeOwnership
 OSCLMemAutoPtr, 427
TDNSRequestParamAllocator
 oscl_dns_param.h, 659
Tell
 Oscl_File, 182
 OsclAsyncFile, 313
 OsclFileCache, 397
 OsclNativeFile, 454
tellg
 OsclBinStream, 333
Terminate
 OsclThread, 565
the_list
 Oscl_MTLLinked_List, 222
THREAD_1_INACTIVE_ERROR
 OsclProcStatus, 465
THREAD_BLOCK_ERROR
 OsclProcStatus, 466
THREAD_NOT OWN_MUTEX_ERROR
 OsclProcStatus, 466
ThreadHasScheduler
 PVThreadContext, 620
ThreadLogoff
 OsclReadyQ, 475
ThreadLogon
 OsclReadyQ, 475
ThreadPriorityAboveNormal
 oscl_thread.h, 774
ThreadPriorityBelowNormal
 oscl_thread.h, 773
ThreadPriorityHighest
 oscl_thread.h, 774
ThreadPriorityLow
 oscl_thread.h, 773
ThreadPriorityLowest
 oscl_thread.h, 773
ThreadPriorityNormal
 oscl_thread.h, 773
ThreadPriorityTimeCritical
 oscl_thread.h, 774
TickCount
 OsclTickCount, 567
TickCountFrequency
 OsclTickCount, 567
TickCountPeriod
 OsclTickCount, 567
TicksToMsec
 OsclTickCount, 567
TimeoutOccurred
 OsclTimerObserver, 577

TimerBaseElapsed
CallbackTimerObserver, 124
OsclTimer, 571

TimerCallback
OsclReadyQ, 475

timestamp
MediaData, 141

TimeUnits
osclbase, 34

TimeValue, 634
TimeValue, 635, 636

TimeValue
get_local_time, 636
get_pv8601_str_time, 636
get_rfc822_gmtime_str, 636
get_sec, 637
get_str_ctime, 637
get_timeval_ptr, 637
get_usec, 637
is_zero, 637
NTPTime, 639
*operator *=*, 638
operator!=, 639
operator+=, 638
operator-=, 638
operator<, 639
operator<=, 639
operator=, 638
operator==, 639
operator>, 639
operator>=, 639
set_from_ntp_time, 638
set_to_current_time, 638
set_to_zero, 638
TimeValue, 635, 636
to_msec, 638

TLSStorageOps, 640

TLSStorageOps
get_registry, 640
save_registry, 640

to_msec
TimeValue, 638

to_system_time
NTPTime, 166

TOO_MANY_FRAGS
BufFragStatusClass, 121

TOO_MANY_THREADS_ERROR
OsclProcStatus, 465

Top
OsclJump, 410
OsclReadyQ, 475
OsclTimerQ, 578

top
OsclPriorityQueue, 462

TOSCL_StringOp
osclutil, 68

TOSCL_wStringOp
osclutil, 68

TOsclBasicLockObject
osclconfig_unix_android.h, 822
osclconfig_unix_common.h, 826

TOsclConditionObject
osclconfig_proc_unix_android.h, 814
osclconfig_proc_unix_common.h, 816

TOsclFileHandle
osclio, 95

TOsclFileOffset
osclconfig_io.h, 801

TOsclFileOffsetInt32
osclio, 95

TOsclFileOp
osclio, 96

TOsclHostent
osclconfig_io.h, 801

TOsclMutexObject
osclconfig_proc_unix_android.h, 814
osclconfig_proc_unix_common.h, 816

TOsclReady
osclproc, 104

TOsclSemaphoreObject
osclconfig_proc_unix_android.h, 814
osclconfig_proc_unix_common.h, 816

TOsclSockAddr
osclconfig_io.h, 801

TOsclSockAddrLen
osclconfig_io.h, 801

TOsclSocket
osclconfig_io.h, 801

TOsclSocketServStatEvent
oscl_socket_stats.h, 755

TOsclSocketStatEvent
oscl_socket_stats.h, 755

TOsclThreadFuncArg
osclconfig_proc_unix_android.h, 814
osclconfig_proc_unix_common.h, 816

TOsclThreadFuncPtr
oscl_thread.h, 773

TOsclThreadFuncRet
osclconfig_proc_unix_android.h, 814
osclconfig_proc_unix_common.h, 816

TOsclThreadId
osclconfig_proc_unix_android.h, 814
osclconfig_proc_unix_common.h, 816

TOsclThreadObject
osclconfig_proc_unix_android.h, 814
osclconfig_proc_unix_common.h, 816

TOsclTlsKey
osclbase, 33

osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
totalbytes
 oscl_fsstat, 192
totalNumAllocs
 MM_Stats_t, 162
totalNumBytes
 MM_Stats_t, 162
TOtherExecStats
 OsclExecSchedulerCommonBase, 389
TPVDNSEvent
 osclio, 97
TPVDNSFxn
 osclio, 97
TPVSocketEvent
 oscl_socket_types.h, 759
TPVSocketFxn
 oscl_socket_types.h, 759
TPVSocketShutdown
 oscl_socket_types.h, 760
TPVThreadContext
 osclproc, 104
Trap
 OsclErrorTrapImp, 372
TrapNoTls
 OsclErrorTrapImp, 372
TReadyQueLink, 641
 TReadyQueLink, 641
TReadyQueLink
 iAOPriority, 641
 iIsIn, 641
 iSeqNum, 641
 iTimeQueuedTicks, 641
 iTimeToRunTicks, 641
 TReadyQueLink, 641
trim
 OsclMemPoolResizableAllocator, 442
TryLock
 OsclMutex, 450
TryWait
 OsclSemaphore, 512
TSocketServState
 OsclSocketServIBase, 548
TSymbianAccessMode
 Oscl_File, 177

uint
 osclbase, 33
UINT64
 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
uint64
 osclbase, 33
UINT64_HILO

 osclconfig_unix_android.h, 822
 osclconfig_unix_common.h, 826
Unbind
 OsclSharedPtr, 519
UninstallScheduler
 OsclExecSchedulerCommonBase, 392
unix_ntp_offset
 osclbase, 44
Unlock
 OsclLockBase, 413
 OsclMutex, 450
 OsclNullLock, 457
 OsclThreadLock, 566
UnRegister
 OsclRegistryClient, 500
 OsclRegistryClientImpl, 502
 OsclRegistryServTlsImpl, 505
Unregister
 OsclComponentRegistry, 339
UnTrap
 OsclErrorTrapImp, 372
update
 MM_Stats_t, 162
UpdateData
 OsclAsyncFileBuffer, 315
updateEnd
 OsclFileCacheBuffer, 399
updateStart
 OsclFileCacheBuffer, 399
updateStatsNode
 MM_Audit_Imp, 156
updateStatsNodeInFailure
 MM_Audit_Imp, 156
UpdateTimers
 OsclExecSchedulerCommonBase, 392
UpdateTimersMsec
 OsclExecSchedulerCommonBase, 392
upper_bound
 Oscl_Map, 216, 217
 Oscl_Rb_Tree, 238
usableSize
 OsclFileCacheBuffer, 399
USEC_PER_SEC
 osclbase, 44

validate
 MM_Audit_Imp, 156
 OsclPriorityQueue, 463
validate_all_heap
 MM_Audit_Imp, 156
validateblock
 OsclMemPoolResizableAllocator, 442
Value
 OsclAOStatus, 310

value
 Oscl_Rb_Tree_Node, 247
 Oscl_TagTree::Node, 275

value_comp
 Oscl_Map, 217

value_compare
 Oscl_Map::value_compare, 218

value_type
 Oscl_Map, 213
 Oscl_Queue, 231
 Oscl_Rb_Tree, 238
 Oscl_Rb_Tree_Const_Iterator, 242
 Oscl_Rb_Tree_Iterator, 245
 Oscl_Rb_Tree_Node, 247
 Oscl_TagTree, 265
 Oscl_TAlloc, 277
 Oscl_Vector, 281
 OsclPriorityQueue, 461

vec
 OsclPriorityQueue, 463

Wait
 OsclSemaphore, 512

WAIT_ABANDONED_ERROR
 OsclProcStatus, 466

WAIT_TIMEOUT_ERROR
 OsclProcStatus, 466

WaitAndPopTop
 OsclReadyQ, 475

WaitForReadyAO
 OsclExecSchedulerCommonBase, 392

WaitForRequestComplete
 OsclReadyQ, 475

WaitOnRequests
 OsclSocketServRequestList, 550

Wakeup
 OsclSocketServRequestList, 550

writable
 CFastRep, 126

Write
 Oscl_File, 182
 OsclAsyncFile, 313
 OsclFileCache, 397
 OsclNativeFile, 454

write
 OSCL_String, 258
 OSCL_wString, 302
 OsclBinOStream, 326

WriteUnsignedLong
 OsclBinOStreamBigEndian, 328
 OsclBinOStreamLittleEndian, 330

WriteUnsignedShort
 OsclBinOStreamBigEndian, 328
 OsclBinOStreamLittleEndian, 330

WriteUpdatesToFile
 OsclFileCacheBuffer, 399

WStrPtrLen, 642
 osclutil, 67
 WStrPtrLen, 643

WStrPtrLen
 c_str, 643
 isCIEquivalentTo, 643
 len, 643
 length, 643
 operator!=, 643
 operator=, 643
 operator==, 643
 ptr, 643
 setPtrLen, 643
 size, 643
 WStrPtrLen, 643

xsubi
 MM_FailInsertParam, 159

Zero
 OsclPtr, 467
 OsclPtrC, 470