



packetvideoTM

OSCL API

Build Version: OPENCORE_20090120

January 20, 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 Module Documentation	19
5.1 OSCL config	19
5.2 OSCL Base	23
5.3 OSCL Memory	44
5.4 OSCL Util	60
5.5 OSCL Error	81
5.6 OSCL IO	91
5.7 OSCL Proc	99
5.8 OSCL Init	103
6 oscl Data Structure Documentation	104
6.1 _OsclBasicAllocator Class Reference	104
6.2 _OsclHeapBase Class Reference	106
6.3 AcceptParam Class Reference	108
6.4 allocator Class Reference	109
6.5 AllPassFilter Class Reference	110
6.6 BindParam Class Reference	112
6.7 BufferFragment Class Reference	113

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

6.44 Oscl_FileFind Class Reference	181
6.45 Oscl_FileServer Class Reference	185
6.46 oscl_fsstat Struct Reference	187
6.47 OSCL_HeapString< Alloc > Class Template Reference	188
6.48 OSCL_HeapStringA Class Reference	190
6.49 Oscl_Int64_Utils Class Reference	194
6.50 Oscl_Less< T > Struct Template Reference	196
6.51 Oscl_Linked_List< LLClass, Alloc > Class Template Reference	197
6.52 Oscl_Linked_List_Base Class Reference	201
6.53 Oscl_Map< Key, T, Alloc, Compare > Class Template Reference	205
6.54 Oscl_Map< Key, T, Alloc, Compare >::value_compare Class Reference	212
6.55 Oscl_MTLinked_List< LLClass, Alloc, TheLock > Class Template Reference	214
6.56 Oscl_Opaque_Type_Alloc Class Reference	218
6.57 Oscl_Opaque_Type_Alloc_LL Class Reference	219
6.58 Oscl_Opaque_Type_Compare Class Reference	221
6.59 Oscl_Pair< T1, T2 > Struct Template Reference	223
6.60 Oscl_Queue< T, Alloc > Class Template Reference	224
6.61 Oscl_Queue_Base Class Reference	227
6.62 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference	230
6.63 Oscl_Rb_Tree_Base Class Reference	234
6.64 Oscl_Rb_Tree_Const_Iterator< Value > Struct Template Reference	235
6.65 Oscl_Rb_Tree_Iterator< Value > Struct Template Reference	238
6.66 Oscl_Rb_Tree_Node< Value > Struct Template Reference	241
6.67 Oscl_Rb_Tree_Node_Base Struct Reference	242
6.68 Oscl_Select1st< V, U > Struct Template Reference	244
6.69 OSCL_StackString< MaxBufSize > Class Template Reference	245
6.70 oscl_stat_buf Struct Reference	247
6.71 OSCL_String Class Reference	248
6.72 Oscl_Tag< Alloc > Struct Template Reference	253
6.73 Oscl_Tag_Base Struct Reference	255
6.74 Oscl_TagTree< T, Alloc > Class Template Reference	257
6.75 Oscl_TagTree< T, Alloc >::const_iterator Struct Reference	261
6.76 Oscl_TagTree< T, Alloc >::iterator Struct Reference	264
6.77 Oscl_TagTree< T, Alloc >::Node Struct Reference	267
6.78 Oscl_TAlloc< T, Alloc > Class Template Reference	269
6.79 Oscl_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference	272

6.80 Oscl_Vector< T, Alloc > Class Template Reference	273
6.81 Oscl_Vector_Base Class Reference	278
6.82 OSCL_wFastString Class Reference	282
6.83 OSCL_wHeapString< Alloc > Class Template Reference	285
6.84 OSCL_wHeapStringA Class Reference	287
6.85 OSCL_wStackString< MaxBufSize > Class Template Reference	290
6.86 OSCL_wString Class Reference	292
6.87 OsclAcceptMethod Class Reference	296
6.88 OsclAcceptRequest Class Reference	297
6.89 OsclActiveObject Class Reference	298
6.90 OsclAllocDestructDealloc Class Reference	302
6.91 OsclAOStatus Class Reference	303
6.92 OsclAsyncFile Class Reference	304
6.93 OsclAsyncFileBuffer Class Reference	307
6.94 OsclAuditCB Class Reference	309
6.95 OsclBindMethod Class Reference	310
6.96 OsclBindRequest Class Reference	311
6.97 OsclBinIStream Class Reference	312
6.98 OsclBinIStreamBigEndian Class Reference	314
6.99 OsclBinIStreamLittleEndian Class Reference	317
6.100 OsclBinOStream Class Reference	319
6.101 OsclBinOStreamBigEndian Class Reference	320
6.102 OsclBinOStreamLittleEndian Class Reference	322
6.103 OsclBinStream Class Reference	324
6.104 OsclBuf Class Reference	328
6.105 OsclCompareLess< T > Class Template Reference	330
6.106 OsclComponentRegistry Class Reference	331
6.107 OsclComponentRegistryData Class Reference	333
6.108 OsclComponentRegistryElement Class Reference	334
6.109 OsclConnectMethod Class Reference	336
6.110 OsclConnectRequest Class Reference	337
6.111 OsclDestructDealloc Class Reference	338
6.112 OsclDNS Class Reference	339
6.113 OsclDNSI Class Reference	341
6.114 OsclDNSIBase Class Reference	343
6.115 OsclDNSMethod Class Reference	346

6.116OsclDNSObserver Class Reference	349
6.117OsclDNSRequest Class Reference	350
6.118OsclDNSRequestAO Class Reference	351
6.119OsclDoubleLink Class Reference	354
6.120OsclDoubleList< T > Class Template Reference	355
6.121OsclDoubleListBase Class Reference	356
6.122OsclDoubleRunner< T > Class Template Reference	358
6.123OsclError Class Reference	360
6.124OsclErrorAllocator Class Reference	362
6.125OsclErrorTrap Class Reference	364
6.126OsclErrorTrapImp Class Reference	365
6.127OsclException< LeaveCode > Class Template Reference	367
6.128OsclExclusiveArrayPtr< T > Class Template Reference	368
6.129OsclExclusivePtr< T > Class Template Reference	371
6.130OsclExclusivePtrA< T, Alloc > Class Template Reference	374
6.131OsclExecScheduler Class Reference	377
6.132OsclExecSchedulerBase Class Reference	379
6.133OsclExecSchedulerCommonBase Class Reference	380
6.134OsclFileCache Class Reference	389
6.135OsclFileHandle Class Reference	391
6.136OsclFileStats Class Reference	392
6.137OsclFileStatsItem Class Reference	393
6.138OsclGetHostByNameMethod Class Reference	394
6.139OsclGetHostByNameRequest Class Reference	395
6.140OsclInit Class Reference	396
6.141OsclInteger64Transport Struct Reference	397
6.142OsclIPSocketI Class Reference	398
6.143OsclJump Class Reference	401
6.144OsclListenMethod Class Reference	402
6.145OsclListenRequest Class Reference	403
6.146OsclLockBase Class Reference	404
6.147OsclMem Class Reference	405
6.148OsclMemAllocator Class Reference	406
6.149OsclMemAllocDestructDealloc< T > Class Template Reference	407
6.150OsclMemAudit Class Reference	409
6.151OSCLMemAutoPtr< T, _Allocator > Class Template Reference	415

6.152OsclMemBasicAllocator Class Reference	419
6.153OsclMemBasicAllocDestructDealloc< T > Class Template Reference	420
6.154OsclMemGlobalAuditObject Class Reference	421
6.155OsclMemoryFragment Struct Reference	422
6.156OsclMemPoolAllocator Class Reference	423
6.157OsclMemPoolFixedChunkAllocator Class Reference	424
6.158OsclMemPoolFixedChunkAllocatorObserver Class Reference	428
6.159OsclMemPoolResizableAllocator Class Reference	429
6.160OsclMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference	435
6.161OsclMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference	436
6.162OsclMemPoolResizableAllocatorMemoryObserver Class Reference	437
6.163OsclMemPoolResizableAllocatorObserver Class Reference	438
6.164OsclMemStatsNode Class Reference	439
6.165OsclMutex Class Reference	440
6.166OsclNameString< __len > Class Template Reference	442
6.167OsclNativeFile Class Reference	443
6.168OsclNativeFileParams Class Reference	446
6.169OsclNetworkAddress Class Reference	447
6.170OsclNullLock Class Reference	448
6.171OsclPriorityLink Class Reference	449
6.172OsclPriorityList< T > Class Template Reference	450
6.173OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference	451
6.174OsclPriorityQueueBase Class Reference	455
6.175OsclProcStatus Class Reference	456
6.176OsclPtr Class Reference	458
6.177OsclPtrC Class Reference	460
6.178OsclRand Class Reference	462
6.179OsclReadyAlloc Class Reference	463
6.180OsclReadyCompare Class Reference	464
6.181OsclReadyQ Class Reference	465
6.182OsclRecvFromMethod Class Reference	467
6.183OsclRecvFromRequest Class Reference	469
6.184OsclRecvMethod Class Reference	471
6.185OsclRecvRequest Class Reference	472
6.186OsclRefCounter Class Reference	473
6.187OsclRefCounterDA Class Reference	475

6.188OsclRefCounterMemFrag Class Reference	477
6.189OsclRefCounterMTDA< LockType > Class Template Reference	479
6.190OsclRefCounterMTSA< DeallocType, LockType > Class Template Reference	481
6.191OsclRefCounterSA< DeallocType > Class Template Reference	483
6.192OsclRegistryAccessClient Class Reference	485
6.193OsclRegistryAccessClientImpl Class Reference	487
6.194OsclRegistryAccessClientTlsImpl Class Reference	488
6.195OsclRegistryAccessElement Class Reference	489
6.196OsclRegistryClient Class Reference	490
6.197OsclRegistryClientImpl Class Reference	492
6.198OsclRegistryClientTlsImpl Class Reference	494
6.199OsclRegistryServTlsImpl Class Reference	495
6.200OsclScheduler Class Reference	497
6.201OsclSchedulerObserver Class Reference	498
6.202OsclScopedLock< LockClass > Class Template Reference	499
6.203OsclSelect Class Reference	500
6.204OsclSemaphore Class Reference	502
6.205OsclSendMethod Class Reference	504
6.206OsclSendRequest Class Reference	505
6.207OsclSendToMethod Class Reference	506
6.208OsclSendToRequest Class Reference	507
6.209OsclSharedPtr< TheClass > Class Template Reference	508
6.210OsclShutdownMethod Class Reference	511
6.211OsclShutdownRequest Class Reference	512
6.212OsclSingleton< T, ID, Registry > Class Template Reference	513
6.213OsclSingletonRegistry Class Reference	515
6.214OsclSocketI Class Reference	516
6.215OsclSocketIBase Class Reference	521
6.216OsclSocketMethod Class Reference	526
6.217OsclSocketObserver Class Reference	529
6.218OsclSocketRequest Class Reference	530
6.219OsclSocketRequestAO Class Reference	531
6.220OsclSocketServ Class Reference	535
6.221OsclSocketServI Class Reference	537
6.222OsclSocketServIBase Class Reference	539
6.223OsclSocketServRequestList Class Reference	541

6.224OsclSocketServRequestQElem Class Reference	543
6.225OsclTCPSocket Class Reference	544
6.226OsclTCPSocketI Class Reference	550
6.227OsclThread Class Reference	553
6.228OsclThreadLock Class Reference	557
6.229OsclTickCount Class Reference	558
6.230OsclTimer< Alloc > Class Template Reference	560
6.231OsclTimerCompare Class Reference	563
6.232OsclTimerObject Class Reference	564
6.233OsclTimerObserver Class Reference	568
6.234OsclTimerQ Class Reference	569
6.235OsclTLS< T, ID, Registry > Class Template Reference	570
6.236OsclTLSE< T, ID, Registry > Class Template Reference	572
6.237OsclTLSRegistry Class Reference	574
6.238OsclTLSRegistryEx Class Reference	575
6.239OsclTrapItem Class Reference	576
6.240OsclTrapStack Class Reference	577
6.241OsclTrapStackItem Class Reference	578
6.242OsclUDPSocket Class Reference	579
6.243OsclUDPSocketI Class Reference	584
6.244OsclUuid Struct Reference	586
6.245PVActiveBase Class Reference	588
6.246PVActiveStats Class Reference	592
6.247PVLogger Class Reference	593
6.248PVLoggerAppender Class Reference	599
6.249PVLoggerFilter Class Reference	600
6.250PVLoggerLayout Class Reference	602
6.251PVLoggerRegistry Class Reference	604
6.252PVSchedulerStopper Class Reference	607
6.253PVSockBufRecv Class Reference	608
6.254PVSockBufSend Class Reference	609
6.255PVThreadContext Class Reference	610
6.256RecvFromParam Class Reference	612
6.257RecvParam Class Reference	614
6.258SendParam Class Reference	615
6.259SendToParam Class Reference	616

6.260	ShutdownParam Class Reference	617
6.261	SocketRequestParam Class Reference	618
6.262	StrCSumPtrLen Struct Reference	620
6.263	StrPtrLen Struct Reference	623
6.264	TimeValue Class Reference	625
6.265	TLSStorageOps Class Reference	631
6.266	TReadyQueLink Class Reference	632
6.267	WStrPtrLen Struct Reference	633
7	oscl File Documentation	635
7.1	oscl_aostatus.h File Reference	635
7.2	oscl_assert.h File Reference	636
7.3	oscl_base.h File Reference	637
7.4	oscl_base_alloc.h File Reference	638
7.5	oscl_base_macros.h File Reference	639
7.6	oscl_bin_stream.h File Reference	640
7.7	oscl_byte_order.h File Reference	641
7.8	oscl_defalloc.h File Reference	642
7.9	oscl_dll.h File Reference	643
7.10	oscl_dns.h File Reference	644
7.11	oscl_dns_gethostname.h File Reference	645
7.12	oscl_dns_imp.h File Reference	646
7.13	oscl_dns_imp_base.h File Reference	647
7.14	oscl_dns_imp_pv.h File Reference	648
7.15	oscl_dns_method.h File Reference	649
7.16	oscl_dns_param.h File Reference	650
7.17	oscl_dns_request.h File Reference	651
7.18	oscl_dns_tuneables.h File Reference	652
7.19	oscl_double_list.h File Reference	653
7.20	oscl_errno.h File Reference	654
7.21	oscl_error.h File Reference	655
7.22	oscl_error_allocator.h File Reference	656
7.23	oscl_error_codes.h File Reference	657
7.24	oscl_error_imp.h File Reference	658
7.25	oscl_error_imp_cppexceptions.h File Reference	659
7.26	oscl_error_imp_fatalerror.h File Reference	660
7.27	oscl_error_imp_jumps.h File Reference	661

7.28 oscl_error_trapcleanup.h File Reference	663
7.29 oscl_exception.h File Reference	664
7.30 oscl_exclusive_ptr.h File Reference	665
7.31 oscl_file_async_read.h File Reference	666
7.32 oscl_file_cache.h File Reference	667
7.33 oscl_file_dir_utils.h File Reference	668
7.34 oscl_file_find.h File Reference	670
7.35 oscl_file_handle.h File Reference	671
7.36 oscl_file_io.h File Reference	672
7.37 oscl_file_native.h File Reference	673
7.38 oscl_file_server.h File Reference	674
7.39 oscl_file_stats.h File Reference	675
7.40 oscl_file_types.h File Reference	676
7.41 oscl_heapbase.h File Reference	677
7.42 oscl_init.h File Reference	678
7.43 oscl_int64_utils.h File Reference	679
7.44 oscl_ip_socket.h File Reference	680
7.45 oscl_linked_list.h File Reference	681
7.46 oscl_lock_base.h File Reference	682
7.47 oscl_map.h File Reference	683
7.48 oscl_math.h File Reference	684
7.49 oscl_media_data.h File Reference	685
7.50 oscl_media_status.h File Reference	686
7.51 oscl_mem.h File Reference	687
7.52 oscl_mem_align.h File Reference	690
7.53 oscl_mem_audit.h File Reference	691
7.54 oscl_mem_audit_internals.h File Reference	693
7.55 oscl_mem_auto_ptr.h File Reference	694
7.56 oscl_mem_basic_functions.h File Reference	695
7.57 oscl_mem_inst.h File Reference	696
7.58 oscl_mem_mempool.h File Reference	697
7.59 oscl_mempool_allocator.h File Reference	698
7.60 oscl_mutex.h File Reference	699
7.61 oscl_namestring.h File Reference	700
7.62 oscl_opaque_type.h File Reference	701
7.63 oscl_pqueue.h File Reference	702

7.64 oscl_proctstatus.h File Reference	703
7.65 oscl_queue.h File Reference	704
7.66 oscl_rand.h File Reference	705
7.67 oscl_refcounter.h File Reference	706
7.68 oscl_refcounter_memfrag.h File Reference	707
7.69 oscl_registry_access_client.h File Reference	708
7.70 oscl_registry_client.h File Reference	709
7.71 oscl_registry_client_impl.h File Reference	710
7.72 oscl_registry_serv_impl.h File Reference	711
7.73 oscl_registry_serv_impl_global.h File Reference	712
7.74 oscl_registry_serv_impl_tls.h File Reference	713
7.75 oscl_registry_types.h File Reference	714
7.76 oscl_scheduler.h File Reference	715
7.77 oscl_scheduler_ao.h File Reference	716
7.78 oscl_scheduler_aobase.h File Reference	717
7.79 oscl_scheduler_readyq.h File Reference	718
7.80 oscl_scheduler_threadcontext.h File Reference	719
7.81 oscl_scheduler_tuneables.h File Reference	720
7.82 oscl_scheduler_types.h File Reference	721
7.83 oscl_semaphore.h File Reference	722
7.84 oscl_shared_ptr.h File Reference	723
7.85 oscl_singleton.h File Reference	724
7.86 oscl_snprintf.h File Reference	726
7.87 oscl_socket.h File Reference	727
7.88 oscl_socket_accept.h File Reference	728
7.89 oscl_socket_bind.h File Reference	729
7.90 oscl_socket_connect.h File Reference	730
7.91 oscl_socket_imp.h File Reference	731
7.92 oscl_socket_imp_base.h File Reference	732
7.93 oscl_socket_imp_pv.h File Reference	733
7.94 oscl_socket_listen.h File Reference	734
7.95 oscl_socket_method.h File Reference	735
7.96 oscl_socket_recv.h File Reference	736
7.97 oscl_socket_recv_from.h File Reference	737
7.98 oscl_socket_request.h File Reference	738
7.99 oscl_socket_send.h File Reference	739

7.100oscl_socket_send_to.h File Reference	740
7.101oscl_socket_serv_imp.h File Reference	741
7.102oscl_socket_serv_imp_base.h File Reference	742
7.103oscl_socket_serv_imp_pv.h File Reference	743
7.104oscl_socket_serv_imp_reqlist.h File Reference	744
7.105oscl_socket_shutdown.h File Reference	745
7.106oscl_socket_stats.h File Reference	746
7.107oscl_socket_tuneables.h File Reference	748
7.108oscl_socket_types.h File Reference	750
7.109oscl_stdstring.h File Reference	752
7.110oscl_str_ptr_len.h File Reference	753
7.111oscl_string.h File Reference	754
7.112oscl_string_containers.h File Reference	755
7.113oscl_string_rep.h File Reference	756
7.114oscl_string_uri.h File Reference	757
7.115oscl_string_utf8.h File Reference	758
7.116oscl_string_utils.h File Reference	759
7.117oscl_string_xml.h File Reference	760
7.118oscl_tagtree.h File Reference	761
7.119oscl_tcp_socket.h File Reference	762
7.120oscl_thread.h File Reference	763
7.121oscl_tickcount.h File Reference	765
7.122oscl_time.h File Reference	766
7.123oscl_timer.h File Reference	768
7.124oscl_tls.h File Reference	769
7.125oscl_tree.h File Reference	770
7.126oscl_types.h File Reference	771
7.127oscl_udp_socket.h File Reference	772
7.128oscl_utf8conv.h File Reference	773
7.129oscl_uuid.h File Reference	774
7.130oscl_vector.h File Reference	775
7.131osclconfig.h File Reference	776
7.132osclconfig_ansi_memory.h File Reference	778
7.133osclconfig_check.h File Reference	779
7.134osclconfig_compiler_warnings.h File Reference	780
7.135osclconfig_error.h File Reference	781

7.136osclconfig_error_check.h File Reference	782
7.137osclconfig_global_new_delete.h File Reference	783
7.138osclconfig_global_placement_new.h File Reference	784
7.139osclconfig_io.h File Reference	785
7.140osclconfig_io_check.h File Reference	792
7.141osclconfig_ix86.h File Reference	793
7.142osclconfig_lib.h File Reference	794
7.143osclconfig_lib_check.h File Reference	795
7.144osclconfig_limits_typedefs.h File Reference	796
7.145osclconfig_memory.h File Reference	797
7.146osclconfig_memory_check.h File Reference	798
7.147osclconfig_no_os.h File Reference	799
7.148osclconfig_proc.h File Reference	800
7.149osclconfig_proc_check.h File Reference	801
7.150osclconfig_proc_unix_common.h File Reference	803
7.151osclconfig_proc_unix_nj.h File Reference	805
7.152osclconfig_time.h File Reference	807
7.153osclconfig_time_check.h File Reference	808
7.154osclconfig_unix_common.h File Reference	809
7.155osclconfig_unix_nj.h File Reference	813
7.156osclconfig_util.h File Reference	817
7.157osclconfig_util_check.h File Reference	818
7.158pvlogger.h File Reference	819
7.159pvlogger_accessories.h File Reference	827
7.160pvlogger_c.h File Reference	828
7.161pvlogger_registry.h File Reference	830

Chapter 1

oscl Module Index

1.1 oscl Modules

Here is a list of all modules:

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

Chapter 2

oscl Hierarchical Index

2.1 oscl Class Hierarchy

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

_OscIHeapBase	106
HeapBase	132
Oscl_File	174
OSCL_String	248
OSCL_FastString	170
OSCL_HeapString< Alloc >	188
OSCL_HeapStringA	190
OSCL_StackString< MaxBufSize >	245
OsclActiveObject	298
OsclAsyncFile	304
OsclDNSRequestAO	351
OsclGetHostByNameRequest	395
OsclSocketRequestAO	531
OsclAcceptRequest	297
OsclBindRequest	311
OsclConnectRequest	337
OsclListenRequest	403
OsclRecvFromRequest	469
OsclRecvRequest	472
OsclSendRequest	505
OsclSendToRequest	507
OsclShutdownRequest	512
PVSchedulerStopper	607
OsclAsyncFileBuffer	307
OsclBuf	328
OsclDNS	339
OsclFileCache	389
OsclNativeFile	443
OsclPtr	458
OsclPtrC	460
OsclRegistryClient	490
OsclSocketServ	535
OsclTCPSocket	544

OsclTimerObject	564
CallbackTimer< Alloc >	120
OsclDNSMethod	346
OsclGetHostByNameMethod	394
OsclSocketMethod	526
OsclAcceptMethod	296
OsclBindMethod	310
OsclConnectMethod	336
OsclListenMethod	402
OsclRecvFromMethod	467
OsclRecvMethod	471
OsclSendMethod	504
OsclSendToMethod	506
OsclShutdownMethod	511
OsclSocketServI	537
OsclUDPSocket	579
OsclExecSchedulerBase	379
OsclExecScheduler	377
allocator	109
BufferMgr	114
BufferState	115
BuFragGroup< ChainClass, max_frags >	116
MediaData< ChainClass, max_frags, local_bufsize >	137
BufFragStatusClass	119
MediaStatusClass	140
CallbackTimerObserver	122
OsclTimer< Alloc >	560
CFastRep	123
CHheapRep	125
CStackRep	128
DNSRequestParam	129
GetHostByNameParam	131
internalLeave	134
LinkedListElement< LLClass >	135
MemAllocator< T >	141
MM_AllocBlockFence	142
MM_AllocBlockHdr	143
MM_AllocInfo	144
MM_AllocNode	146
MM_AllocQueryInfo	147
MM_Audit_Imp	148
MM_AuditOverheadStats	156
MM_FailInsertParam	157
MM_Stats_CB	158
MM_Stats_t	159
NTPTime	161
Oscl_Alloc	165
Oscl_DefAlloc	167
_OsclBasicAllocator	104
OsclAllocDestructDealloc	302
OsclMemAllocDestructDealloc< T >	407
OsclMemBasicAllocDestructDealloc< T >	420

OsclMemAllocator	406
OsclMemBasicAllocator	419
OsclMemPoolFixedChunkAllocator	424
OsclMemPoolResizableAllocator	429
OsclReadyAlloc	463
Oscl_Dealloc	166
Oscl_DefAlloc	167
Oscl_FileFind	181
Oscl_FileServer	185
oscl_fsstat	187
Oscl_Int64_Utils	194
Oscl_Less< T >	196
Oscl_Linked_List_Base	201
Oscl_Linked_List< LLClass, Alloc >	197
Oscl_Map< Key, T, Alloc, Compare >	205
Oscl_Map< Key, T, Alloc, Compare >::value_compare	212
Oscl_MTLinked_List< LLClass, Alloc, TheLock >	214
Oscl_Opaque_Type_Alloc	218
Oscl_Queue< T, Alloc >	224
Oscl_Vector< T, Alloc >	273
Oscl_Vector< TOsclReady, OsclReadyAlloc >	273
Oscl_Opaque_Type_Alloc_LL	219
Oscl_Linked_List< LLClass, Alloc >	197
Oscl_Opaque_Type_Compare	221
OsclPriorityQueue< Qelem, Alloc, Container, Compare >	451
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclReadyCompare >	451
OsclReadyQ	465
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclTimerCompare >	451
OsclTimerQ	569
Oscl_Pair< T1, T2 >	223
Oscl_Queue_Base	227
Oscl_Queue< T, Alloc >	224
Oscl_Rb_Tree_Base	234
Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >	230
Oscl_Rb_Tree_Const_Iterator< Value >	235
Oscl_Rb_Tree_Iterator< Value >	238
Oscl_Rb_Tree_Node_Base	242
Oscl_Rb_Tree_Node< Value >	241
Oscl_Select1st< V, U >	244
oscl_stat_buf	247
Oscl_Tag_Base	255
Oscl_Tag< Alloc >	253
Oscl_TagTree< T, Alloc >	257
Oscl_TagTree< T, Alloc >::const_iterator	261
Oscl_TagTree< T, Alloc >::iterator	264
Oscl_TagTree< T, Alloc >::Node	267
Oscl_TAlloc< T, Alloc >::rebind< U, V >	272
Oscl_Vector_Base	278

Oscl_Vector< T, Alloc >	273
Oscl_Vector< TOsclReady, OsclReadyAlloc >	273
OSCL_wString	292
OSCL_wFastString	282
OSCL_wHeapString< Alloc >	285
OSCL_wHeapStringA	287
OSCL_wStackString< MaxBufSize >	290
OsclAOStatus	303
OsclAuditCB	309
OsclBinStream	324
OsclBinIStream	312
OsclBinIStreamBigEndian	314
OsclBinIStreamLittleEndian	317
OsclBinOStream	319
OsclBinOStreamBigEndian	320
OsclBinOStreamLittleEndian	322
OsclCompareLess< T >	330
OsclComponentRegistry	331
OsclComponentRegistryData	333
OsclComponentRegistryElement	334
OsclDestructDealloc	338
Oscl_TAlloc< T, Alloc >	269
OsclAllocDestructDealloc	302
OsclDNSIBase	343
OsclDNSI	341
OsclDNSObserver	349
OsclDNSRequest	350
OsclDoubleLink	354
OsclPriorityLink	449
OsclDoubleListBase	356
OsclDoubleList< T >	355
OsclPriorityList< T >	450
OsclDoubleRunner< T >	358
OsclError	360
OsclErrorAllocator	362
OsclErrorTrap	364
OsclErrorTrapImp	365
OsclException< LeaveCode >	367
OsclExclusiveArrayPtr< T >	368
OsclExclusivePtr< T >	371
OsclExclusivePtrA< T, Alloc >	374
OsclExecSchedulerCommonBase	380
OsclExecScheduler	377
OsclFileHandle	391
OsclFileStats	392
OsclFileStatsItem	393
OsclInit	396
OsclInteger64Transport	397
OsclIPSocketI	398
OsclTCPSocketI	550
OsclUDPSocketI	584

OsclJump	401
OsclLockBase	404
OsclMutex	440
OsclNullLock	448
OsclThreadLock	557
OsclMem	405
OsclMemAudit	409
OSCLMemAutoPtr< T, _Allocator >	415
OsclMemGlobalAuditObject	421
OsclMemoryFragment	422
BufferFragment	113
OsclMemPoolAllocator	423
OsclMemPoolFixedChunkAllocatorObserver	428
OsclMemPoolResizableAllocator::MemPoolBlockInfo	435
OsclMemPoolResizableAllocator::MemPoolBufferInfo	436
OsclMemPoolResizableAllocatorMemoryObserver	437
OsclMemPoolResizableAllocatorObserver	438
OsclMemStatsNode	439
OsclNameString< __len >	442
OsclNativeFileParams	446
OsclNetworkAddress	447
OsclPriorityQueueBase	455
OsclPriorityQueue< Qelem, Alloc, Container, Compare >	451
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady-Alloc >, OsclReadyCompare >	451
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReady-Alloc >, OsclTimerCompare >	451
OsclProcStatus	456
OsclRand	462
OsclReadyCompare	464
OsclRefCounter	473
Oscl_DefAllocWithRefCounter< DefAlloc >	168
OsclRefCounterDA	475
OsclRefCounterMTDA< LockType >	479
OsclRefCounterMTSA< DeallocType, LockType >	481
OsclRefCounterSA< DeallocType >	483
OsclRefCounterMemFrag	477
OsclRegistryAccessClient	485
OsclRegistryAccessElement	489
OsclRegistryClientImpl	492
OsclRegistryAccessClientImpl	487
OsclRegistryServTlsImpl	495
OsclRegistryAccessClientTlsImpl	488
OsclRegistryClientTlsImpl	494
OsclScheduler	497
OsclSchedulerObserver	498
OsclScopedLock< LockClass >	499
OsclSelect	500
OsclSemaphore	502
OsclSharedPtr< TheClass >	508
OsclSingleton< T, ID, Registry >	513
OsclSingletonRegistry	515

OsclSocketIBase	521
OsclSocketI	516
OsclSocketObserver	529
OsclSocketRequest	530
OsclSocketServIBase	539
OsclSocketServI	537
OsclSocketServRequestList	541
OsclSocketServRequestQElem	543
OsclThread	553
OsclTickCount	558
OsclTimerCompare	563
OsclTimerObserver	568
OsclTLS< T, ID, Registry >	570
OsclTLSE< T, ID, Registry >	572
OsclTLSRegistry	574
OsclTLSRegistryEx	575
OsclTrapItem	576
OsclTrapStack	577
OsclTrapStackItem	578
OsclUuid	586
PVActiveBase	588
OsclActiveObject	298
OsclTimerObject	564
PVActiveStats	592
PVLogger	593
PVLoggerAppender	599
PVLoggerFilter	600
AllPassFilter	110
PVLoggerLayout	602
PVLoggerRegistry	604
PVSockBufRecv	608
PVSockBufSend	609
PVThreadContext	610
SocketRequestParam	618
AcceptParam	108
BindParam	112
ConnectParam	127
ListenParam	136
RecvFromParam	612
RecvParam	614
SendParam	615
SendToParam	616
ShutdownParam	617
StrPtrLen	623
StrCSumPtrLen	620
TimeValue	625
TLSStorageOps	631
TReadyQueLink	632
WStrPtrLen	633

Chapter 3

oscl Data Structure Index

3.1 oscl Data Structures

Here are the data structures with brief descriptions:

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

MM_FailInsertParam	157
MM_Stats_CB	158
MM_Stats_t	159
NTPTime (Time value as the number of seconds since 0h (UTC) Jan. 1, 1900)	161
OscI_Alloc	165
OscI_Dealloc	166
OscI_DefAlloc	167
OscI_DefAllocWithRefCounter< DefAlloc >	168
OSCL_FastString	170
OscI_File	174
OscI_FileFind	181
OscI_FileServer	185
oscl_fstat	187
OSCL_HeapString< Alloc >	188
OSCL_HeapStringA	190
OscI_Int64_Utils (Wrapper for commonly used int64/uint64 operations)	194
OscI_Less< T >	196
OscI_Linked_List< LLClass, Alloc >	197
OscI_Linked_List_Base	201
OscI_Map< Key, T, Alloc, Compare >	205
OscI_Map< Key, T, Alloc, Compare >::value_compare	212
OscI_MTLinked_List< LLClass, Alloc, TheLock >	214
OscI_Opaque_Type_Alloc	218
OscI_Opaque_Type_Alloc_LL	219
OscI_Opaque_Type_Compare	221
OscI_Pair< T1, T2 >	223
OscI_Queue< T, Alloc >	224
OscI_Queue_Base	227
OscI_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >	230
OscI_Rb_Tree_Base	234
OscI_Rb_Tree_Const_Iterator< Value >	235
OscI_Rb_Tree_Iterator< Value >	238
OscI_Rb_Tree_Node< Value >	241
OscI_Rb_Tree_Node_Base	242
OscI_Select1st< V, U >	244
OSCL_StackString< MaxBufSize >	245
oscl_stat_buf	247
OSCL_String	248
OscI_Tag< Alloc >	253
OscI_Tag_Base	255
OscI_TagTree< T, Alloc >	257
OscI_TagTree< T, Alloc >::const_iterator	261
OscI_TagTree< T, Alloc >::iterator	264
OscI_TagTree< T, Alloc >::Node	267
OscI_TAlloc< T, Alloc >	269
OscI_TAlloc< T, Alloc >::rebind< U, V >	272
OscI_Vector< T, Alloc >	273
OscI_Vector_Base	278
OSCL_wFastString	282
OSCL_wHeapString< Alloc >	285
OSCL_wHeapStringA	287
OSCL_wStackString< MaxBufSize >	290
OSCL_wString	292
OscIAcceptMethod	296

OsclAcceptRequest	297
OsclActiveObject	298
OsclAllocDestructDealloc	302
OsclAOStatus	303
OsclAsyncFile	304
OsclAsyncFileBuffer	307
OsclAuditCB	309
OsclBindMethod	310
OsclBindRequest	311
OsclBinIStream	312
OsclBinIStreamBigEndian	314
OsclBinIStreamLittleEndian	317
OsclBinOStream (Class OsclBinOStream implements the basic stream functions for an output stream)	319
OsclBinOStreamBigEndian (Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering)	320
OsclBinOStreamLittleEndian (Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering)	322
OsclBinStream	324
OsclBuf	328
OsclCompareLess< T >	330
OsclComponentRegistry	331
OsclComponentRegistryData	333
OsclComponentRegistryElement	334
OsclConnectMethod	336
OsclConnectRequest	337
OsclDestructDealloc	338
OsclDNS	339
OsclDNSI	341
OsclDNSIBase	343
OsclDNSMethod	346
OsclDNSObserver	349
OsclDNSRequest	350
OsclDNSRequestAO	351
OsclDoubleLink	354
OsclDoubleList< T >	355
OsclDoubleListBase	356
OsclDoubleRunner< T >	358
OsclError	360
OsclErrorAllocator (This class provides static methods to invoke the user defined memory allocation routines)	362
OsclErrorTrap	364
OsclErrorTrapImp	365
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)	367
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)	368
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)	371

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)	374
OsclExecScheduler	377
OsclExecSchedulerBase	379
OsclExecSchedulerCommonBase	380
OsclFileCache	389
OsclHandle	391
OsclFileStats	392
OsclFileStatsItem	393
OsclGetHostNameMethod	394
OsclGetHostNameRequest	395
OsclInit	396
OsclInteger64Transport	397
OsclIPSocketI	398
OsclJump	401
OsclListenMethod	402
OsclListenRequest	403
OsclLockBase	404
OsclMem	405
OsclMemAllocator	406
OsclMemAllocDestructDealloc< T >	407
OsclMemAudit	409
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)	415
OsclMemBasicAllocator	419
OsclMemBasicAllocDestructDealloc< T >	420
OsclMemGlobalAuditObject	421
OsclMemoryFragment	422
OsclMemPoolAllocator	423
OsclMemPoolFixedChunkAllocator	424
OsclMemPoolFixedChunkAllocatorObserver	428
OsclMemPoolResizableAllocator	429
OsclMemPoolResizableAllocator::MemPoolBlockInfo	435
OsclMemPoolResizableAllocator::MemPoolBufferInfo	436
OsclMemPoolResizableAllocatorMemoryObserver	437
OsclMemPoolResizableAllocatorObserver	438
OsclMemStatsNode	439
OsclMutex	440
OsclNameString< __len >	442
OsclNativeFile	443
OsclNativeFileParams	446
OsclNetworkAddress	447
OsclNullLock	448
OsclPriorityLink	449
OsclPriorityList< T >	450
OsclPriorityQueue< Qelem, Alloc, Container, Compare >	451
OsclPriorityQueueBase	455
OsclProcStatus	456
OsclPtr	458
OsclPtrC	460
OsclRand	462

OsclReadyAlloc	463
OsclReadyCompare	464
OsclReadyQ	465
OsclRecvFromMethod	467
OsclRecvFromRequest	469
OsclRecvMethod	471
OsclRecvRequest	472
OsclRefCounter	473
OsclRefCounterDA	475
OsclRefCounterMemFrag	477
OsclRefCounterMTDA< LockType >	479
OsclRefCounterMTSA< DeallocType, LockType >	481
OsclRefCounterSA< DeallocType >	483
OsclRegistryAccessClient	485
OsclRegistryAccessClientImpl	487
OsclRegistryAccessClientTlsImpl	488
OsclRegistryAccessElement	489
OsclRegistryClient	490
OsclRegistryClientImpl	492
OsclRegistryClientTlsImpl	494
OsclRegistryServTlsImpl	495
OsclScheduler	497
OsclSchedulerObserver	498
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)	499
OsclSelect	500
OsclSemaphore	502
OsclSendMethod	504
OsclSendRequest	505
OsclSendToMethod	506
OsclSendToRequest	507
OsclSharedPtr< TheClass > (A parameterized smart pointer class)	508
OsclShutdownMethod	511
OsclShutdownRequest	512
OsclSingleton< T, ID, Registry >	513
OsclSingletonRegistry	515
OsclSocketI	516
OsclSocketIBase	521
OsclSocketMethod	526
OsclSocketObserver	529
OsclSocketRequest	530
OsclSocketRequestAO	531
OsclSocketServ	535
OsclSocketServI	537
OsclSocketServIBase	539
OsclSocketServRequestList	541
OsclSocketServRequestQELEM	543
OsclTCPSocket	544
OsclTCPSocketI	550
OsclThread	553
OsclThreadLock	557
OsclTickCount	558
OsclTimer< Alloc >	560

OsclTimerCompare	563
OsclTimerObject	564
OsclTimerObserver	568
OsclTimerQ	569
OsclTLS< T, ID, Registry >	570
OsclTLSE< T, ID, Registry >	572
OsclTLSRegistry	574
OsclTLSRegistryEx	575
OsclTrapItem	576
OsclTrapStack	577
OsclTrapStackItem	578
OsclUDPSocket	579
OsclUDPSocketI	584
OsclUuid	586
PVActiveBase	588
PVActiveStats	592
PVLogger	593
PVLoggerAppender	599
PVLoggerFilter	600
PVLoggerLayout	602
PVLoggerRegistry	604
PVSchedulerStopper	607
PVSockBufRecv	608
PVSockBufSend	609
PVThreadContext	610
RecvFromParam	612
RecvParam	614
SendParam	615
SendToParam	616
ShutdownParam	617
SocketRequestParam	618
StrCSumPtrLen (Same as StrPtrLen, but includes checksum field and method to speed up querying)	620
StrPtrLen (This data structure encapsulates a set of functions used to perform)	623
TimeValue (Time value in a format native to the system)	625
TLSStorageOps	631
TReadyQueLink	632
WStrPtrLen (This data structure encapsulates a set of functions used to perform)	633

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)	635
<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)	636
<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)	637
<code>oscl_base_alloc.h</code> (A basic allocator that does not rely on other modules)	638
<code>oscl_base_macros.h</code> (This file defines common macros and constants for basic compilation support)	639
<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)	640
<code>oscl_byte_order.h</code> (This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders))	641
<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)	642
<code>oscl_dll.h</code> (Defines a DLL entry point)	643
<code>oscl_dns.h</code> (The file <code>oscl_socket.h</code> defines the OSCL DNS APIs)	644
<code>oscl_dns_gethostbyname.h</code>	645
<code>oscl_dns_imp.h</code>	646
<code>oscl_dns_imp_base.h</code>	647
<code>oscl_dns_imp_pv.h</code>	648
<code>oscl_dns_method.h</code>	649
<code>oscl_dns_param.h</code>	650
<code>oscl_dns_request.h</code>	651
<code>oscl_dns_tuneables.h</code>	652
<code>oscl_double_list.h</code> (Internal use types for scheduler)	653
<code>oscl_errno.h</code> (Defines functions to access additional information on errors where supported through an errno or similar service)	654
<code>oscl_error.h</code> (OSCL Error trap and cleanup include file)	655
<code>oscl_error_allocator.h</code> (Defines a memory allocation class used by the oscl error layer)	656
<code>oscl_error_codes.h</code> (Defines basic error and leave codes)	657
<code>oscl_error_imp.h</code> (Internal error implementation support)	658
<code>oscl_error_imp_cppexceptions.h</code> (Implementation File for Leave using C++ exceptions)	659
<code>oscl_error_imp_fatalerror.h</code> (Implementation File for Leave using system fatal error)	660
<code>oscl_error_imp_jumps.h</code> (Implemenation of using Setjmp / Longjmp)	661

oscl_error_trapcleanup.h (OSCL Error trap and cleanup implementation include file)	663
oscl_exception.h (Contains all the exception handling macros and classes)	664
oscl_exclusive_ptr.h (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)	665
oscl_file_async_read.h	666
oscl_file_cache.h (The file <code>oscl_file_cache.h</code> defines the class <code>OsclFileCache</code>)	667
oscl_file_dir_utils.h (The file <code>oscl_file_dir_utils.h</code> defines some unix-style directory ops)	668
oscl_file_find.h (The file <code>oscl_file_find.h</code> defines the class <code>Oscl_FileFind</code>)	670
oscl_file_handle.h (The file <code>oscl_file_handle.h</code> defines the class <code>OsclFileHandle</code>)	671
oscl_file_io.h (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)	672
oscl_file_native.h (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)	673
oscl_file_server.h (The file <code>oscl_file_server.h</code> defines the class <code>Oscl_FileServer</code> . This is the porting layer for file server implementations)	674
oscl_file_stats.h (File stats class)	675
oscl_file_types.h (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)	676
oscl_heapbase.h (OSCL Heap Base include file)	677
oscl_init.h (Global oscl initialization)	678
oscl_int64_utils.h	679
oscl_ip_socket.h	680
oscl_linked_list.h (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)	681
oscl_lock_base.h (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)	682
oscl_map.h (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)	683
oscl_math.h (Provides math functions)	684
oscl_media_data.h (Defines a container class for media data made up of a collection of memory fragments)	685
oscl_media_status.h (Defines a status values for the <code>MediaData</code> containers)	686
oscl_mem.h (This file contains basic memory definitions for common use across platforms)	687
oscl_mem_align.h	690
oscl_mem_audit.h (This file contains the definition and partial implementation of MM_Audit class)	691
oscl_mem_audit_internals.h (This file contains the internal definitions for the mem audit library)	693
oscl_mem_auto_ptr.h (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)	694
oscl_mem_basic_functions.h (This file contains prototypes for the basic memory functions)	695
oscl_mem_inst.h (The file defines default memory instrumentation level)	696
oscl_mem_mempool.h (This file contains the definition of memory pool allocators)	697
oscl_mempool_allocator.h (This file contains the definition of memory pool allocator for leave/trap)	698
oscl_mutex.h (This file provides implementation of mutex)	699
oscl_namestring.h (Name string class include file)	700

<code>oscl_opaque_type.h</code> (The file <code>oscl_opaque_type.h</code> defines pure virtual classes for working with opaque types)	701
<code>oscl_pqueue.h</code> (Implements a priority queue data structure similar to STL)	702
<code>oscl_proctstatus.h</code>	703
<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)	704
<code>oscl_rand.h</code> (Provides pseudo-random number generation)	705
<code>oscl_refcounter.h</code> (A general purpose reference counter to object lifetimes)	706
<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)	707
<code>oscl_registry_access_client.h</code> (Client-side implementation Registry Access implementation)	708
<code>oscl_registry_client.h</code> (Client-side implementation of <code>OsclRegistry</code>)	709
<code>oscl_registry_client_impl.h</code> (Client-side implementation of <code>OsclRegistryInterface</code>)	710
<code>oscl_registry_serv_impl.h</code> (Server-side implementation of <code>OsclRegistry</code> interfaces)	711
<code>oscl_registry_serv_impl_global.h</code>	712
<code>oscl_registry_serv_impl_tls.h</code>	713
<code>oscl_registry_types.h</code> (Common types used in <code>Oscl registry</code> interfaces)	714
<code>oscl_scheduler.h</code>	715
<code>oscl_scheduler_ao.h</code> (<code>Oscl Scheduler</code> user execution object classes)	716
<code>oscl_scheduler_aobase.h</code> (<code>Oscl Scheduler</code> internal active object classes)	717
<code>oscl_scheduler_readyq.h</code> (Ready q types for <code>oscl scheduler</code>)	718
<code>oscl_scheduler_threadcontext.h</code> (Thread context functions needed by <code>oscl scheduler</code>)	719
<code>oscl_scheduler_tuneables.h</code> (Tunable settings for <code>Oscl Scheduler</code>)	720
<code>oscl_scheduler_types.h</code> (Scheduler common types include file)	721
<code>oscl_semaphore.h</code> (This file provides implementation of mutex)	722
<code>oscl_shared_ptr.h</code> (This file defines a template class <code>OsclSharedPtr</code> which is a "smart pointer" to the parameterized type)	723
<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)	724
<code>oscl_snprintf.h</code> (Provides a portable implementation of <code>snprintf</code>)	726
<code>oscl_socket.h</code> (The file <code>oscl_socket.h</code> defines the OSCL Socket APIs)	727
<code>oscl_socket_accept.h</code>	728
<code>oscl_socket_bind.h</code>	729
<code>oscl_socket_connect.h</code>	730
<code>oscl_socket_imp.h</code>	731
<code>oscl_socket_imp_base.h</code>	732
<code>oscl_socket_imp_pv.h</code>	733
<code>oscl_socket_listen.h</code>	734
<code>oscl_socket_method.h</code>	735
<code>oscl_socket_recv.h</code>	736
<code>oscl_socket_recv_from.h</code>	737
<code>oscl_socket_request.h</code>	738
<code>oscl_socket_send.h</code>	739
<code>oscl_socket_send_to.h</code>	740
<code>oscl_socket_serv_imp.h</code>	741
<code>oscl_socket_serv_imp_base.h</code>	742
<code>oscl_socket_serv_imp_pv.h</code>	743
<code>oscl_socket_serv_imp_reqlist.h</code>	744
<code>oscl_socket_shutdown.h</code>	745

oscl_socket_stats.h	746
oscl_socket_tunables.h	748
oscl_socket_types.h	750
oscl_stdstring.h (This file provides standard string operations such as strlen, strcpy, etc)	752
oscl_str_ptr_len.h (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally)	753
oscl_string.h (Provides a standardized set of string containers that can be used in place of character arrays)	754
oscl_string_containers.h (Provides a standardized set of string containers that can be used in place of character arrays)	755
oscl_string_rep.h (Contains some internal implementation for string containers)	756
oscl_string_uri.h (Utilities to unescape URIs)	757
oscl_string_utf8.h (Utilities to validate and truncate UTF-8 encoded strings)	758
oscl_string_utils.h (Utilities to parse and convert strings)	759
oscl_string_xml.h (Utilities to escape special characters in XML strings)	760
oscl_tagtree.h (The file oscl_tagtree.h ..)	761
oscl_tcp_socket.h	762
oscl_thread.h	763
oscl_tickcount.h (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally)	765
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)	766
oscl_timer.h	768
oscl_tls.h	769
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)	770
oscl_types.h (This file contains basic type definitions for common use across platforms)	771
oscl_udp_socket.h	772
oscl_utf8conv.h (Utilities to convert unicode to utf8 and vice versa)	773
oscl_uuid.h (This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUuid32)	774
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)	775
osclconfig.h (This file contains configuration information for the linux platform)	776
osclconfig_ansi_memory.h (This file contains common typedefs based on the ANSI C limits.h header)	778
osclconfig_check.h	779
osclconfig_compiler_warnings.h (This file contains the ability to turn off/on compiler warnings)	780
osclconfig_error.h (This file contains the common typedefs and header files needed to compile osclerror)	781
osclconfig_error_check.h	782
osclconfig_global_new_delete.h	783
osclconfig_global_placement_new.h	784
osclconfig_io.h (This file contains common typedefs based on the ANSI C limits.h header)	785
osclconfig_io_check.h	792
osclconfig_ix86.h (This file contains configuration information for the ix86 processor family)	793
osclconfig_lib.h (This file contains configuration information for the ANSI build)	794
osclconfig_lib_check.h	795

osclconfig_limits_typedefs.h (This file contains common typedefs based on the ANSI C limits.h header)	796
osclconfig_memory.h	797
osclconfig_memory_check.h	798
osclconfig_no_os.h	799
osclconfig_proc.h (This file contains configuration information for the linux platform)	800
osclconfig_proc_check.h	801
osclconfig_proc_unix_common.h	803
osclconfig_proc_unix_nj.h	805
osclconfig_time.h	807
osclconfig_time_check.h	808
osclconfig_unix_common.h	809
osclconfig_unix_nj.h	813
osclconfig_util.h	817
osclconfig_util_check.h	818
pvlogger.h (This file contains basic logger interfaces for common use across platforms)	819
pvlogger_accessories.h	827
pvlogger_c.h (This file contains basic logger interfaces for common use across platforms. C-callable version)	828
pvlogger_registry.h	830

Chapter 5

oscl Module Documentation

5.1 OSCL config

Defines

- #define OSCL_ASSERT_ALWAYS 0
- #define OSCL_INTEGERS_WORD_ALIGNED 1
- #define OSCL_BYTE_ORDER_BIG_ENDIAN 0
- #define OSCL_BYTE_ORDER_LITTLE_ENDIAN 1
- #define OSCL_HAS_UNIX_SUPPORT 0
- #define OSCL_HAS_MSWIN_SUPPORT 0
- #define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_SUPPORT 0
- #define OSCL_HAS_SAVAJE_SUPPORT 0
- #define OSCL_HAS_PV_C_OS_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_ERRORTRAP 0
- #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0
- #define OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS 0
- #define OSCL_HAS_PV_C_OS_TIME_FUNCS 0
- #define OSCL_HAS_UNIX_TIME_FUNCS 0
- #define OSCL_HAS_SYMBIAN_TIMERS 0
- #define OSCL_HAS_SYMBIAN_MATH 0
- #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0
- #define OSCL_HAS_PTHREAD_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0
- #define OSCL_HAS_SAVAJE_IO_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0
- #define OSCL_HAS_SYMBIAN_DNS_SERVER 0
- #define OSCL_HAS_BERKELEY_SOCKETS 0

Typedefs

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

5.1.1 Define Documentation

5.1.1.1 #define OSCL_ASSERT_ALWAYS 0

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

5.1.1.2 #define OSCL_BYTE_ORDER_BIG_ENDIAN 0

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

5.1.1.3 #define OSCL_BYTE_ORDER_LITTLE_ENDIAN 1

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

- 5.1.1.4 #define OSCL_HAS_BERKELEY_SOCKETS 0
- 5.1.1.5 #define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0
- 5.1.1.6 #define OSCL_HAS_MSWIN_SUPPORT 0
- 5.1.1.7 #define OSCL_HAS_PTHREAD_SUPPORT 0
- 5.1.1.8 #define OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS 0
- 5.1.1.9 #define OSCL_HAS_PV_C_OS_SUPPORT 0
- 5.1.1.10 #define OSCL_HAS_PV_C_OS_TIME_FUNCS 0
- 5.1.1.11 #define OSCL_HAS_SAVAJE_IO_SUPPORT 0
- 5.1.1.12 #define OSCL_HAS_SAVAJE_SUPPORT 0
- 5.1.1.13 #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0
- 5.1.1.14 #define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0
- 5.1.1.15 #define OSCL_HAS_SYMBIAN_DNS_SERVER 0
- 5.1.1.16 #define OSCL_HAS_SYMBIAN_ERRORTRAP 0
- 5.1.1.17 #define OSCL_HAS_SYMBIAN_MATH 0
- 5.1.1.18 #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0
- 5.1.1.19 #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- 5.1.1.20 #define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0
- 5.1.1.21 #define OSCL_HAS_SYMBIAN_SUPPORT 0
- 5.1.1.22 #define OSCL_HAS_SYMBIAN_TIMERS 0
- 5.1.1.23 #define OSCL_HAS_UNIX_SUPPORT 0
- 5.1.1.24 #define OSCL_HAS_UNIX_TIME_FUNCS 0
- 5.1.1.25 #define OSCL_INTEGERS_WORD_ALIGNED 1

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

5.1.2 Typedef Documentation

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

5.2 OSCL Base

Files

- file [oscl_assert.h](#)

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

- file [oscl_base.h](#)

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

- file [oscl_base_alloc.h](#)

A basic allocator that does not rely on other modules.

- file [oscl_base_macros.h](#)

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

- file [oscl_byte_order.h](#)

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

- file [oscl_defalloc.h](#)

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

- file [oscl_dll.h](#)

Defines a DLL entry point.

- file [oscl_exclusive_ptr.h](#)

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

- file [oscl_linked_list.h](#)

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

- file [oscl_lock_base.h](#)

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

- file [oscl_map.h](#)

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

- file [oscl_mem_inst.h](#)

The file defines default memory instrumentation level.

- file [oscl_opaque_type.h](#)

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

- file [oscl_queue.h](#)

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

- file [oscl_refcounter.h](#)

A general purpose reference counter to object lifetimes.

- file [oscl_refcounter_memfrag.h](#)

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

- file [oscl_shared_ptr.h](#)

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

- file [oscl_stdstring.h](#)

This file provides standard string operations such as `strlen`, `strncpy`, etc.

- 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_Dalloc
- 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 OsclRefCount
- class OsclRefCountDA
- class OsclRefCountMemFrag
- class OsclRefCountMTDA

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

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

- class [OsclSharedPtr](#)

A parameterized smart pointer class.

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

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

- class [TLSStorageOps](#)

Defines

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

The NULL_TERM_CHAR is used to terminate c-style strings.

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

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

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

Type casting macros.

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

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

TypeDefs

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

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

- typedef void OsclAny

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

- typedef char mbchar

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

- typedef unsigned int uint

The uint type is a convenient abbreviation for unsigned int.

- typedef uint8 octet

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

- typedef float OsclFloat

The Float type defined as OsclFloat.

- typedef OSCL_NATIVE_INT64_TYPE int64

- typedef OSCL_NATIVE_UINT64_TYPE uint64

- typedef OSCL_NATIVE_WCHAR_TYPE oscl_wchar

- typedef oscl_wchar OSCL_TCHAR

define OSCL_TCHAR

Enumerations

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

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

Functions

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

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

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

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

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

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

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

Variables

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

5.2.1 Detailed Description

Additional osclbase comment

Additional osclbase comment

Additional osclbase comment

5.2.2 Define Documentation

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

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

5.2.2.3 #define NULL (0)

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

5.2.2.4 #define NULL_TERM_CHAR '\0'

The NULL_TERM_CHAR is used to terminate c-style strings.

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

5.2.2.6 #define OSCL_ASSERT(_expr) ((_expr)?((void)0):OSCLAssert(#_expr,__FILE__,__LINE__))

5.2.2.7 #define OSCL_BEGIN_PACKED "error"

5.2.2.8 #define OSCL_COND_EXPORT_REF

5.2.2.9 #define OSCL_COND_IMPORT_REF

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

Type casting macros.

Parameters:

type Destination type of cast

exp Expression to cast

5.2.2.11 #define OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT

5.2.2.12 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

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

DLL entry/exit point.

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

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

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

Usage :

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

5.2.2.14 #define OSCL_DLL_ENTRY_POINT_DEFAULT()

Default DLL entry/exit point function.

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

Usage :

```
OSCL_DLL_ENTRY_POINT_DEFAULT()
```

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

5.2.2.16 #define OSCL_END_PACKED "error"

5.2.2.17 #define OSCL_HAS_SINGLETON_SUPPORT 1

5.2.2.18 #define OSCL_INLINE inline

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

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

5.2.2.21 #define OSCL_PACKED_VAR "error"

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

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

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

5.2.2.25 #define OSCL_TLS_BASE_SLOTS OSCL_TLS_ID_BASE_LAST +1

5.2.2.26 #define OSCL_TLS_EXTERNAL_SLOTS 0

5.2.2.27 #define OSCL_TLS_MAX_SLOTS (OSCL_TLS_BASE_SLOTS + OSCL_TLS_EXTERNAL_SLOTS)

5.2.2.28 #define OSCL_UNSIGNED_CONST(x) x

5.2.2.29 #define OSCL_UNUSED_ARG(vbl) (void)(vbl)

The following two macros are used to avoid compiler warnings.

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

5.2.2.30 `#define OSCL_UNUSED_RETURN(value) return value`

5.2.2.31 `#define PVMEM_INST_LEVEL 1`

5.2.3 Typedef Documentation

5.2.3.1 `typedef int c_bool`

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

5.2.3.2 `typedef char CtimeStrBuff[CTIME_BUFFER_SIZE]`

5.2.3.3 `typedef OSCL_NATIVE_INT64_TYPE int64`

5.2.3.4 `typedef char mbchar`

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

5.2.3.5 `typedef uint8 octet`

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

5.2.3.6 `typedef oscl_wchar OSCL_TCHAR`

define `OSCL_TCHAR`

5.2.3.7 `typedef OSCL_NATIVE_WCHAR_TYPE oscl_wchar`

5.2.3.8 `typedef void OsclAny`

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

5.2.3.9 `typedef float OsclFloat`

The `Float` type defined as `OsclFloat`.

5.2.3.10 `typedef char PV8601timeStrBuf[PV8601TIME_BUFFER_SIZE]`

5.2.3.11 `typedef OsclAny TOsclTlsKey`

5.2.3.12 `typedef unsigned int uint`

The uint type is a convenient abbreviation for unsigned int.

5.2.3.13 `typedef OSCL_NATIVE_UINT64_TYPE uint64`

5.2.4 Enumeration Type Documentation

5.2.4.1 `enum TimeUnits`

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

Enumeration values:

SECONDS

MILLISECONDS

MICROSECONDS

5.2.5 Function Documentation

5.2.5.1 `OSCL_COND_IMPORT_REF void _OSCL_Abort()`

This function terminates the current process abnormally.

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

Convert big endian to host format.

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

Parameters:

data A pointer to the input/output buffer

size The number of bytes in the buffer.

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

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

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

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

5.2.5.5 void host_to_big_endian (char * *data*, unsigned int *size*)

Convert host to big endian format.

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

Parameters:

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

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

Convert host to little endian format.

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

Parameters:

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

5.2.5.7 void little_endian_to_host (char * *data*, uint32 *size*)

Convert little endian to host format.

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

Parameters:

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

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

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

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

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

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

Parameters:

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

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

Case in-sensitive string comparision.

Parameters:

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

Returns:

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

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

Case in-sensitive string comparision.

Parameters:

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

Returns:

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

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

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

Parameters:

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

Returns:

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

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

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

Parameters:

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

Returns:

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

5.2.5.15 OSCL_IMPORT_REF bool oscl_isLetter (const char *car*)

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

Parameters:*car***Returns:**

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

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

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

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

dest

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

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

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

dest

5.2.5.18 OSCL_IMPORT_REF oscl_wchar* oscl_strchr (oscl_wchar * *str*, int32 *c*)**5.2.5.19 OSCL_IMPORT_REF const oscl_wchar* oscl_strchr (const oscl_wchar * *str*, int32 *c*)**

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

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

5.2.5.20 OSCL_IMPORT_REF char* oscl_strchr (char * str, int32 c)**5.2.5.21 OSCL_IMPORT_REF const char* oscl_strchr (const char * str, int32 c)**

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

Parameters:

str null terminated source string

c character to search for

Returns:**5.2.5.22 OSCL_IMPORT_REF int32 oscl_strcmp (const oscl_wchar * str1, const oscl_wchar * str2)**

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

Parameters:

str1 String to compare

str2 String to compare

Returns:

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

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

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

Parameters:

str1 String to compare

str2 String to compare

Returns:

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

5.2.5.24 OSCL_IMPORT_REF uint32 oscl_strlen (const oscl_wchar * str)

Gets the length of a wide char string

Parameters:

str NULL terminated string.

Returns:

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

5.2.5.25 OSCL_IMPORT_REF uint32 oscl_strlen (const char * str)

Gets the length of a string

Parameters:

str NULL terminated string.

Returns:

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

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

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

Parameters:

dest null terminated destination string

src source string

count number of characters to append.

Returns:

dest

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

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

Parameters:

dest null terminated destination string

src source string

count number of characters to append.

Returns:

dest

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

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

Parameters:

str1 String to compare

str2 String to compare

count Number of characters to compare

Returns:

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

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

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

Parameters:

str1 String to compare

str2 String to compare

count Number of characters to compare

Returns:

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

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

Copies the chars of one string to another.

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

Parameters:

dest Destination string

src NULL terminated source string

count Number of chars to copy

Returns:

Returns dest.

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

Copies the chars of one string to another.

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

Parameters:

dest Destination string

src NULL terminated source string

count Number of chars to copy

Returns:

Returns dest.

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

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

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

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

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

Parameters:

str null terminated source string

c character to search for

Returns:

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

Sets the characters of a string to a specified character

Parameters:

dest buffer to modify

val character to set

count number of chars to set

Returns:

the value of dest

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

Sets the characters of a string to a specified character

Parameters:

dest buffer to modify

val character to set

count number of chars to set

Returns:

the value of dest

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

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

find the occurrence of sub-string in a string.

Parameters:

str1 string.

str2 sub-string

Returns:

pointer to the begining of sub-string.

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

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

find the occurrence of sub-string in a string.

Parameters:

str1 string.

str2 sub-string

Returns:

pointer to the begining of sub-string.

5.2.5.42 OSCL_IMPORT_REF oscl_wchar oscl_tolower (const oscl_wchar car)

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

Parameters:

car upper case character.

Returns:

lower case character.

5.2.5.43 OSCL_IMPORT_REF char oscl_tolower (const char car)

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

Parameters:

car upper case character.

Returns:

lower case character.

5.2.5.44 OSCL_IMPORT_REF void PV8601ToRFC822 (PV8601timeStrBuf** *pv8601_buffer*,
CtimeStrBuf *ctime_buffer*)**

5.2.5.45 void PVOsclBase_Cleanup ()

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

5.2.5.46 void PVOsclBase_Init ()

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

Exceptions:

leaves if out-of-memory

5.2.5.47 OSCL_IMPORT_REF void RFC822ToPV8601 (**CtimeStrBuf** *ctime_buffer*,
PV8601timeStrBuf)

5.2.6 Variable Documentation

5.2.6.1 const int **CTIME_BUFFER_SIZE** = 26

5.2.6.2 const long **MSEC_PER_SEC** = 1000

5.2.6.3 const uint32 **OSCL_TLS_ID_BASE_LAST** = 11

5.2.6.4 const uint32 **OSCL_TLS_ID_ERRORHOOK** = 1

5.2.6.5 const uint32 **OSCL_TLS_ID_MAGICNUM** = 0

5.2.6.6 const uint32 **OSCL_TLS_ID_OSCLREGISTRY** = 10

5.2.6.7 const uint32 **OSCL_TLS_ID_PAYLOADPARSER** = 7

5.2.6.8 const uint32 **OSCL_TLS_ID_PVERRORTRAP** = 5

5.2.6.9 const uint32 **OSCL_TLS_ID_PVLOGGER** = 2

5.2.6.10 const uint32 **OSCL_TLS_ID_PVMFRECOGNIZER** = 8

5.2.6.11 const uint32 **OSCL_TLS_ID_PVSCHEDULER** = 4

5.2.6.12 const uint32 **OSCL_TLS_ID_SDPMEDIAPARSER** = 6

5.2.6.13 const uint32 **OSCL_TLS_ID_SQLITE3** = 11

5.2.6.14 const uint32 **OSCL_TLS_ID_TEST** = 3

5.2.6.15 const uint32 **OSCL_TLS_ID_WMDRM** = 9

5.2.6.16 const int **PV8601TIME_BUFFER_SIZE** = 21

5.2.6.17 const uint32 **unix_ntp_offset** = 2208988800U

5.2.6.18 const long **USEC_PER_SEC** = 1000000

5.3 OSCL Memory

Files

- file [oscl_mem.h](#)

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

- file [oscl_mem_audit.h](#)

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

- file [oscl_mem_audit_internals.h](#)

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

- file [oscl_mem_auto_ptr.h](#)

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

- file [oscl_mem_basic_functions.h](#)

This file contains prototypes for the basic memory functions.

- file [oscl_mem_mempool.h](#)

This file contains the definition of memory pool allocators.

Data Structures

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

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

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

Defines

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

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

Typedefs

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

Functions

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

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

Variables

- const uint32 `ALLOC_NODE_FLAG` = 0x80000000

5.3.1 Define Documentation

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

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

Parameters:

T: base class name.

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

Value:

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

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

Parameters:

exp: expression to allocate memory.

Tptr:variable to hold result.

T: type

params: constructor arg list

freeFunc: delete or free function.

- 5.3.1.3 #define COMPUTE_MEM_ALIGN_SIZE(x, y, z) (y+((x+y)%z) ? (z - (x+y)%z) : 0))
- 5.3.1.4 #define DEFAULT_MM_AUDIT_MODE 0
- 5.3.1.5 #define DEFAULT_POSTFILL_PATTERN 0x5A
- 5.3.1.6 #define DEFAULT_PREFILL_PATTERN 0x96
- 5.3.1.7 #define FENCE_PATTERN 0xAA
- 5.3.1.8 #define MEM_ALIGN_SIZE 8
- 5.3.1.9 #define MIN_FENCE_SIZE 4
- 5.3.1.10 #define MM_ALLOC_MAX_QUERY_FILENAME_LEN 128
- 5.3.1.11 #define MM_ALLOC_MAX_QUERY_TAG_LEN 64
- 5.3.1.12 #define MM_AUDIT_ALLOC_NODE_ENABLE_FLAG 0x10
- 5.3.1.13 #define MM_AUDIT_ALLOC_NODE_SUPPORT 1
- 5.3.1.14 #define MM_AUDIT_FAILURE_SIMULATION_SUPPORT 1
- 5.3.1.15 #define MM_AUDIT_FENCE_SUPPORT 0
- 5.3.1.16 #define MM_AUDIT_FILL_SUPPORT 0
- 5.3.1.17 #define MM_AUDIT_INCLUDE_ALL_HEAP_VALIDATION 1
- 5.3.1.18 #define MM_AUDIT_POSTFILL_FLAG 0x2
- 5.3.1.19 #define MM_AUDIT_PREFILL_FLAG 0x1
- 5.3.1.20 #define MM_AUDIT_SUPPRESS_FILENAME_FLAG 0x20
- 5.3.1.21 #define MM_AUDIT_VALIDATE_ALL_HEAP_FLAG 0x4
- 5.3.1.22 #define MM_AUDIT_VALIDATE_BLOCK 1
- 5.3.1.23 #define MM_AUDIT_VALIDATE_ON_FREE_FLAG 0x8
- 5.3.1.24 #define OSCL_ALLOC_DELETE(ptr, T_allocator, T)

Value:

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

Deletes the object of type T using the given allocator

Parameters:

T_allocator allocator for objects of type T

T type of object to delete

ptr pointer to previously created object

Exceptions:

none , unless thrown by the given allocator

**5.3.1.25 #define OSCL_ALLOC_NEW(T_allocator, T, params) new(T_allocator.allocate(1)) T
params**

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

Parameters:

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

T type of object to create

params object initialization parameters

Returns:

pointer to created object

Exceptions:

none , unless thrown by the given allocator

5.3.1.26 #define OSCL_ARRAY_DELETE(ptr) delete [] ptr

Oscl array delete operator..

Parameters:

ptr pointer to memory block previously allocated with OSCL_ARRAY_NEW

Returns:

void

5.3.1.27 #define OSCL_ARRAY_NEW(T, count) new T[count]

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

Parameters:

T data type for 'new' operation

count number of elements to create

Returns:

pointer to the newly created object array of type T

Exceptions:

may leave with code = bad alloc

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

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

Parameters:

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

Returns:

pointer to the newly created object array of type T

Exceptions:

may leave with code = bad alloc

5.3.1.29 #define OSCL_AUDIT_CALLOC(auditCB, num, size) _oscl_audit_calloc(num,size, auditCB)

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

Parameters:

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

Returns:

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

Exceptions:

none

5.3.1.30 #define OSCL_AUDIT_MALLOC(auditCB, count) _oscl_audit_malloc(count, auditCB)

Allocates a memory block using the given audit object.

Parameters:

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

Returns:

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

Exceptions:

none

5.3.1.31 #define OSCL_AUDIT_NEW(auditCB, T, params) new(_oscl_audit_new(sizeof(T),audit-CB)) T params

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

Parameters:

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

Returns:

pointer to the newly created object of type *T*

Exceptions:

may leave with code = bad alloc

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

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

Parameters:

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

Returns:

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

Exceptions:

none

5.3.1.33 #define oscl_calloc(a, b) OSCL_CALLOC(a,b)**5.3.1.34 #define OSCL_CALLOC(num, size) _oscl_default_audit_calloc(num,size)**

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

Parameters:

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

Returns:

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

Exceptions:

none

5.3.1.35 #define OSCL_CLEANUP_BASE_CLASS(T) _OSCL_CLEANUP_BASE_CLASS(T)

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

Parameters:

T: name of the base class.

5.3.1.36 #define OSCL_DEFAULT_FREE(x) OSCL_FREE(x)

Another back-compatibility definition.

5.3.1.37 #define OSCL_DEFAULT_MALLOC(x) OSCL_MALLOC(x)

Another back-compatibility definition.

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

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

Oscl "delete" operator.

Parameters:

ptr pointer to memory block previously allocated with OSCL_NEW

Returns:

void

5.3.1.39 #define OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT**5.3.1.40 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE**

Previously this was in oscl_mem_imp.h

5.3.1.41 #define oscl_free(x) OSCL_FREE(x)**5.3.1.42 #define OSCL_FREE(ptr) _oscl_audit_free(ptr)**

Deallocates or frees a memory block.

Parameters:

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

5.3.1.43 #define OSCL_HAS_GLOBAL_NEW_DELETE 1

5.3.1.44 #define oscl_malloc(a) OSCL_MALLOC(a)

5.3.1.45 #define OSCL_MALLOC(count) _oscl_default_audit_malloc(count)

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

Parameters:

count number of bytes to allocate

Returns:

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

Exceptions:

none

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

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

Parameters:

T data type for 'new' operation

params object initialization parameters

Returns:

pointer to the newly created object of type T

Exceptions:

may leave with code = bad alloc

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

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

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

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

Parameters:

ptr original memory block

new_size New size of the block

Returns:

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

Exceptions:

none

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

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

Parameters:

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

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

T type of object to create

params object initialization parameters

Returns:

pointer to created object

Exceptions:

none , unless thrown by the given allocator

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

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

Parameters:

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

auditCB input memory management audit object

T data type for 'new' operation

params object initialization parameters

Returns:

pointer to the newly created object of type T

Exceptions:

may leave with code = bad alloc

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

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

Parameters:

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

T data type for 'new' operation

params object initialization parameters

Returns:

pointer to the newly created object of type T

Exceptions:

may leave with code = bad alloc

5.3.2 Typedef Documentation

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

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

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

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

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

5.3.2.6 `typedef Oscl_TagTree<MM_StatsNodeTagTreeType, TagTree_Allocator> OsclTagTreeType`

5.3.2.7 `typedef Oscl_TAlloc<MM_StatsNodeTagTreeType, OsclMemBasicAllocator> TagTree_Allocator`

5.3.3 Function Documentation

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

5.3.3.2 `OSCL_IMPORT_REF void _oscl_audit_free (void *)`

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

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

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

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

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

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

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

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

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

5.3.3.11 OSCL_COND_IMPORT_REF void _oscl_free (void **src*)

5.3.3.12 OSCL_COND_IMPORT_REF void* _oscl_malloc (int32 *count*)

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

5.3.3.14 void operator delete (void *) [inline]

5.3.3.15]

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

5.3.3.16 void* operator new (size_t) [inline]

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

5.3.3.18]

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

5.3.3.19]

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

5.3.3.20 OSCL_COND_IMPORT_REF uint oscl_mem_aligned_size (uint *size*)

Get memory-aligned size of an object.

Parameters:

size size of object

Returns:

memory-aligned size

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

Compare characters in two buffers

Parameters:

buf1 first buffer

buf2 second buffer

count number of bytes to compare

Returns:

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

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

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

Parameters:

dest new buffer

src buffer to copy

count number of bytes to copy

Returns:

the value of dest

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

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

Parameters:

dest new buffer

src buffer to copy

count number of bytes to copy

Returns:

the value of dest

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

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

Parameters:

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

Returns:

the value of dest

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

Sets the bytes of a buffer to a specified character

Parameters:

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

Returns:

the value of dest

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

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

Parameters:

auditCB memory management audit object

5.3.4 Variable Documentation

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

5.4 OSCL Util

Files

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

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

Data Structures

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

- class [OsclBinOStream](#)

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

- class [OsclBinOStreamBigEndian](#)

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

- class [OsclBinOStreamLittleEndian](#)

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

- class [OsclBinStream](#)

- class [OsclCompareLess](#)

- class [OsclComponentRegistry](#)

- class [OsclComponentRegistryData](#)

- class [OsclComponentRegistryElement](#)

- class [OsclPriorityQueue](#)

- class [OsclPriorityQueueBase](#)

- class [OsclRand](#)

- class [OsclRegistryAccessClient](#)

- class [OsclRegistryAccessClientImpl](#)

- class [OsclRegistryAccessClientTlsImpl](#)

- class [OsclRegistryAccessElement](#)

- class [OsclRegistryClient](#)

- class [OsclRegistryClientImpl](#)

- class [OsclRegistryClientTlsImpl](#)

- class [OsclRegistryServTlsImpl](#)

- class [OsclTickCount](#)

- struct [StrCSumPtrLen](#)

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

- struct [StrPtrLen](#)

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

- struct [WStrPtrLen](#)

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

Defines

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

Typedefs

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

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

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

Functions

- OSCL_IMPORT_REF const char * [skip_whitespace](#) (const char *ptr)
- OSCL_IMPORT_REF char * [skip_whitespace](#) (char *ptr)
- OSCL_IMPORT_REF const char * [skip_whitespace](#) (const char *start, const char *end)
- OSCL_IMPORT_REF const char * [skip_to_whitespace](#) (const char *start, const char *end)
- OSCL_IMPORT_REF const char * [skip_to_line_term](#) (const char *start_ptr, const char *end_ptr)
- OSCL_IMPORT_REF const char * [skip_whitespace_and_line_term](#) (const char *start, const char *end)
- OSCL_IMPORT_REF int [extract_string](#) (const char *in_ptr, char *outstring, int maxsize)
- OSCL_IMPORT_REF int [extract_string](#) (const char *start, const char *end, char *outstring, int maxsize)
- OSCL_IMPORT_REF bool [PV_atoi](#) (const char *buf, const char new_format, uint32 &value)
- OSCL_IMPORT_REF bool [PV_atoi](#) (const char *buf, const char new_format, int length, uint32 &value)
- OSCL_IMPORT_REF bool [PV_atoi](#) (const char *buf, const char new_format, int length, [uint64](#) &value)
- OSCL_IMPORT_REF bool [PV_atof](#) (const char *buf, [OsclFloat](#) &value)
- OSCL_IMPORT_REF bool [PV_atof](#) (const char *buf, int length, [OsclFloat](#) &value)
- OSCL_IMPORT_REF int [oscl_abs](#) (int aVal)
- OSCL_COND_IMPORT_REF double [oscl_log](#) (double value)
- OSCL_COND_IMPORT_REF double [oscl_log10](#) (double value)
- OSCL_COND_IMPORT_REF double [oscl_sqrt](#) (double value)
- OSCL_COND_IMPORT_REF double [oscl_pow](#) (double x, double y)
- OSCL_COND_IMPORT_REF double [oscl_exp](#) (double value)
- OSCL_COND_IMPORT_REF double [oscl_sin](#) (double value)
- OSCL_COND_IMPORT_REF double [oscl_cos](#) (double value)
- OSCL_COND_IMPORT_REF double [oscl_tan](#) (double value)
- OSCL_COND_IMPORT_REF double [oscl_asin](#) (double value)
- OSCL_COND_IMPORT_REF double [oscl_atan](#) (double value)
- OSCL_COND_IMPORT_REF double [oscl_floor](#) (double value)
- OSCL_IMPORT_REF int32 [oscl_snprintf](#) (char *str, uint32 count, const char *fmt,...)
- OSCL_IMPORT_REF int32 [oscl_snprintf](#) ([oscl_wchar](#) *str, uint32 count, const [oscl_wchar](#) *fmt,...)
- OSCL_IMPORT_REF int32 [oscl_vsnprintf](#) (char *str, uint32 count, const char *fmt, va_list args)
- OSCL_IMPORT_REF int32 [oscl_vsnprintf](#) ([oscl_wchar](#) *str, uint32 count, const [oscl_wchar](#) *fmt, va_list args)
- OSCL_IMPORT_REF bool [oscl_str_unescape_uri](#) (const char *str_buf_in, char *str_buf_out, uint32 max_out_buf_bytes, uint32 max_bytes, uint32 &out_buf_len)
unescape any of the special escape sequence in the uri string
- OSCL_IMPORT_REF bool [oscl_str_unescape_uri](#) (const [OSCL_String](#) &oscl_str_in, [OSCL_String](#) &oscl_str_out, uint32 &out_buf_len)

unescape any of the special escape sequence in the uri string

- OSCL_IMPORT_REF bool [oscl_str_is_valid_utf8](#) (const uint8 *str_buf, uint32 &num_valid_characters, uint32 max_bytes=0, uint32 max_char_2_valid=0, uint32 *num_byte_4_char=NULL)

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

- OSCL_IMPORT_REF int32 [oscl_str_truncate_utf8](#) (uint8 *str_buf, uint32 max_char, uint32 max_bytes=0)

Truncates the UTF-8 string upto the required size.

- OSCL_IMPORT_REF bool [oscl_str_need_escape_xml](#) (const char *str_buf, uint32 &num_escape_bytes, uint32 max_bytes=0)

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

- OSCL_IMPORT_REF int32 [oscl_str_escape_xml](#) (const char *str_buf_in, char *str_buf_out, uint32 max_out_buf_bytes, uint32 max_bytes=0, uint32 *num_bytes_written=NULL)

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

- OSCL_IMPORT_REF int32 [oscl_UTF8ToUnicode](#) (const char *input, int32 inLength, oscl_wchar *output, int32 outLength)

Convert UTF8 byte sequence to Unicode string.

- OSCL_IMPORT_REF int32 [oscl_UncodeToUTF8](#) (const oscl_wchar *input, int32 inLength, char *output, int32 outLength)

Convert Unicode string to UTF8 byte sequence.

- [BufferFragment * GetFragment](#) (const int32 idx)
- [BufferState * GetBufferState](#) (const int32 idx)
- uint32 [get_size](#) () const
- uint32 [get_size](#) () const
- uint32 [get_maxsize](#) () const
- uint32 [get_maxsize](#) () const
- const chartype * [get_cstr](#) () const
- const chartype * [get_cstr](#) () const
- chartype * [get_str](#) () const
- chartype * [get_str](#) () const
- [OSCL_HeapString](#) ()
- [OSCL_wHeapString](#) ()
- [OSCL_HeapString](#) (const chartype *cstr)
- [OSCL_wHeapString](#) (const chartype *cstr)
- void [set](#) (const chartype *buf, uint32 length)
- void [set](#) (const chartype *buf, uint32 length)
- [OSCL_HeapString](#) (const chartype *buf, uint32 length)
- [OSCL_wHeapString](#) (const chartype *buf, uint32 length)
- [OSCL_HeapString](#) (const OSCL_HeapString &src)
- [OSCL_wHeapString](#) (const OSCL_wHeapString &src)

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

Variables

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

5.4.1 Define Documentation

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

5.4.1.2 `#define OSCLTICKCOUNT_MAX_TICKS 0xffffffff`

5.4.2 Typedef Documentation

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

5.4.2.2 `typedef uint32 MediaTimestamp`

5.4.2.3 `typedef WStrPtrLen OSCL_TStrPtrLen`

5.4.2.4 `typedef OsclAny* OsclComponentFactory`

`OsclComponentFactory` is an opaque pointer.

5.4.2.5 `typedef StrCSumPtrLen StrCSumPtrLen`

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

5.4.2.6 `typedef struct StrPtrLen StrPtrLen`

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

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

5.4.2.7 `typedef struct WStrPtrLen WStrPtrLen`

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

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

5.4.3 Function Documentation

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

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

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

Implements `OSCL_wString`.

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

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

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

Implements [OSCL_wString](#).

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

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

Implements [OSCL_String](#).

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

Reimplemented from [OSCL_wString](#).

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

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

- 5.4.3.23 template<uint32 MaxBufSize> **OSCL_wStackString**< MaxBufSize > &
OSCL_wStackString< MaxBufSize >::operator= (const [OSCL_wString](#) & *src*)
 [inherited]

Reimplemented from [OSCL_wString](#).

- 5.4.3.24 template<uint32 MaxBufSize> **OSCL_StackString**< MaxBufSize > &
OSCL_StackString< MaxBufSize >::operator= (const [OSCL_String](#) & *src*)
 [inherited]

Assignment operator

Reimplemented from [OSCL_String](#).

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

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

Assignment operators

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

Reimplemented from [OSCL_wString](#).

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

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

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

Reimplemented from [OSCL_wString](#).

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

Assignment operator

Reimplemented from [OSCL_String](#).

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

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

Assignment operators

5.4.3.33 OSCL_IMPORT_REF int oscl_abs (int *aVal*)

5.4.3.34 OSCL_COND_IMPORT_REF double oscl_asin (double *value*)

Calculates the arc sine of a number

Parameters:

value source value

5.4.3.35 OSCL_COND_IMPORT_REF double oscl_atan (double *value*)

Calculates the arc tangent of a number

Parameters:

value source value

5.4.3.36 OSCL_COND_IMPORT_REF double oscl_cos (double *value*)

Calculates the cosine of a number

Parameters:

value source value

5.4.3.37 OSCL_COND_IMPORT_REF double oscl_exp (double *value*)

Calculates the exponential of e for a number

Parameters:

value source value

5.4.3.38 OSCL_COND_IMPORT_REF double oscl_floor (double *value*)

Calculates the floor of a number

Parameters:

value source value

5.4.3.39 template<class Alloc> OSCL_HeapString< Alloc >::OSCL_HeapString (const OSCL_String & *src*) [inherited]**5.4.3.40 template<class Alloc> OSCL_HeapString< Alloc >::OSCL_HeapString (const OSCL_HeapString< Alloc > & *src*) [inherited]**

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

Parameters:

src: input string.

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

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

Parameters:

src: character array, not necessarily null-terminated.

length: number of characters to copy.

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

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

Parameters:

cp: null-terminated string.

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

The default constructor creates an empty string.

5.4.3.44 OSCL_COND_IMPORT_REF double oscl_log (double *value*)

Calculates the natural log of a number

Parameters:

value source value

5.4.3.45 OSCL_COND_IMPORT_REF double oscl_log10 (double *value*)

Calculates the logarithm to base 10 of a number

Parameters:

value source value

5.4.3.46 OSCL_COND_IMPORT_REF double oscl_pow (double *x*, double *y*)

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

Parameters:

x base value

y power

5.4.3.47 OSCL_COND_IMPORT_REF double oscl_sin (double *value*)

Calculates the sine of a number

Parameters:

value source value

5.4.3.48 OSCL_IMPORT_REF int32 oscl_snprintf (*oscl_wchar* * *str*, uint32 *count*, const *oscl_wchar* * *fmt*, ...)**5.4.3.49 OSCL_IMPORT_REF int32 oscl_snprintf (char * *str*, uint32 *count*, const char * *fmt*, ...)****5.4.3.50 OSCL_COND_IMPORT_REF double oscl_sqrt (double *value*)**

Calculates the square root of a number

Parameters:

value source value

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

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

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

Parameters:

src: input string.

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

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

Parameters:

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

length: the number of characters to copy.

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

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

Parameters:

cp: a null-terminated string.

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

Creates an OSCL_StackString initialized with an empty string.

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

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

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

Parameters:

str_buf_in Ptr to an input string

str_buf_out Ptr to an output buffer which stores the modified string

max_out_buf_bytes The size of str_buf_out.

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

num_bytes_written Number of bytes written in the output buffer, str_buf_out

Returns:

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

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

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

Parameters:

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

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

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

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

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

Returns:

True if the string is valid and false otherwise.

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

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

Parameters:

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

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

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

Returns:

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

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

Truncates the UTF-8 string upto the required size.

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

Parameters:

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

max_char The max number of the UTF-8 CHARACTERS

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

Returns:

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

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

unescape any of the special escape sequence in the uri string

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

Parameters:

oscl_str_in Ptr to an input OSCL_String

oscl_str_out Ptr to an output OSCL_String which stores the modified string

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

Returns:

It returns true if succeeds, otherwise false.

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

unescape any of the special escape sequence in the uri string

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

Parameters:

str_buf_in Ptr to an input string

str_buf_out Ptr to an output buffer which stores the modified string

max_out_buf_bytes The size of str_buf_out.

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

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

Returns:

It returns true if succeeds, otherwise false.

5.4.3.62 OSCL_COND_IMPORT_REF double oscl_tan (double *value*)

Calculates the tangential of a number

Parameters:

value source value

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

Convert Unicode string to UTF8 byte sequence.

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

Parameters:

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

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

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

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

Returns:

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

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

Convert UTF8 byte sequence to Unicode string.

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

Parameters:

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

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

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

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

Returns:

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

- 5.4.3.65 **OSCL_IMPORT_REF int32 oscl_vsnprintf (oscl_wchar * str, uint32 count, const oscl_wchar * fmt, va_list args)**
- 5.4.3.66 **OSCL_IMPORT_REF int32 oscl_vsnprintf (char * str, uint32 count, const char * fmt, va_list args)**
- 5.4.3.67 **template<class Alloc> OSCL_wHeapString< Alloc >::OSCL_wHeapString (const OSCL_wString & src) [inherited]**
- 5.4.3.68 **template<class Alloc> OSCL_wHeapString< Alloc >::OSCL_wHeapString (const OSCL_wHeapString< Alloc > & src) [inherited]**
- 5.4.3.69 **template<class Alloc> OSCL_wHeapString< Alloc >::OSCL_wHeapString (const chartype * buf, uint32 length) [inherited]**
- 5.4.3.70 **template<class Alloc> OSCL_wHeapString< Alloc >::OSCL_wHeapString (const chartype * cstr) [inherited]**
- 5.4.3.71 **template<class Alloc> OSCL_wHeapString< Alloc >::OSCL_wHeapString () [inherited]**
- 5.4.3.72 **template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::OSCL_wStackString (const OSCL_wString & src) [inherited]**
- 5.4.3.73 **template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::OSCL_wStackString (const OSCL_wStackString< MaxBufSize > & src) [inherited]**
- 5.4.3.74 **template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::OSCL_wStackString (const chartype * buf, uint32 length) [inherited]**
- 5.4.3.75 **template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::OSCL_wStackString (const chartype * cstr) [inherited]**
- 5.4.3.76 **template<uint32 MaxBufSize> OSCL_wStackString< MaxBufSize >::OSCL_wStackString () [inherited]**
- 5.4.3.77 **OSCL_IMPORT_REF bool PV_atof (const char * buf, int length, OsclFloat & value)**
- 5.4.3.78 **OSCL_IMPORT_REF bool PV_atof (const char * buf, OsclFloat & value)**
- 5.4.3.79 **OSCL_IMPORT_REF bool PV_atoi (const char * buf, const char new_format, int length, uint64 & value)**
- 5.4.3.80 **OSCL_IMPORT_REF bool PV_atoi (const char * buf, const char new_format, int length, uint32 & value)**
- 5.4.3.81 **OSCL_IMPORT_REF bool PV_atoi (const char * buf, const char new_format, uint32 & value)**
- 5.4.3.82 **template<uint32 MaxBufSize> void OSCL_wStackString< MaxBufSize >::set (const chartype * buf, uint32 length) [inherited]**
- 5.4.3.83 **template<uint32 MaxBufSize> void OSCL_StackString< MaxBufSize >::set (const chartype * buf, uint32 length) [inherited]**

Parameters:

buf: string or character array.

length: number of characters to copy.

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

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

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

Parameters:

buf: string or character array.

length: number of characters to copy.

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

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

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

5.4.3.89 OSCL_IMPORT_REF char* skip_whitespace (char * *ptr*)

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

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

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

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

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

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

5.4.4 Variable Documentation

5.4.4.1 const int32 APPEND_MEDIA_AT_END = -1

5.4.4.2 const uint8 OSCL_ASCII_CASE_MAGIC_BIT = 0x20

5.5 OSCL Error

Files

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

Data Structures

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

- class [OsclErrorAllocator](#)

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

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

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

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

Defines

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Typedefs

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

Functions

- **OSCL_IMPORT_REF** bool **OSCL_IsErrnoSupported ()**

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

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

5.5.1 Define Documentation

5.5.1.1 #define _PV_TRAP(__r, __s)

Value:

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

5.5.1.2 #define _PV_TRAP_NO_TLS(__trapimp, __r, __s)

Value:

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

5.5.1.3 #define internalLeave (-1)

5.5.1.4 #define OSCL_BAD_ALLOC_EXCEPTION_CODE OsclErrNoMemory

5.5.1.5 #define OSCL_CATCH(_leave_status, _catch_value, _statements) else if $(_leave_status!=\text{OsclErrNone} \&\& _leave_status == _catch_value)\{_statements;\}$

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

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

Parameters:

oscl_leave_status is the result of any OSCL_THROW

exceptiontype is the exception handled by this catch block

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

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

Parameters:

_leave_status

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

5.5.1.7 #define OSCL_ERR_NONE OsclErrNone

For backward compatibility with old definitions

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

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

Parameters:

oscl_leave_status is the leave code that was returned by OSCL_THROW

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

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

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

Parameters:

_leave_status

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

5.5.1.10 #define OSCL_JUMP_MAX_JUMP_MARKS OSCL_MAX_TRAP_LEVELS

**5.5.1.11 #define OSCL_LAST_CATCH(_leave_status) else if (_leave_status!=OsclErr-
None){OSCL_LEAVE(_leave_status);}**

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

Parameters:

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

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

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

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

Parameters:

oscl_leave_status tells the cause for the Leave

5.5.1.13 #define OSCL_MAX_TRAP_LEVELS 20**5.5.1.14 #define OSCL_TRAPSTACK_POP() OsclError::Pop()****5.5.1.15 #define OSCL_TRAPSTACK_POPDEALLOC() OsclError::PopDealloc()****5.5.1.16 #define OSCL_TRAPSTACK_PUSH(a) OsclError::PushL(a)**

Cleanup Stack user macros

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

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

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

Parameters:

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

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

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

5.5.1.19 #define OsclErrAlreadyExists 106

5.5.1.20 #define OsclErrAlreadyInstalled 116

5.5.1.21 #define OsclErrArgument 104

5.5.1.22 #define OsclErrBadHandle 105

5.5.1.23 #define OsclErrBusy 107

5.5.1.24 #define OsclErrCancelled 102

5.5.1.25 #define OsclErrCorrupt 109

5.5.1.26 #define OsclErrGeneral 100

5.5.1.27 #define OsclErrInvalidState 113

5.5.1.28 #define OsclErrNoHandler 118

5.5.1.29 #define OsclErrNoMemory 101

5.5.1.30 #define OsclErrNone 0

5.5.1.31 #define OsclErrNoResources 114

5.5.1.32 #define OsclErrNotInstalled 115

5.5.1.33 #define OsclErrNotReady 108

5.5.1.34 #define OsclErrNotSupported 103

5.5.1.35 #define OsclErrOverflow 111

5.5.1.36 #define OsclErrSystemCallFailed 117

5.5.1.37 #define OsclErrThreadContextIncorrect 119

5.5.1.38 #define OsclErrTimeout 110

5.5.1.39 #define OsclErrUnderflow 112

5.5.1.40 #define OsclFailure -1

5.5.1.41 #define OsclPending 1

5.5.1.42 #define OsclSuccess 0

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

5.5.1.44 #define PVERROR_IMP_JUMPS

5.5.1.45 #define PVERRORTRAP_REGISTRY OsclTLSRegistry

5.5.1.46 #define PVERRORTRAP_REGISTRY_ID OSCL_TLS_ID_PVERRORTRAP

5.5.2 Typedef Documentation

5.5.2.1 typedef int32 OsclLeaveCode

Leave Codes

5.5.2.2 typedef int32 OsclReturnCode

Return Codes

5.5.2.3 typedef void(* OsclTrapOperation)(OsclAny*)

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

5.5.3 Function Documentation

5.5.3.1 OSCL_IMPORT_REF int OSCL_GetLastError ()

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

Returns:

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

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

5.5.3.2 OSCL_IMPORT_REF bool OSCL_IsErrnoSupported ()

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

Returns:

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

5.5.3.3 OSCL_IMPORT_REF bool OSCL_SetLastError (int *newVal*)

This function sets the last error code for the system.

Parameters:

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

5.5.3.4 OSCL_IMPORT_REF char* OSCL_StrError (int *errnum*)

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

Parameters:

errnum This value represents the error number to map

Returns:

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

5.6 OSCL IO

Files

- file `oscl_dns.h`

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

- file `oscl_file_cache.h`

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

- file `oscl_file_dir_utils.h`

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

- file `oscl_file_find.h`

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

- file `oscl_file_handle.h`

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

- file `oscl_file_io.h`

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

- file `oscl_file_native.h`

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

- file `oscl_file_server.h`

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

- file `oscl_file_stats.h`

File stats class.

- file `oscl_file_types.h`

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

- file `oscl_socket.h`

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

Data Structures

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

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

Defines

- #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 { EPVDNSGetHostByName }
- enum TPVDNSEvent { EPVDNSSuccess, EPVDNSPending, EPVDNSTimeout, EPVDNSFailure, EPVDNSCancel }
- enum OSCL_FILEMGMT_PERMS { OSCL_FILEMGMT_PERMS_READ = 0x1, OSCL_FILEMGMT_PERMS_WRITE = 0x2, OSCL_FILEMGMT_PERMS_EXECUTE = 0x4 }
- enum OSCL_FILEMGMT_MODES { OSCL_FILEMGMT_MODE_DIR = 0x1 }
- enum OSCL_FILEMGMT_ERR_TYPE { OSCL_FILEMGMT_E_OK = 0, OSCL_FILEMGMT_E_PATH_TOO_LONG, OSCL_FILEMGMT_E_PATH_NOT_FOUND, OSCL_FILEMGMT_E_ALREADY_EXISTS, OSCL_FILEMGMT_E_NOT_EMPTY, OSCL_FILEMGMT_E_PERMISSION_DENIED, OSCL_FILEMGMT_E_NO_MATCH, OSCL_FILEMGMT_E_UNKNOWN, OSCL_FILEMGMT_E_SYS_SPECIFIC, OSCL_FILEMGMT_E_NOT_IMPLEMENTED }
- enum TOsclFileOp { 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_stats (OSCL_FSSTAT *stats, const char *path)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_stats (OSCL_FSSTAT *stats, const oscl_wchar *path)

5.6.1 Define Documentation

5.6.1.1 #define OSCL_FILE_CHAR_PATH_DELIMITER _STRLIT_CHAR("/")

5.6.1.2 #define OSCL_FILE_STATS_LOGGER_NODE "OsclFileStats"

5.6.1.3 #define OSCL_FILE_WCHAR_PATH_DELIMITER _STRLIT("/")

5.6.1.4 #define OSCL_IO_EXTENSION_MAXLEN 512

5.6.1.5 #define OSCL_IO_FILENAME_MAXLEN 512

5.6.2 Typedef Documentation

5.6.2.1 typedef struct oscl_fsstat OSCL_FSSTAT

5.6.2.2 typedef struct oscl_stat_buf OSCL_STAT_BUF

5.6.2.3 typedef FILE* TOsclFileHandle

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

5.6.3 Enumeration Type Documentation

5.6.3.1 enum OSCL_FILEMGMT_ERR_TYPE

Enumeration values:

OSCL_FILEMGMT_E_OK

OSCL_FILEMGMT_E_PATH_TOO_LONG

OSCL_FILEMGMT_E_PATH_NOT_FOUND

```
OSCL_FILEMGMT_E_ALREADY_EXISTS  
OSCL_FILEMGMT_E_NOT_EMPTY  
OSCL_FILEMGMT_E_PERMISSION_DENIED  
OSCL_FILEMGMT_E_NO_MATCH  
OSCL_FILEMGMT_E_UNKNOWN  
OSCL_FILEMGMT_E_SYS_SPECIFIC  
OSCL_FILEMGMT_E_NOT_IMPLEMENTED
```

5.6.3.2 enum OSCL_FILEMGMT_MODES

Enumeration values:

```
OSCL_FILEMGMT_MODE_DIR
```

5.6.3.3 enum OSCL_FILEMGMT_PERMS

Enumeration values:

```
OSCL_FILEMGMT_PERMS_READ  
OSCL_FILEMGMT_PERMS_WRITE  
OSCL_FILEMGMT_PERMS_EXECUTE
```

5.6.3.4 enum TOsclFileOp

Enumeration values:

```
EOsclFileOp_Open  
EOsclFileOp_Close  
EOsclFileOp_Read  
EOsclFileOp_Write  
EOsclFileOp_Seek  
EOsclFileOp_Tell  
EOsclFileOp_Size  
EOsclFileOp_Flush  
EOsclFileOp_EndOfFile  
EOsclFileOp_NativeOpen  
EOsclFileOp_NativeClose  
EOsclFileOp_NativeRead  
EOsclFileOp_NativeWrite  
EOsclFileOp_NativeSeek  
EOsclFileOp_NativeTell  
EOsclFileOp_NativeSize  
EOsclFileOp_NativeFlush  
EOsclFileOp_NativeEndOfFile  
EOsclFileOp_Last
```

5.6.3.5 enum TPVDNSEvent

Enumeration values:

EPVDNSSuccess
EPVDNSPending
EPVDNSTimeout
EPVDNSFailure
EPVDNSCancel

5.6.3.6 enum TPVDNSFxn

Enumeration values:

EPVDNSGetHostByName

5.6.4 Function Documentation

5.6.4.1 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_chdir (const char **path*)

oscl_chdir changes the current directory to the path given

Parameters:

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

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.2 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_chdir (const oscl_wchar **path*)

oscl_chdir changes the current directory to the path given

Parameters:

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

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.3 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_getcwd (char **path*, uint32 *size*)

oscl_getcwd function can be used to determine the full path name of the current directory.

Parameters:

pointer to character buffer to receive the current directory

size size of buffer in characters

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.4 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_getcwd (oscl_wchar *path, uint32 size)

oscl_getcwd function can be used to determine the full path name of the current directory.

Parameters:

pointer to wide character buffer to receive the current directory

size size of buffer in wide characters

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.5 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_mkdir (const char *path)

oscl_mkdir function creates a directory in the path given

Parameters:

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

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.6 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_mkdir (const oscl_wchar *path)

oscl_mkdir function creates a directory in the path given

Parameters:

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

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.7 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rename (const char *oldpath, const char *newpath)

oscl_rmdir removes an empty directory in the path given

Parameters:

character path the full path of the directory to remove.

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.8 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rename (const oscl_wchar * *oldpath*, const oscl_wchar * *newpath*)

oscl_rename function renames a file or directory

Parameters:

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

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

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.9 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rmdir (const char * *path*)

oscl_rmdir removes an empty directory in the path given

Parameters:

character path the full path of the directory to remove.

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.10 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rmdir (const oscl_wchar * *path*)

oscl_rmdir function removes and empty directory in the path given

Parameters:

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

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.11 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_stat (const char * *path*, OSCL_STAT_BUF * *statbuf*)

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

Parameters:

character path the full path of the file to stat.

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.12 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_stat (const oscl_wchar * path, OSCL_STAT_BUF * statbuf)

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

Parameters:

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

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.13 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_statfs (OSCL_FSSTAT * stats, const oscl_wchar * path)

Oscl_StatFS function populates a general structure describing free space available on a filesystem

Parameters:

stats pointer to structure to hold information

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

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.6.4.14 OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_statfs (OSCL_FSSTAT * stats, const char * path)

Oscl_StatFS function populates a general structure describing free space available on a filesystem

Parameters:

stats pointer to structure to hold information

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

Returns:

OSCL_FILEMGMT_ERR_TYPE, see enumeration for this type.

5.7 OSCL Proc

Files

- file [oscl_aostatus.h](#)
Some basic types used with active objects.
- file [oscl_double_list.h](#)
Internal use types for scheduler.
- file [oscl_scheduler_ao.h](#)
Oscl Scheduler user execution object classes.
- file [oscl_scheduler_aobase.h](#)
Oscl Scheduler internal active object classes.
- file [oscl_scheduler_readyq.h](#)
ready q types for oscl scheduler
- file [oscl_scheduler_threadcontext.h](#)
Thread context functions needed by oscl scheduler.
- file [oscl_scheduler_tuneables.h](#)
Tuneable settings for Oscl Scheduler.
- file [oscl_scheduler_types.h](#)
Scheduler common types include file.

Data Structures

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

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

Defines

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

Typedefs

- typedef PVActiveBase * TOsclReady

Enumerations

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

Functions

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

Variables

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

5.7.1 Define Documentation

5.7.1.1 #define OSCL_PERF_SUMMARY_LOGGING 0

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

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

5.7.1.3 #define PV_SCHED_CHECK_Q 0

5.7.1.4 #define PV_SCHED_ENABLE_AO_STATS 1

5.7.1.5 #define PV_SCHED_ENABLE_LOOP_STATS 0

5.7.1.6 #define PV_SCHED_ENABLE_PERF_LOGGING 1

5.7.1.7 #define PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS 1

5.7.1.8 #define PV_SCHED_FAIR_SCHEDULING 1

5.7.1.9 #define PV_SCHED_LOG_Q 0

5.7.1.10 #define PVEEXECNAMELEN 30

5.7.1.11 #define PVSCEDNAMELEN 30

PV Scheduler class

5.7.1.12 #define QUE_ITER_BEGIN(_type, _qname)

Value:

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

5.7.1.13 #define QUE_ITER_END(_qname)

Value:

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

5.7.2 Typedef Documentation

5.7.2.1 `typedef PVActiveBase* TOsclReady`

5.7.3 Enumeration Type Documentation

5.7.3.1 `enum TPVThreadContext`

Thread context type

Enumeration values:

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

5.7.4 Function Documentation

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

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

5.7.5 Variable Documentation

5.7.5.1 `const int32 OSCL_REQUEST_ERR_CANCEL = (-1)`

5.7.5.2 `const int32 OSCL_REQUEST_ERR_GENERAL = (-2)`

5.7.5.3 `const int32 OSCL_REQUEST_ERR_NONE = 0`

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

5.8 OSCL Init

Files

- file [oscl_init.h](#)

Global oscl initialization.

Data Structures

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

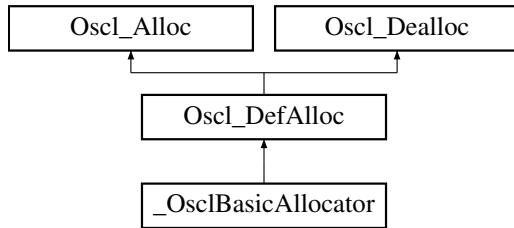
Chapter 6

oscl Data Structure Documentation

6.1 _OsclBasicAllocator Class Reference

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

Inheritance diagram for _OsclBasicAllocator::



Public Methods

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

6.1.1 Detailed Description

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

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

6.1.2 Constructor & Destructor Documentation

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

6.1.3 Member Function Documentation

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

Implements [Oscl_DefAlloc](#).

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

Implements [Oscl_DefAlloc](#).

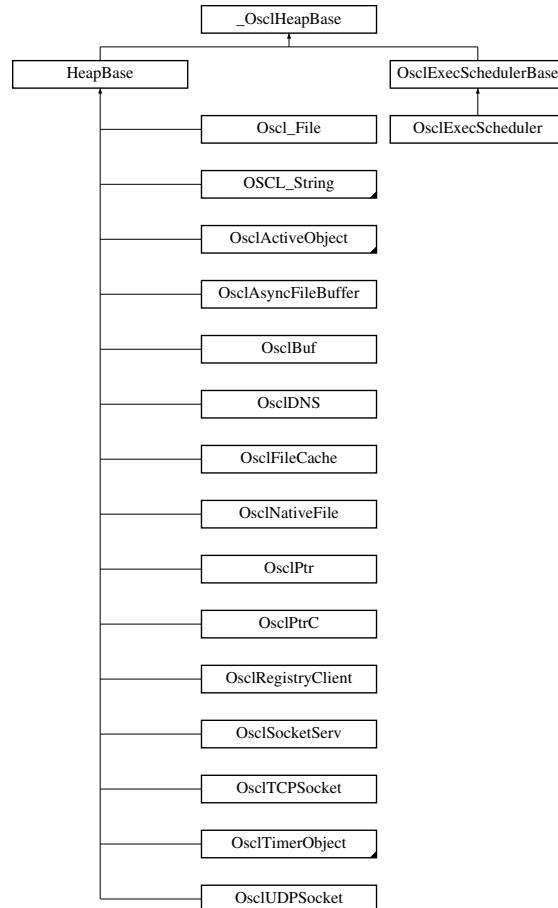
The documentation for this class was generated from the following file:

- [oscl_base_alloc.h](#)

6.2 _OsclHeapBase Class Reference

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

Inheritance diagram for _OsclHeapBase::



Public Methods

- virtual ~_OsclHeapBase ()

Protected Methods

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

Friends

- class [PVCleanupStack](#)

6.2.1 Detailed Description

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

6.2.2 Constructor & Destructor Documentation

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

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

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

6.2.3 Friends And Related Function Documentation

6.2.3.1 `friend class PVCleanupStack [friend]`

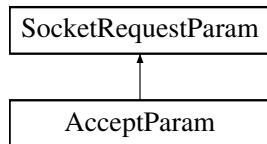
The documentation for this class was generated from the following file:

- [oscl_heapbase.h](#)

6.3 AcceptParam Class Reference

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

Inheritance diagram for AcceptParam::



Public Methods

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

Data Fields

- [OsclSocketI * iBlankSocket](#)

6.3.1 Constructor & Destructor Documentation

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

6.3.2 Field Documentation

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

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

- [oscl_socket_request.h](#)

6.4 allocator Class Reference

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

6.4.1 Detailed Description

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

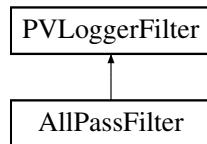
The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

6.5 AllPassFilter Class Reference

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

Inheritance diagram for AllPassFilter::



Public Types

- `typedef PVLoggerFilter::message_id_type message_id_type`
- `typedef PVLoggerFilter::log_level_type log_level_type`
- `typedef PVLoggerFilter::filter_status_type filter_status_type`

Public Methods

- `AllPassFilter ()`
- `virtual ~AllPassFilter ()`
- `filter_status_type FilterString (char *tag, message_id_type msgID, log_level_type level)`
- `filter_status_type FilterOpaqueMessge (char *tag, message_id_type msgID, log_level_type level)`

6.5.1 Detailed Description

Example filter that allows all messages to be logged.

6.5.2 Member Typedef Documentation

6.5.2.1 `typedef PVLoggerFilter::filter_status_type AllPassFilter::filter_status_type`

Reimplemented from [PVLoggerFilter](#).

6.5.2.2 `typedef PVLoggerFilter::log_level_type AllPassFilter::log_level_type`

Reimplemented from [PVLoggerFilter](#).

6.5.2.3 `typedef PVLoggerFilter::message_id_type AllPassFilter::message_id_type`

Reimplemented from [PVLoggerFilter](#).

6.5.3 Constructor & Destructor Documentation

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

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

6.5.4 Member Function Documentation

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

Implements [PVLoggerFilter](#).

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

Implements [PVLoggerFilter](#).

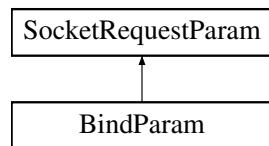
The documentation for this class was generated from the following file:

- [pvlogger_accessories.h](#)

6.6 BindParam Class Reference

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

Inheritance diagram for BindParam::



Public Methods

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

Data Fields

- [OsclNetworkAddress iAddr](#)

6.6.1 Constructor & Destructor Documentation

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

6.6.2 Field Documentation

6.6.2.1 [OsclNetworkAddress BindParam::iAddr](#)

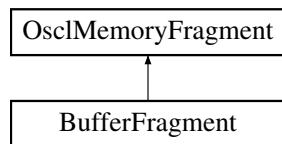
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

6.7 BufferFragment Class Reference

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

Inheritance diagram for BufferFragment::



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

- [oscl_media_data.h](#)

6.8 BufferMgr Class Reference

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

Public Methods

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

6.8.1 Constructor & Destructor Documentation

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

6.8.2 Member Function Documentation

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

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

- [oscl_media_data.h](#)

6.9 BufferState Class Reference

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

Public Methods

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

6.9.1 Constructor & Destructor Documentation

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

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

6.9.2 Member Function Documentation

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

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

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

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

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

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

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

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

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

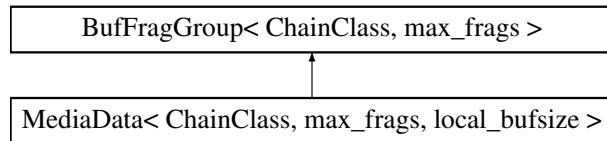
The documentation for this class was generated from the following file:

- `oscl_media_data.h`

6.10 BufFragGroup< ChainClass, max_frags > Class Template Reference

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

Inheritance diagram for BufFragGroup< ChainClass, max_frags >::



Public Methods

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

Protected Methods

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

Protected Attributes

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

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

6.10.1 Constructor & Destructor Documentation

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

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

6.10.2 Member Function Documentation

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

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

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

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

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

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

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

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

6.10.3 Field Documentation

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

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

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

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

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

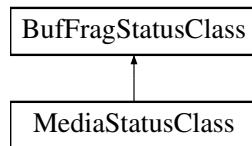
The documentation for this class was generated from the following file:

- [oscl_media_data.h](#)

6.11 BufFragStatusClass Class Reference

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

Inheritance diagram for BufFragStatusClass::



Public Types

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

6.11.1 Member Enumeration Documentation

6.11.1.1 enum BufFragStatusClass::status_t

Enumeration values:

`BFG_SUCCESS`
`TOO_MANY_FRAGS`
`NOT_ENOUGH_SPACE`
`EMPTY_FRAGMENT`
`NULL_INPUT`
`FIXED_FRAG_LOC_FULL`
`INTERNAL_ERROR`
`INVALID_ID`

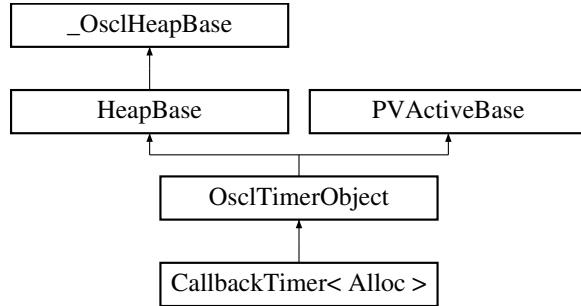
The documentation for this class was generated from the following file:

- [oscl_media_status.h](#)

6.12 CallbackTimer< Alloc > Class Template Reference

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

Inheritance diagram for CallbackTimer< Alloc >::



Public Methods

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

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

6.12.1 Constructor & Destructor Documentation

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

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

6.12.2 Member Function Documentation

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

Handles an active object's request completion event.

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

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

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

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

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

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

Implements [PVActiveBase](#).

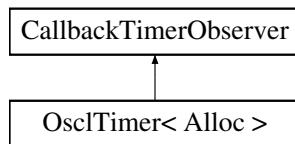
The documentation for this class was generated from the following file:

- [oscl_timer.h](#)

6.13 CallbackTimerObserver Class Reference

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

Inheritance diagram for CallbackTimerObserver::



Public Methods

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

6.13.1 Constructor & Destructor Documentation

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

6.13.2 Member Function Documentation

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

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

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

- [oscl_timer.h](#)

6.14 CFastRep Class Reference

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

Public Methods

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

Data Fields

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

6.14.1 Detailed Description

For internal use only– fast string representation

6.14.2 Constructor & Destructor Documentation

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

6.14.3 Member Function Documentation

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

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

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

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

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

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

6.14.4 Field Documentation

6.14.4.1 `OsclAny* CFastRep::buffer`

6.14.4.2 `uint32 CFastRep::maxsize`

6.14.4.3 `uint32 CFastRep::size`

6.14.4.4 `bool CFastRep::writable`

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

- `oscl_string_rep.h`

6.15 CHeapRep Class Reference

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

Public Methods

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

Static Public Methods

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

Data Fields

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

6.15.1 Detailed Description

For internal use only– heap string representation

6.15.2 Constructor & Destructor Documentation

6.15.2.1 `CHheapRep::CHheapRep () [inline]`

6.15.3 Member Function Documentation

6.15.3.1 `OSCL_IMPORT_REF void CHheapRep::add_ref ()`

6.15.3.2 `OSCL_IMPORT_REF bool CHheapRep::append (uint32, const oscl_wchar *, uint32, const oscl_wchar *, Oscl_DefAlloc &)`

6.15.3.3 `OSCL_IMPORT_REF bool CHheapRep::append (uint32, const char *, uint32, const char *, Oscl_DefAlloc &)`

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

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

6.15.3.6 `OSCL_IMPORT_REF void CHheapRep::assign (CHheapRep *&, CHheapRep *, Oscl_DefAlloc &) [static]`

6.15.3.7 `OSCL_IMPORT_REF void CHheapRep::remove_ref (Oscl_DefAlloc &)`

6.15.3.8 `OSCL_IMPORT_REF bool CHheapRep::set (uint32, const oscl_wchar *, Oscl_DefAlloc &)`

6.15.3.9 `OSCL_IMPORT_REF bool CHheapRep::set (uint32, const char *, Oscl_DefAlloc &)`

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

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

6.15.4 Field Documentation

6.15.4.1 `OsclAny* CHheapRep::buffer`

6.15.4.2 `uint32 CHheapRep::maxsize`

6.15.4.3 `uint32 CHheapRep::refcount`

6.15.4.4 `uint32 CHheapRep::size`

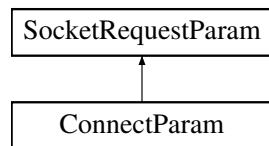
The documentation for this class was generated from the following file:

- `oscl_string_rep.h`

6.16 ConnectParam Class Reference

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

Inheritance diagram for ConnectParam::



Public Methods

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

Data Fields

- [OsclNetworkAddress iAddr](#)

6.16.1 Constructor & Destructor Documentation

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

6.16.2 Field Documentation

6.16.2.1 OsclNetworkAddress ConnectParam::iAddr

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

- [oscl_socket_request.h](#)

6.17 CStackRep Class Reference

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

Public Methods

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

Data Fields

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

6.17.1 Detailed Description

For internal use only– stack string representation

6.17.2 Constructor & Destructor Documentation

6.17.2.1 CStackRep::CStackRep () [inline]

6.17.3 Member Function Documentation

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

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

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

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

6.17.4 Field Documentation

6.17.4.1 OsclAny* CStackRep::buffer

6.17.4.2 uint32 CStackRep::maxsize

6.17.4.3 uint32 CStackRep::size

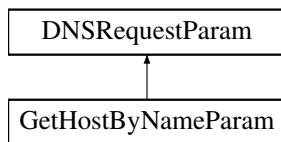
The documentation for this class was generated from the following file:

- [oscl_string_rep.h](#)

6.18 DNSRequestParam Class Reference

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

Inheritance diagram for DNSRequestParam::



Public Methods

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

Data Fields

- TPVDNSFx_n iFx_n
- OsclDNSRequest * iDNSRequest

Protected Methods

- DNSRequestParam (TPVDNSFx_n aFx_n)

Protected Attributes

- uint32 iRefCount

6.18.1 Constructor & Destructor Documentation

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

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

6.18.2 Member Function Documentation

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

Implemented in [GetHostByNameParam](#).

6.18.2.2 void DNSRequestParam::InThread ()

6.18.2.3 void DNSRequestParam::RemoveRef ()

6.18.3 Field Documentation

6.18.3.1 OsclDNSRequest* DNSRequestParam::iDNSRequest

6.18.3.2 TPVDNSFxn DNSRequestParam::iFxn

6.18.3.3 uint32 DNSRequestParam::iRefCount [protected]

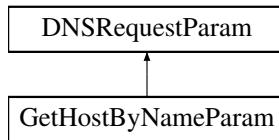
The documentation for this class was generated from the following file:

- [oscl_dns_param.h](#)

6.19 GetHostByNameParam Class Reference

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

Inheritance diagram for GetHostByNameParam::



Public Methods

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

Static Public Methods

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

Data Fields

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

6.19.1 Constructor & Destructor Documentation

6.19.1.1 [GetHostByNameParam::~GetHostByNameParam \(\)](#)

6.19.2 Member Function Documentation

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

6.19.2.2 [void GetHostByNameParam::Destroy \(\) \[virtual\]](#)

Implements [DNSRequestParam](#).

6.19.3 Field Documentation

6.19.3.1 [OsclNetworkAddress* GetHostByNameParam::iAddr](#)

6.19.3.2 [char* GetHostByNameParam::iName](#)

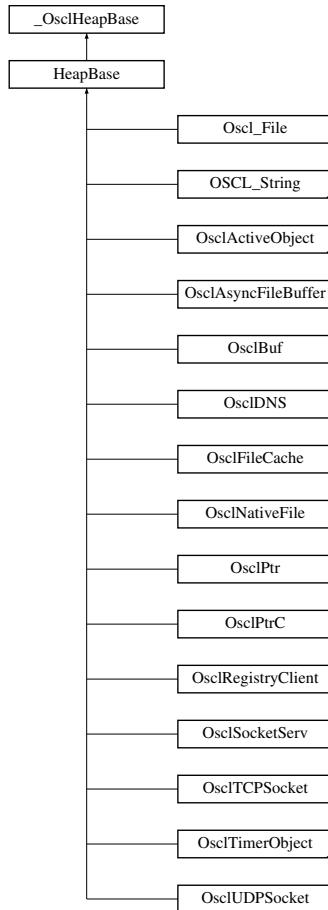
The documentation for this class was generated from the following file:

- [oscl_dns_param.h](#)

6.20 HeapBase Class Reference

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

Inheritance diagram for HeapBase::



Public Methods

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

6.20.1 Detailed Description

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

HeapBase has overloaded new and delete operators.

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

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

6.20.2 Constructor & Destructor Documentation

6.20.2.1 HeapBase::HeapBase () [inline]

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

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

- [oscl_mem.h](#)

6.21 internalLeave Class Reference

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

Data Fields

- int a

6.21.1 Field Documentation

6.21.1.1 int internalLeave::a

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

- [oscl_error_imp_cppexceptions.h](#)

6.22 LinkedListElement< LLClass > Class Template Reference

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

Public Methods

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

Data Fields

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

6.22.1 Detailed Description

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

Linked List Element Class

6.22.2 Constructor & Destructor Documentation

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

6.22.3 Field Documentation

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

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

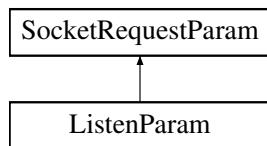
The documentation for this class was generated from the following file:

- [oscl_linked_list.h](#)

6.23 ListenParam Class Reference

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

Inheritance diagram for ListenParam::



Public Methods

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

Data Fields

- uint32 [iQSize](#)

6.23.1 Constructor & Destructor Documentation

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

6.23.2 Field Documentation

6.23.2.1 [uint32 ListenParam::iQSize](#)

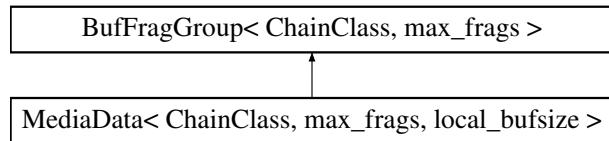
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

6.24 `MediaData< ChainClass, max_frags, local_bufsize >` Class Template Reference

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

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



Public Methods

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

Protected Methods

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

Protected Attributes

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

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

6.24.1 Constructor & Destructor Documentation

6.24.1.1 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> MediaData< ChainClass, max_frags, local_bufsize >::MediaData () [inline]

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

6.24.2 Member Function Documentation

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

6.24.2.2 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> virtual void MediaData< ChainClass, max_frags, local_bufsize >::Clear () [inline, virtual]

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

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

- 6.24.2.3 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> uint32 MediaData<ChainClass, max_frags, local_bufsize >::GetAvailableBufferSize () const [inline]
- 6.24.2.4 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> uint32 MediaData<ChainClass, max_frags, local_bufsize >::GetLocalBufsize () const [inline]
- 6.24.2.5 template<class ChainClass, uint32 max_frags, uint32 local_bufsize>
MediaStatusClass::status_t MediaData<ChainClass, max_frags, local_bufsize >::GetLocalFragment (**BufferFragment** & *fragment*) [inline]
- 6.24.2.6 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> **BufferFragment*** MediaData<ChainClass, max_frags, local_bufsize >::GetMediaFragment (const uint32 *idx*) [inline]
- 6.24.2.7 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> int MediaData<ChainClass, max_frags, local_bufsize >::GetMediaSize () const [inline]
- 6.24.2.8 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> uint32 MediaData<ChainClass, max_frags, local_bufsize >::GetNumMediaFrags (const uint32 *idx*) const [inline]
- 6.24.2.9 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> **MediaTimestamp** MediaData<ChainClass, max_frags, local_bufsize >::GetTimestamp () const [inline]
- 6.24.2.10 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> bool MediaData<ChainClass, max_frags, local_bufsize >::IsLocalData (const **OsclMemoryFragment** & *frag*) const [inline]
- 6.24.2.11 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> void MediaData<ChainClass, max_frags, local_bufsize >::SetTimestamp (**MediaTimestamp** *in_timestamp*) [inline]

6.24.3 Field Documentation

- 6.24.3.1 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> uint32 MediaData<ChainClass, max_frags, local_bufsize >::available_localbuf [protected]
- 6.24.3.2 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> uint8 MediaData<ChainClass, max_frags, local_bufsize >::localbuf[local_bufsize] [protected]
- 6.24.3.3 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> int MediaData<ChainClass, max_frags, local_bufsize >::num_reserved.fragments [protected]
- 6.24.3.4 template<class ChainClass, uint32 max_frags, uint32 local_bufsize> **MediaTimestamp** MediaData<ChainClass, max_frags, local_bufsize >::timestamp [protected]

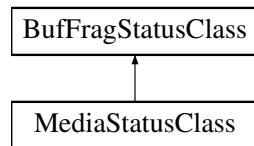
The documentation for this class was generated from the following file:

- [oscl_media_data.h](#)

6.25 MediaStatusClass Class Reference

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

Inheritance diagram for MediaStatusClass::



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

- [oscl_media_status.h](#)

6.26 MemAllocator< T > Class Template Reference

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

Public Types

- `typedef T * pointer`

Public Methods

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

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

6.26.1 Member Typedef Documentation

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

6.26.2 Constructor & Destructor Documentation

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

6.26.3 Member Function Documentation

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

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

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

- [oscl_media_data.h](#)

6.27 MM_AllocBlockFence Struct Reference

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

Public Methods

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

Data Fields

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

6.27.1 Constructor & Destructor Documentation

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

6.27.2 Member Function Documentation

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

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

6.27.3 Field Documentation

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

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

- [oscl_mem_audit_internals.h](#)

6.28 MM_AllocBlockHdr Struct Reference

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

Public Methods

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

Data Fields

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

Static Public Attributes

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

6.28.1 Constructor & Destructor Documentation

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

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

6.28.2 Member Function Documentation

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

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

6.28.3 Field Documentation

6.28.3.1 [uint32 MM_AllocBlockHdr::pad](#)

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

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

6.28.3.4 [uint32 MM_AllocBlockHdr::size](#)

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

- [oscl_mem_audit_internals.h](#)

6.29 MM_AllocInfo Struct Reference

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

Public Methods

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

Data Fields

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

6.29.1 Constructor & Destructor Documentation

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

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

6.29.2 Member Function Documentation

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

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

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

6.29.3 Field Documentation

6.29.3.1 `uint32 MM_AllocInfo::allocNum`

6.29.3.2 `bool MM_AllocInfo::bSetFailure`

6.29.3.3 `uint32 MM_AllocInfo::lineNo`

6.29.3.4 `char* MM_AllocInfo::pFileName`

6.29.3.5 `void* MM_AllocInfo::pMemBlock`

6.29.3.6 `OsclMemStatsNode* MM_AllocInfo::pStatsNode`

6.29.3.7 `uint32 MM_AllocInfo::size`

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

- [oscl_mem_audit.h](#)

6.30 MM_AllocNode Struct Reference

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

Public Methods

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

Data Fields

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

6.30.1 Constructor & Destructor Documentation

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

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

6.30.2 Member Function Documentation

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

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

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

6.30.3 Field Documentation

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

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

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

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

- [oscl_mem_audit.h](#)

6.31 MM_AllocQueryInfo Struct Reference

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

Data Fields

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

6.31.1 Field Documentation

6.31.1.1 uint32 MM_AllocQueryInfo::allocNum

6.31.1.2 char MM_AllocQueryInfo::fileName[MM_ALLOC_MAX_QUERY_FILENAME_LEN]

6.31.1.3 uint32 MM_AllocQueryInfo::lineNo

6.31.1.4 const void* MM_AllocQueryInfo::pMemBlock

6.31.1.5 uint32 MM_AllocQueryInfo::size

6.31.1.6 char MM_AllocQueryInfo::tag[MM_ALLOC_MAX_QUERY_TAG_LEN]

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

- [oscl_mem_audit.h](#)

6.32 MM_Audit_Imp Class Reference

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

Public Methods

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

- uint32 `getTagActualSize` (const char *tagIn)
- bool `isSetFailure` (const char *tagIn)
- bool `isSetFailure` (OsclMemStatsNode *statsNode)
- bool `validate_all_heap` ()

Static Public Methods

- bool `validate` (void *ptrIn)
- OsclMemAudit * `getAuditRoot` (void *ptrIn)
- uint32 `getSize` (void *ptrIn)

6.32.1 Constructor & Destructor Documentation

6.32.1.1 MM_Audit_Imp::MM_Audit_Imp ()

Constructor, create the root node in statistics table

6.32.1.2 MM_Audit_Imp::~MM_Audit_Imp ()

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

6.32.2 Member Function Documentation

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

Returns:

true if operation succeeds;

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

Returns:

true if operation succeeds;

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

Returns:

audit root pointer.

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

Returns:

original block size. leaves if bad pointer.

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

the size of the truncated tag; 0 means NO truncation

6.32.2.6 bool MM_Audit_Imp::isSetFailure (OsclMemStatsNode * statsNode)**6.32.2.7 bool MM_Audit_Imp::isSetFailure (const char * tagIn)****Returns:**

true if operation succeeds;

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

a valid tag; NULL will be converted into root tag

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

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

Parameters:

tagIn input tag

Returns:

true if operation succeeds;

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

The following are APIs t __nothrow_ / const __nothrow_

Returns:

the memory pointer if operation succeeds.

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

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

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

true if operation succeeds;

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

API to get the current allocation number

Returns:

the current allocation number

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

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

Parameters:

output_array the array where the data will be written

max_array_size the max number of output array elements

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

Returns:

the number of valid nodes in the output array

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

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

Parameters:

tagIn input tag

Returns:

true if operation succeeds;

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

API to get the operating mode of the mm_audit class.

6.32.2.17 uint32 MM_Audit_Imp::MM_GetNumAllocNodes () [inline]

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

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

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

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

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

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

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

6.32.2.21 OSCL_IMPORT_REF const OsclMemStatsNode* MM_Audit_Imp::MM_GetRootNode () [inline]**6.32.2.22 OSCL_IMPORT_REF MM_Stats_t* MM_Audit_Imp::MM_GetStats (const char *const tagIn)**

API to get memory statistics through context string(tag)

Returns:

statistics pointer if operation succeeds

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

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

Returns:

statistics pointer array and actual number of nodes if operation succeeds

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

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

Parameters:

tagIn input tag

Returns:

pointer to [OsclMemStatsNode](#) which should be passed to MM_allocate

6.32.2.25 OSCL_IMPORT_REF uint32 MM_Audit_Imp::MM_GetTreeNodes (const char * tagIn)

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

Parameters:

tagIn input tag

Returns:

the number of tree nodes ; 0 means no tag node

**6.32.2.26 OSCL_IMPORT_REF void MM_Audit_Imp::MM_ReleaseAllocNodeInfo
([MM_AllocQueryInfo](#) * *info*)**

**6.32.2.27 OSCL_IMPORT_REF bool MM_Audit_Imp::MM_SetFailurePoint (const char * *tagIn*,
uint32 *alloc_number*)**

API to insert allocation failure deterministically according to allocation number associated with tag

Parameters:

tagIn input tag

alloc_number allocation number associated with tag

Returns:

true if operation succeeds;

6.32.2.28 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetMode (uint32 *inMode*)

API to set the operating mode of the mm_audit class.

6.32.2.29 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetPostfillPattern (uint8 *pattern*)

API to set the postfill pattern.

6.32.2.30 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetPrefillPattern (uint8 *pattern*)

API to set the prefill pattern.

6.32.2.31 OSCL_IMPORT_REF void MM_Audit_Imp::MM_SetTagLevel (uint32 *level*)

API to set the maximum tag level,i.e. tag level for a.b.c.d = 4

Parameters:

level input tag level to be set

**6.32.2.32 OSCL_IMPORT_REF void MM_Audit_Imp::MM_UnsetFailurePoint (const char *
tagIn)**

API to cancel the allocation failure point associated with tag

Parameters:

tagIn input tag

6.32.2.33 OSCL_IMPORT_REF bool MM_Audit_Imp::MM_Validate (const void * *ptrIn*)

API to check the input pointer is a valid pointer to a chunk of memory

Parameters:

ptrIn input pointer to be validated

Returns:

true if operation succeeds;

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

6.32.2.35 `bool MM_Audit_Imp::pruneSubtree (OsclMemStatsNode * pNode)`

Returns:

true if operation succeeds;

6.32.2.36 `void MM_Audit_Imp::removeALLAllocNodes ()`

6.32.2.37 `OsclMemStatsNode* MM_Audit_Imp::removeAllocNode (void * pMemBlockIn, uint32 & size)`

Returns:

true if operation succeeds;

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

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

Returns:

the length of a immediate parent tag for the input tag

6.32.2.40 `bool MM_Audit_Imp::updateStatsNode (OsclMemStatsNode * pCurrStatsNode, const MM_Stats_t & pDelta, bool bAdd)`

Returns:

true if operation succeeds;

6.32.2.41 `bool MM_Audit_Imp::updateStatsNodeInFailure (OsclMemStatsNode * pStatsNode)`

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

Returns:

true if operation succeeds;

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

Returns:

true if operation succeeds;

6.32.2.44 bool MM_Audit_Imp::validate_all_heap ()**Returns:**

true if operation succeeds;

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

- [oscl_mem_audit.h](#)

6.33 MM_AuditOverheadStats Struct Reference

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

Data Fields

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

6.33.1 Field Documentation

6.33.1.1 uint32 MM_AuditOverheadStats::per_allocation_overhead

6.33.1.2 uint32 MM_AuditOverheadStats::stats_overhead

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

- [oscl_mem_audit.h](#)

6.34 MM_FailInsertParam Struct Reference

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

Public Methods

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

Data Fields

- [uint32 nAllocNum](#)
- [uint16 xsubi \[3\]](#)

6.34.1 Constructor & Destructor Documentation

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

6.34.2 Member Function Documentation

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

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

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

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

6.34.3 Field Documentation

6.34.3.1 [uint32 MM_FailInsertParam::nAllocNum](#)

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

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

- [oscl_mem_audit.h](#)

6.35 MM_Stats_CB Struct Reference

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

Public Methods

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

Data Fields

- [const char * tag](#)
- [const MM_Stats_t * pStats](#)
- [uint32 num_child_nodes](#)

6.35.1 Constructor & Destructor Documentation

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

6.35.2 Member Function Documentation

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

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

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

6.35.3 Field Documentation

6.35.3.1 [uint32 MM_Stats_CB::num_child_nodes](#)

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

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

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

- [oscl_mem_audit.h](#)

6.36 MM_Stats_t Struct Reference

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

Public Methods

- [MM_Stats_t \(\)](#)
- [MM_Stats_t \(uint32 sizeIn\)](#)
- [void reset \(\)](#)
- [void update \(const MM_Stats_t &delta, bool add\)](#)
- [void * operator new \(oscl_memsize_t size\)](#)
- [void * operator new \(oscl_memsize_t size, MM_Stats_t *ptr\)](#)
- [void operator delete \(void *ptr\) throw \(\)](#)

Data Fields

- [uint32 numBytes](#)
- [uint32 peakNumBytes](#)
- [uint32 numAllocs](#)
- [uint32 peakNumAllocs](#)
- [uint32 numAllocFails](#)
- [uint32 totalNumAllocs](#)
- [uint32 totalNumBytes](#)

6.36.1 Constructor & Destructor Documentation

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

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

6.36.2 Member Function Documentation

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

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

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

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

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

6.36.3 Field Documentation

6.36.3.1 `uint32 MM_Stats_t::numAllocFails`

6.36.3.2 `uint32 MM_Stats_t::numAllocs`

6.36.3.3 `uint32 MM_Stats_t::numBytes`

6.36.3.4 `uint32 MM_Stats_t::peakNumAllocs`

6.36.3.5 `uint32 MM_Stats_t::peakNumBytes`

6.36.3.6 `uint32 MM_Stats_t::totalNumAllocs`

6.36.3.7 `uint32 MM_Stats_t::totalNumBytes`

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

- [oscl_mem_audit.h](#)

6.37 NTPTime Class Reference

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

```
#include <oscl_time.h>
```

Public Methods

- **OSCL_COND_IMPORT_REF NTPTime ()**
The default constructor creates an NTPTime instance representing the current system time.
- **OSCL_COND_IMPORT_REF NTPTime (const NTPTime &src)**
Copy constructor to create a new NTPTime from an existing one.
- **OSCL_COND_IMPORT_REF NTPTime (const uint32 seconds)**
Construct an NTPTime from a uint32.
- **OSCL_COND_IMPORT_REF NTPTime (const int32 seconds)**
Construct an NTPTime from a int.
- **OSCL_COND_IMPORT_REF NTPTime (const TimeValue &t)**
Construct a NTPTime instance from a TimeValue instance.
- **OSCL_COND_IMPORT_REF NTPTime (const uint64 value)**
Construct a NTPTime instance from a uint64 value.
- **OSCL_COND_IMPORT_REF NTPTime & operator= (uint32 newval)**
The assignment operator for a 32 bit integer.
- **OSCL_COND_IMPORT_REF NTPTime & operator= (uint64 newval)**
The assignment operator for a 64 bit integer.
- **OSCL_COND_IMPORT_REF NTPTime & operator+= (uint64 val)**
The += operator is used to add a 64 bit 32.32 value to an existing NTPTime value.
- **OSCL_COND_IMPORT_REF NTPTime operator- (const NTPTime &npt) const**
The - operator allows subtraction of one NTPTime value from another. This is useful to measure an interval.
- **void set_from_system_time (const uint32 systemtime)**
This method converts a 32-bit system time to NTP time.
- **OSCL_COND_IMPORT_REF uint32 get_middle32 () const**
Grab the middle 32 bits of the 64 bit 32.32 representation.
- **OSCL_COND_IMPORT_REF uint32 get_upper32 () const**
This method returns the upper 32 bits of the 32.32 representation.
- **OSCL_COND_IMPORT_REF uint32 get_lower32 () const**
This method returns the lower 32 bits of the 32.32 representation.

- int32 [to_system_time \(\) const](#)

This method converts the ntp time value to system time.

- OSCL_COND_IMPORT_REF [uint64 get_value \(\) const](#)

This method returns the 32.32 ntp representation.

- OSCL_IMPORT_REF int [set_to_current_time \(\)](#)

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

6.37.1 Detailed Description

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

The NTPTime class: Conversion to/from Unix (epoch at 0h Jan. 1, 1970) amount to addition/subtraction of 2208988800. A single 64 bit value is used to represent the time. This value represents the number of seconds since 0h (UTC) Jan. 1, 1900. There is an implied binary point between the upper 32 bits and lower 32 bits (this is referred to as a 32.32 fractional representation). For example a binary value of 00000000 00000000 00000011 10000000 00000000 00000000 00000000 represents 3.5 seconds since Jan 1, 1900.

6.37.2 Constructor & Destructor Documentation

6.37.2.1 OSCL_COND_IMPORT_REF NTPTime::NTPTime ()

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

6.37.2.2 OSCL_COND_IMPORT_REF NTPTime::NTPTime (const NTPTime & src)

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

6.37.2.3 OSCL_COND_IMPORT_REF NTPTime::NTPTime (const uint32 seconds)

Construct an NTPTime from a uint32.

Parameters:

seconds The uint32 input represents the number of seconds since Jan. 1, 1900.

6.37.2.4 OSCL_COND_IMPORT_REF NTPTime::NTPTime (const int32 seconds)

Construct an NTPTime from a int.

Parameters:

seconds The int input represents the number of seconds since Jan. 1, 1900.

6.37.2.5 OSCL_COND_IMPORT_REF NTPTime::NTPTime (const TimeValue & t)

Construct a NTPTime instance from a TimeValue instance.

This constructor creates an NTPTime value representing the same absolute time as the TimeValue parameter.

Parameters:

t A reference to a TimeValue object.

6.37.2.6 OSCL_COND_IMPORT_REF NTPTime::NTPTime (const uint64 value)

Construct a NTPTime instance from a uint64 value.

Parameters:

value A 64 bit integer argument which is used as the ntp 32.32 fractional representation.

6.37.3 Member Function Documentation

6.37.3.1 OSCL_COND_IMPORT_REF uint32 NTPTime::get_lower32 ()

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

6.37.3.2 OSCL_COND_IMPORT_REF uint32 NTPTime::get_middle32 ()

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

6.37.3.3 OSCL_COND_IMPORT_REF uint32 NTPTime::get_upper32 ()

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

6.37.3.4 OSCL_COND_IMPORT_REF uint64 NTPTime::get_value ()

This method returns the 32.32 ntp representation.

6.37.3.5 OSCL_COND_IMPORT_REF NTPTime& NTPTime::operator+= (uint64 val)

The += operator is used to add a 64 bit 32.32 value to an existing NTPTime value.

Parameters:

val The 64 bit 32.32 value to add to this object's value.

6.37.3.6 OSCL_COND_IMPORT_REF NTPTime NTPTime::operator- (const NTPTime & npt) const

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

Parameters:

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

6.37.3.7 OSCL_COND_IMPORT_REF NTPTime& NTPTime::operator= (*uint64 newval*)

The assignment operator for a 64 bit integer.

Parameters:

newval A 64 bit value which represents the 32.32 fractional representation of the ntp time.

6.37.3.8 OSCL_COND_IMPORT_REF NTPTime& NTPTime::operator= (*uint32 newval*)

The assignment operator for a 32 bit integer.

Parameters:

newval A 32 bit integer representing the upper 32 bits of the 32.32 NTP time (e.g. the number of whole seconds since Jan 1, 1900 UTC).

6.37.3.9 void NTPTime::set_from_system_time (*const uint32 systemtime*)

This method converts a 32-bit system time to NTP time.

This method sets the value of the NTPTime instance to the absolute time represented by the 32 bit input argument.

Parameters:

systemtime This 32-bit value is interpreted as the number of seconds since the unix epoch Jan. 1 1970.

6.37.3.10 OSCL_IMPORT_REF int NTPTime::set_to_current_time ()

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

6.37.3.11 int32 NTPTime::to_system_time ()

This method converts the ntp time value to system time.

This method returns a 32 bit value representing the same absolute time as the NTP time value, but expressed as whole seconds since the unix epoch. The fractional part of the ntp value is discarded.

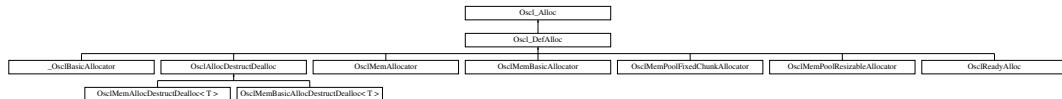
The documentation for this class was generated from the following file:

- [oscl_time.h](#)

6.38 Oscl_Alloc Class Reference

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

Inheritance diagram for Oscl_Alloc::



Public Methods

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

6.38.1 Member Function Documentation

6.38.1.1 virtual [OsclAny* Oscl_Alloc::allocate \(const uint32 size\)](#) [pure virtual]

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

6.38.1.2 virtual [OsclAny* Oscl_Alloc::allocate_fl \(const uint32 size, const char *file_name, const int line_num\)](#) [inline, virtual]

Reimplemented in [Oscl_DefAlloc](#), [OsclMemAllocator](#), [OsclMemAllocDestructDealloc< T >](#), and [OsclReadyAlloc](#).

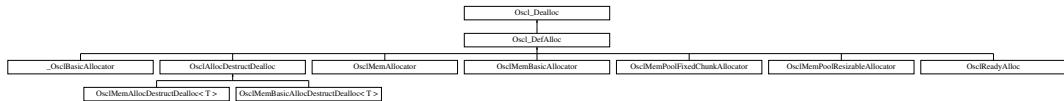
The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

6.39 Oscl_Dealloc Class Reference

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

Inheritance diagram for Oscl_Dealloc::



Public Methods

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

6.39.1 Member Function Documentation

6.39.1.1 virtual void Oscl_Dealloc::deallocate (OsclAny *p) [pure virtual]

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

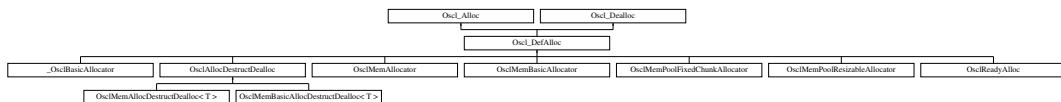
The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

6.40 Oscl_DefAlloc Class Reference

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

Inheritance diagram for Oscl_DefAlloc::



Public Methods

- virtual [OsclAny * allocate \(const uint32 size\)=0](#)
- virtual [OsclAny * allocate_fl \(const uint32 size, const char *file_name, const int line_num\)](#)
- virtual void [deallocate \(OsclAny *p\)=0](#)

6.40.1 Member Function Documentation

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

Implements [Oscl_Alloc](#).

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

6.40.1.2 virtual [OsclAny* Oscl_DefAlloc::allocate_fl \(const uint32 size, const char * file_name, const int line_num\)](#) [inline, virtual]

Reimplemented from [Oscl_Alloc](#).

Reimplemented in [OsclMemAllocator](#), [OsclMemAllocDestructDealloc< T >](#), and [OsclReadyAlloc](#).

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

Implements [Oscl_Dealloc](#).

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

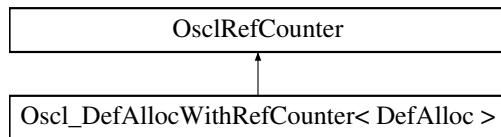
The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

6.41 Oscl_DefAllocWithRefCounter< DefAlloc > Class Template Reference

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

Inheritance diagram for Oscl_DefAllocWithRefCounter< DefAlloc >::



Public Methods

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

Static Public Methods

- Oscl_DefAllocWithRefCounter * [New](#) ()

6.41.1 Detailed Description

template<class DefAlloc> class Oscl_DefAllocWithRefCounter< DefAlloc >

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

6.41.2 Member Function Documentation

6.41.2.1 template<class DefAlloc> void Oscl_DefAllocWithRefCounter< DefAlloc >::addRef () [inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

6.41.2.2 template<class DefAlloc> void Oscl_DefAllocWithRefCounter< DefAlloc >::Delete () [inline]

Delete object

6.41.2.3 template<class DefAlloc> uint32 Oscl_DefAllocWithRefCounter< DefAlloc >::getCount () [inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

**6.41.2.4 template<class DefAlloc> Oscl_DefAllocWithRefCounter*
Oscl_DefAllocWithRefCounter< DefAlloc >::New () [inline, static]**

Create object

**6.41.2.5 template<class DefAlloc> void Oscl_DefAllocWithRefCounter< DefAlloc >::removeRef
() [inline, virtual]**

Delete from reference count

Implements [OsclRefCounter](#).

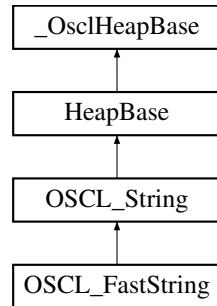
The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

6.42 OSCL_FastString Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_FastString::



Public Types

- `typedef OSCL_String::chartype chartype`

Public Methods

- `OSCL_IMPORT_REF OSCL_FastString()`
- `OSCL_IMPORT_REF OSCL_FastString(const OSCL_FastString &src)`
- `OSCL_IMPORT_REF OSCL_FastString(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_FastString(chartype *buf, uint32 maxlen)`
- `OSCL_IMPORT_REF ~OSCL_FastString()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_FastString & operator=(const OSCL_FastString &src)`
- `OSCL_IMPORT_REF OSCL_FastString & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(chartype *cstr, uint32 maxlen)`
- `OSCL_IMPORT_REF void set_length()`

Friends

- `class OSCL_String`

6.42.1 Detailed Description

`OSCL_FastString` is a simple string class, compatible with regular character array strings.

This class does not allocate internal memory for the string but acts as a container for a user-defined buffer. This means no copying of the string is done and provides a faster way of manipulating strings. Depending on initialization, this container provides either read-only or read-write access to the string.

Implementation assumes the input string is null-terminated.

Parameters:

C: type of character.

6.42.2 Member Typedef Documentation

6.42.2.1 `typedef OSCL_String::chartype OSCL_FastString::chartype`

Reimplemented from [OSCL_String](#).

6.42.3 Constructor & Destructor Documentation

6.42.3.1 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString()`

Default constructor.

6.42.3.2 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(const OSCL_FastString & src)`

Creates a fast string that contains a copy of the input string. The string inherits the writable-ness of the source string.

Parameters:

src: input string.

6.42.3.3 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(const chartype * cstr)`

Create the string and initialize it to contain the input string. The string is not writable.

am: null-terminated string.

6.42.3.4 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(chartype * buf, uint32 maxlen)`

Create the string and initialize it to contain the input string. The string is writable.

Parameters:

cp: null-terminated string.

maxlen: maximum size of storage at *cp*, not incl null terminator. If input string is not null-terminated, the function leaves.

6.42.3.5 `OSCL_IMPORT_REF OSCL_FastString::~OSCL_FastString()`

6.42.4 Member Function Documentation

6.42.4.1 `OSCL_IMPORT_REF const chartype* OSCL_FastString::get_cstr() [virtual]`

This function returns the C-style string for read access.

Implements [OSCL_String](#).

6.42.4.2 OSCL_IMPORT_REF uint32 OSCL_FastString::get_maxsize () [virtual]

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implements [OSCL_String](#).

6.42.4.3 OSCL_IMPORT_REF uint32 OSCL_FastString::get_size () [virtual]

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

6.42.4.4 OSCL_IMPORT_REF chartype* OSCL_FastString::get_str () [virtual]

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implements [OSCL_String](#).

6.42.4.5 OSCL_IMPORT_REF OSCL_FastString& OSCL_FastString::operator= (const chartype * cstr)

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

6.42.4.6 OSCL_IMPORT_REF OSCL_FastString& OSCL_FastString::operator= (const OSCL_FastString & src)

Assignment operators

6.42.4.7 OSCL_IMPORT_REF void OSCL_FastString::set (chartype * cstr, uint32 maxlen)

This function can be used to reassign the string to a new writable string. If input string is not null-terminated, the function leaves.

6.42.4.8 OSCL_IMPORT_REF void OSCL_FastString::set_length ()

This function can be used to refresh the string size in case the contents of the string buffer have been modified since the container was created.

6.42.5 Friends And Related Function Documentation**6.42.5.1 friend class OSCL_String [friend]**

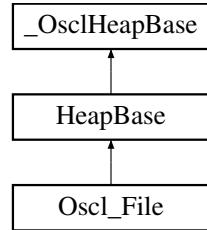
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

6.43 Oscl_File Class Reference

```
#include <oscl_file_io.h>
```

Inheritance diagram for Oscl_File::



Public Types

- enum `seek_type` { `SEEKSET`, `SEEKCUR`, `SEEKEND` }
- enum `mode_type` { `MODE_READ` = 0x0001, `MODE_READWRITE` = 0x0002, `MODE_APPEND` = 0x0004, `MODE_BINARY` = 0x0008, `MODE_TEXT` = 0x0010, `MODE_READ_PLUS` = 0x0020 }
- enum `TSymbianAccessMode` { `ESymbianAccessMode_Rfile` = 0, `ESymbianAccessMode_RfileBuf` = 1 }

Public Methods

- OSCL_IMPORT_REF `Oscl_File` ()
- OSCL_IMPORT_REF `Oscl_File` (uint32 aCacheSize)
- OSCL_IMPORT_REF `Oscl_File` (uint32 aCacheSize, `OsclFileHandle` *aFileHandle)
- OSCL_IMPORT_REF `~Oscl_File` ()
- OSCL_IMPORT_REF void `SetPVCacheSize` (uint32 aSize)
- 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` (int32 offset, `seek_type` origin)
- OSCL_IMPORT_REF int32 `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 int32 `Size` ()
- OSCL_IMPORT_REF void `SetLoggingEnable` (bool aEnable)
- OSCL_IMPORT_REF void `SetSummaryStatsLoggingEnable` (bool aEnable)

Friends

- class [OsclFileCache](#)
- class [asyncfilereadwrite_test](#)
- class [asyncfilereadcancel_test](#)

6.43.1 Member Enumeration Documentation

6.43.1.1 enum Oscl_File::mode_type

Enumeration values:

MODE_READ Opens a file for reading. The file must exist.

MODE_READWRITE Opens the file for reading and writing. If the file exists, its contents will be overwritten unless APPEND mode is specified. If the file does not exist, it will be created.

MODE_APPEND Specifies all write operations to occur at the end of the file. The file pointer can be moved with the Seek command, but will always be moved to the end of the file for write commands.

MODE_BINARY Opens the file in 'binary' mode. This is the default.

MODE_TEXT Opens the file in 'text' mode. The default mode is 'binary'.

MODE_READ_PLUS Open a file for reading and writing. The file must exist. The default mode is 'binary'.

6.43.1.2 enum Oscl_File::seek_type

Enumeration values:

SEEKSET Beginning of file

SEEKCUR Current position of file pointer

SEEKEND End of file

6.43.1.3 enum Oscl_File::TSymbianAccessMode

Defines mode options for SetNativeAccessMode on Symbian.

Enumeration values:

ESymbianAccessMode_Rfile

ESymbianAccessMode_RfileBuf

6.43.2 Constructor & Destructor Documentation

6.43.2.1 OSCL_IMPORT_REF Oscl_File::Oscl_File()

Constructor

6.43.2.2 OSCL_IMPORT_REF Oscl_File::Oscl_File (uint32 *aCacheSize*)

Deprecated Constructor, present for back-compatibility.

Parameters:

aCacheSize: sets native buffer size, and when pv cache is enabled, also sets pv cache size.

6.43.2.3 OSCL_IMPORT_REF Oscl_File::Oscl_File (uint32 *aCacheSize*, **OsclFileHandle** * *aFileHandle*)

Deprecated Constructor, present for back-compatibility.

Parameters:

aCacheSize: sets native buffer size, and when pv cache is enabled, also sets pv cache size.

aFileHandle: open file handle.

6.43.2.4 OSCL_IMPORT_REF Oscl_File::~Oscl_File ()

Destructor

6.43.3 Member Function Documentation

6.43.3.1 OSCL_IMPORT_REF int32 Oscl_File::Close ()

The File Close operation Closes the file after flushing any remaining data in the buffers.

Note: If the file object was opened with an external file handle, then Close will simply flush the file. The file will remain open.

Returns:

returns 0 if successful, and a non-zero value otherwise

6.43.3.2 OSCL_IMPORT_REF int32 Oscl_File::EndOfFile ()

The File EOF(end of file) operation returns a nonzero value after the first read operation that attempts to read past the end of the file

Returns:

6.43.3.3 OSCL_IMPORT_REF int32 Oscl_File::Flush ()

The File Flush operation On an output stream OSCL_FileFlush causes any buffered but unwritten data to be written to the file.

Returns:

returns 0 if successful, and a non-zero value otherwise

6.43.3.4 OSCL_IMPORT_REF int32 Oscl_File::GetError ()

The File Error operation If no error has occurred on stream, returns 0. Otherwise, it returns a nonzero value

Returns:**6.43.3.5 OsclFileHandle* Oscl_File::Handle () [inline]**

Retrieve the file handle.

Returns:

file handle

6.43.3.6 OSCL_IMPORT_REF int32 Oscl_File::Open (const oscl_wchar *filename, uint32 mode, Oscl_FileServer &fileserv)

Opens a file.

Note: when an external file handle is used, Open will attach to the file handle and initialize cacheing features, but will not do a native file open.

Parameters:

filename name of file to open (Unicode)

mode combination of open mode flags

fileserv fileserv to use

Returns:

returns 0 if successful and a non-zero value otherwise

6.43.3.7 OSCL_IMPORT_REF int32 Oscl_File::Open (const char *filename, uint32 mode, Oscl_FileServer &fileserv)

Opens a file.

Note: when an external file handle is used, Open will attach to the file handle and initialize cacheing features, but will not do a native file open.

Parameters:

filename name of file to open (Utf8)

mode combination of open mode flags

fileserv fileserv to use

Returns:

returns 0 if successful and a non-zero value otherwise

6.43.3.8 OSCL_IMPORT_REF uint32 Oscl_File::Read (*OsclAny * buffer, uint32 size, uint32 numelements*)

The File Read operation Reads from the file into the buffer a maximum of 'numelements' of size 'size'.

Parameters:

buffer pointer to buffer of type void
size element size in bytes
numelements max number of elements to read

Returns:

returns the number of full elements actually read, which may be less than count if an error occurs or if the end of the file is encountered before reaching count. Use the CheckEndOfFile or GetError function to distinguish a read error from an end-of-file condition.

6.43.3.9 OSCL_IMPORT_REF int32 Oscl_File::Seek (int32 *offset, seek_type origin*)

The File Seek operation Sets the position for file pointer

Parameters:

offset offset from the specified origin.
origin starting point

Returns:

returns 0 on success, and a non-zero value otherwise

6.43.3.10 OSCL_IMPORT_REF void Oscl_File::SetAsyncReadBufferSize (uint32 *aSize*)

SetAsyncReadBufferSize configures the asynchronous background read function. May not be available on all platforms.

This should be called before opening the file. If used when the file is open, the option will not take effect until the next Open.

Note: if asynchronous read is not available on the platform, this call will have no effect.

Parameters:

aSize: buffer size in bytes. Zero disables the feature.

6.43.3.11 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.

Parameters:

aHandle: container for an open file handle.

Returns:

returns 0 if successful, non-zero if error.

6.43.3.12 OSCL_IMPORT_REF void Oscl_File::SetLoggingEnable (bool *aEnable*)

SetLoggingEnable configures the [PVLogger](#) output for this file. This will enable full logging of each API entry and exit using the logger object "Oscl_File", plus full logging of native operation entry & exit using logger object "OsclNativeFile".

Parameters:

aEnable: true to enable, false to disable logging.

6.43.3.13 OSCL_IMPORT_REF void Oscl_File::SetNativeAccessMode (uint32 *aMode*)

SetNativeAccessMode allows switching between different native file access modes, when available.

Note: for Symbian, use the TSymbianAccessMode values to choose the mode. If multiple access modes are not available on the platform, this call will have no effect.

Parameters:

aMode: access mode.

6.43.3.14 OSCL_IMPORT_REF void Oscl_File::SetNativeBufferSize (int32 *aSize*)

SetNativeBufferSize configures the native file buffering feature, when available.

This should be called before opening the file. If used when the file is open, the option will not take effect until the next Open.

Note: For Symbian, this sets the RFileBuf cache size. If native buffering is not available on the platform, this call will have no effect.

Parameters:

aSize: native buffer size in bytes. Zero disables the feature.

6.43.3.15 OSCL_IMPORT_REF void Oscl_File::SetPVCacheSize (uint32 *aSize*)

SetPVCacheSize configures the read/write cache.

This should be called before opening the file. If used when the file is open, the option will not take effect until the next Open.

Parameters:

aSize: cache size in bytes. Zero disables the cache.

6.43.3.16 OSCL_IMPORT_REF void Oscl_File::SetSummaryStatsLoggingEnable (bool *aEnable*)

SetSummaryStatsLoggingEnable configures the [PVLogger](#) output for this file. This will enable summary statistics logging only, using the logger object "OsclFileStats".

Parameters:

aEnable: true to enable, false to disable stats logging.

6.43.3.17 OSCL_IMPORT_REF int32 Oscl_File::Size ()

Get the file size in bytes.

Returns:

- The size of the file, or -1 on error.

6.43.3.18 OSCL_IMPORT_REF int32 Oscl_File::Tell ()

The File Tell operation Returns the current file position for file specified by fp

6.43.3.19 OSCL_IMPORT_REF uint32 Oscl_File::Write (const OsclAny * *buffer*, uint32 *size*, uint32 *numelements*)

The File Write operation Writes from the buffer '*numelements*' objects of size '*size*'

Parameters:

- buffer* pointer to buffer of type void
- size* element size in bytes
- numelements* number of elements to write

Returns:

The number of elements written

6.43.4 Friends And Related Function Documentation

6.43.4.1 friend class asyncreadcancel_test [friend]

6.43.4.2 friend class asyncreadwrite_test [friend]

6.43.4.3 friend class OsclFileCache [friend]

The documentation for this class was generated from the following file:

- [oscl_file_io.h](#)

6.44 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_OTHER` }
- enum `element_type` { `FILE_TYPE` = 0, `DIR_TYPE`, `INVALID_TYPE` }

Public Methods

- OSCL_IMPORT_REF const char * `FindFirst` (const char *directory, const char *pattern, char *buf, uint32 buflen)
- OSCL_IMPORT_REF const `oscl_wchar` * `FindFirst` (const `oscl_wchar` *directory, const `oscl_wchar` *pattern, `oscl_wchar` *buf, uint32 buflen)
- OSCL_IMPORT_REF char * `FindNext` (char *buf, uint32 buflen)
- OSCL_IMPORT_REF `oscl_wchar` * `FindNext` (`oscl_wchar` *buf, uint32 buflen)
- OSCL_IMPORT_REF void `Close` ()
- OSCL_IMPORT_REF `element_type` `GetElementType` ()
- OSCL_IMPORT_REF `error_type` `GetLastError` ()
- OSCL_IMPORT_REF `Oscl_FileFind` ()
- OSCL_IMPORT_REF `~Oscl_FileFind` ()

6.44.1 Detailed Description

`Oscl_FileFind` class defines the generic way of finding filesystem elements that match a pattern within a directory

6.44.2 Member Enumeration Documentation

6.44.2.1 enum Oscl_FileFind::element_type

Enumeration values:

`FILE_TYPE`

`DIR_TYPE`

`INVALID_TYPE`

6.44.2.2 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_OTHER

6.44.3 Constructor & Destructor Documentation

6.44.3.1 OSCL_IMPORT_REF Oscl_FileFind::Oscl_FileFind ()

constructor.

Returns:

none

6.44.3.2 OSCL_IMPORT_REF Oscl_FileFind::~Oscl_FileFind ()

destructor. will deallocate open handles if necessary

Returns:

none

6.44.4 Member Function Documentation

6.44.4.1 OSCL_IMPORT_REF void Oscl_FileFind::Close ()

closes the handle to directory.

Returns:

none

6.44.4.2 OSCL_IMPORT_REF const oscl_wchar* Oscl_FileFind::FindFirst (const oscl_wchar * *directory*, const oscl_wchar * *pattern*, oscl_wchar * *buf*, uint32 *buflen*)

Opens a directory for reading.

Parameters:

directory directory to search (utf16).

pattern wildcard pattern filter (utf16). passing NULL, results in a universal match.

buf buffer for returned pathname (utf16).

buflen size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E_BUFFER_TOO_SMALL.

Returns:

returns a pointer to buffer supplied, which contains the pathname of the next found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

6.44.4.3 OSCL_IMPORT_REF const char* Oscl_FileFind::FindFirst (const char * *directory*, const char * *pattern*, char * *buf*, uint32 *buflen*)

Finds first element matching the pattern.

Parameters:

directory directory to search (utf8).

pattern wildcard pattern filter (utf8). passing NULL, results in a universal match.

buf buffer for returned pathname (utf8).

buflen size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E_BUFFER_TOO_SMALL.

Returns:

returns a pointer to buffer supplied, which contains the pathname of the next found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

6.44.4.4 OSCL_IMPORT_REF oscl_wchar* Oscl_FileFind::FindNext (oscl_wchar * *buf*, uint32 *buflen*)

Reads the next element in a directory. Note: the pointer returned by this function is not persistent and should be stored. Its scope is limited to the lifetime of the class.

Parameters:

buf buffer to hold directory name(utf16)

buflen size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E_BUFFER_TOO_SMALL.

Returns:

returns a pointer to buffer supplied, which contains the pathname of the next found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

6.44.4.5 OSCL_IMPORT_REF char* Oscl_FileFind::FindNext (char * *buf*, uint32 *buflen*)

Reads the next element in the directory. Note: the pointer returned by this function is not persistent and should be stored. Its scope is limited to the lifetime of the class.

Parameters:

buf buffer to hold directory name(utf8)

buflen size in wide characters of buf. If buf is not large enough to hold the returned string, NULL is returned, and GetLastError is set to E_BUFFER_TOO_SMALL.

Returns:

returns a pointer to buffer supplied, which contains the pathname of the next found element, or NULL otherwise. On a NULL return value, [GetLastError\(\)](#) returns a more detailed error.

6.44.4.6 OSCL_IMPORT_REF element_type Oscl_FileFind::GetElementType ()

Returns the element type for the last element returned

Returns:

see enumeration above for more info.

6.44.4.7 OSCL_IMPORT_REF [error_type](#) Oscl_FileFind::GetLastError ()

Returns the error code for the last operation.

Returns:

see enumeration above for more info.

The documentation for this class was generated from the following file:

- [oscl_file_find.h](#)

6.45 Oscl_FileServer Class Reference

```
#include <oscl_file_server.h>
```

Public Methods

- OSCL_IMPORT_REF [Oscl_FileServer \(\)](#)
- OSCL_IMPORT_REF [~Oscl_FileServer \(\)](#)
- OSCL_IMPORT_REF int32 [Connect \(\)](#)
- OSCL_IMPORT_REF int32 [Close \(\)](#)
- OSCL_IMPORT_REF int32 [Oscl_DeleteFile \(const char *filename\)](#)
- OSCL_IMPORT_REF int32 [Oscl_DeleteFile \(const oscl_wchar *filename\)](#)

Friends

- class [Oscl_File](#)
- class [OsclNativeFile](#)

6.45.1 Constructor & Destructor Documentation

6.45.1.1 OSCL_IMPORT_REF Oscl_FileServer::Oscl_FileServer ()

Constructor

6.45.1.2 OSCL_IMPORT_REF Oscl_FileServer::~Oscl_FileServer ()

Destructor

6.45.2 Member Function Documentation

6.45.2.1 OSCL_IMPORT_REF int32 Oscl_FileServer::Close ()

Closes a file server.

Returns:

returns 0 on success and a non-zero value otherwise

6.45.2.2 OSCL_IMPORT_REF int32 Oscl_FileServer::Connect ()

Connects the server. This must be called before a file server can be used.

Returns:

returns 0 on success and a non-zero value otherwise

**6.45.2.3 OSCL_IMPORT_REF int32 Oscl_FileServer::Oscl_DeleteFile (const oscl_wchar *
filename)**

Deletes a file from the filesystem

Parameters:

filename name of the file to delete (Unicode)

Returns:

returns 0 if successful, and a non-zero value otherwise.

6.45.2.4 OSCL_IMPORT_REF int32 Oscl_FileServer::Oscl_DeleteFile (const char **filename*)

Deletes a file from the filesystem *

Parameters:

filename name of the file to delete (Utf8)

Returns:

returns 0 if successful, and a non-zero value otherwise.

6.45.3 Friends And Related Function Documentation**6.45.3.1 friend class Oscl_File [friend]****6.45.3.2 friend class OsclNativeFile [friend]**

The documentation for this class was generated from the following file:

- [oscl_file_server.h](#)

6.46 oscl_fsstat Struct Reference

```
#include <oscl_file_dir_utils.h>
```

Data Fields

- [uint64 freebytes](#)
- [uint64 totalbytes](#)

6.46.1 Field Documentation

6.46.1.1 [uint64 oscl_fsstat::freebytes](#)

6.46.1.2 [uint64 oscl_fsstat::totalbytes](#)

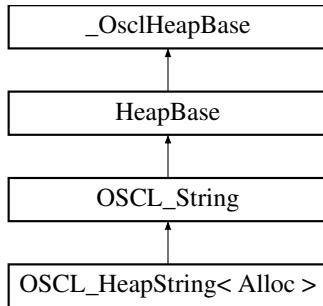
The documentation for this struct was generated from the following file:

- [oscl_file_dir_utils.h](#)

6.47 OSCL_HeapString< Alloc > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_HeapString< Alloc >::



Public Types

- `typedef OSCL_String::chartype chartype`

Public Methods

- `OSCL_HeapString()`
- `OSCL_HeapString(const OSCL_HeapString &src)`
- `OSCL_HeapString(const OSCL_String &src)`
- `OSCL_HeapString(const chartype *cstr)`
- `OSCL_HeapString(const chartype *buf, uint32 length)`
- `~OSCL_HeapString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype * get_cstr() const`
- `chartype * get_str() const`
- `OSCL_HeapString & operator=(const OSCL_HeapString &src)`
- `OSCL_HeapString & operator=(const OSCL_String &src)`
- `OSCL_HeapString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`

Friends

- class `OSCL_String`

6.47.1 Detailed Description

`template<class Alloc> class OSCL_HeapString< Alloc >`

`OSCL_HeapString` is a simple string class, compatible with regular character array strings.

The string array is variable length, is allocated from the heap, and is modifiable. A copy-on-write mechanism is used to minimize unnecessary copying when multiple instances of a string are created for reading.

Allocated memory is automatically freed by the class destructor when the last string referencing the memory is destroyed.

The class HAS NO thread synchronization built-in, so it is NOT MT-SAFE. External locks should be used if the class is to be shared across threads.

Parameters:

Alloc: memory allocator, derived from [Oscl_DefAlloc](#).

6.47.2 Member Typedef Documentation

6.47.2.1 template<class Alloc> typedef OSCL_String::chartype OSCL_HeapString< Alloc >::chartype

Reimplemented from [OSCL_String](#).

6.47.3 Friends And Related Function Documentation

6.47.3.1 template<class Alloc> friend class OSCL_String [friend]

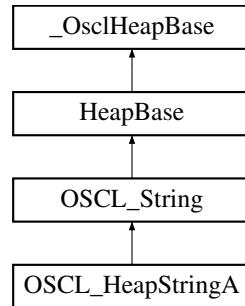
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

6.48 OSCL_HeapStringA Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_HeapStringA::



Public Types

- `typedef OSCL_String::chartype chartype`

Public Methods

- `OSCL_IMPORT_REF OSCL_HeapStringA ()`
- `OSCL_IMPORT_REF OSCL_HeapStringA (Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const OSCL_HeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const OSCL_HeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const OSCL_String &src, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const chartype *cstr, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const chartype *buf, uint32 length, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF ~OSCL_HeapStringA ()`
- `OSCL_IMPORT_REF uint32 get_size () const`
- `OSCL_IMPORT_REF uint32 get_maxsize () const`
- `OSCL_IMPORT_REF const chartype * get_cstr () const`
- `OSCL_IMPORT_REF chartype * get_str () const`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator= (const OSCL_HeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator= (const OSCL_String &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator= (const chartype *cstr)`
- `OSCL_IMPORT_REF void set (const chartype *buf, uint32 length)`

Friends

- `class OSCL_String`

6.48.1 Detailed Description

OSCL_HeapStringA is a simple string class, compatible with regular character array strings. It is similar to [OSCL_HeapString](#), except that the allocator is passed at run-time instead of compile-time. The allocator pointer is passed in the constructor, and may be a reference-counted object. If the allocator is not a reference-counted object then it must persist over the lifetime of all OSCL_HeapStringA objects that use it. If no allocator is provided, then an [OsclMemAllocator](#) will be used.

The string array is variable length, is allocated from the heap, and is modifiable. A copy-on-write mechanism is used to minimize unnecessary copying when multiple instances of a string are created for reading. Allocated memory is automatically freed by the class destructor when the last string referencing the memory is destroyed.

The class HAS NO thread synchronization built-in, so it is NOT MT-SAFE. External locks should be used if the class is to be shared across threads.

6.48.2 Member Typedef Documentation

6.48.2.1 `typedef OSCL_String::chartype OSCL_HeapStringA::chartype`

Reimplemented from [OSCL_String](#).

6.48.3 Constructor & Destructor Documentation

6.48.3.1 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA()`

The default constructor creates an empty string.

am: (optional) allocator or reference-counted allocator.

am: (optional) reference counter associated with allocator object.

If no allocator is provided, this object will use an [OsclMemAllocator](#).

6.48.3.2 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (Oscl_DefAlloc * alloc, OsclRefCounter * ref = NULL)`

6.48.3.3 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const OSCL_HeapStringA & src)`

Creates a heap string that contains a copy of the input string.

Parameters:

src: input string.

am: (optional) allocator or reference-counted allocator.

am: (optional) reference counter associated with allocator object.

If no allocator is provided, this object will use an [OsclMemAllocator](#).

- 6.48.3.4 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const OSCL_HeapStringA & src, Oscl_DefAlloc * alloc, OsclRefCounter * ref = NULL)**
- 6.48.3.5 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const OSCL_String & src, Oscl_DefAlloc * alloc = NULL, OsclRefCounter * ref = NULL)**
- 6.48.3.6 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const chartype * cstr, Oscl_DefAlloc * alloc = NULL, OsclRefCounter * ref = NULL)**

Creates a heap string that contains a copy of the input string.

Parameters:

cp: null-terminated string.

am: (optional) allocator or reference-counted allocator.

am: (optional) reference counter associated with allocator object.

If no allocator is provided, this object will use an [OsclMemAllocator](#).

- 6.48.3.7 OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA (const chartype * buf, uint32 length, Oscl_DefAlloc * alloc = NULL, OsclRefCounter * ref = NULL)**

Creates a heap string that contains a copy of the input string or character array.

Parameters:

src: character array, not necessarily null-terminated.

length: number of characters to copy.

am: (optional) allocator or reference-counted allocator.

am: (optional) reference counter associated with allocator object.

If no allocator is provided, this object will use an [OsclMemAllocator](#).

- 6.48.3.8 OSCL_IMPORT_REF OSCL_HeapStringA::~OSCL_HeapStringA ()**

6.48.4 Member Function Documentation

- 6.48.4.1 OSCL_IMPORT_REF const chartype* OSCL_HeapStringA::get_cstr () [virtual]**

This function returns the C-style string for read access.

Implements [OSCL_String](#).

- 6.48.4.2 OSCL_IMPORT_REF uint32 OSCL_HeapStringA::get_maxsize () [virtual]**

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implements [OSCL_String](#).

6.48.4.3 OSCL_IMPORT_REF uint32 OSCL_HeapStringA::get_size () [virtual]

Pure virtuals from [OSCL_String](#)

Implements [OSCL_String](#).

6.48.4.4 OSCL_IMPORT_REF chartype* OSCL_HeapStringA::get_str () [virtual]

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implements [OSCL_String](#).

6.48.4.5 OSCL_IMPORT_REF OSCL_HeapStringA& OSCL_HeapStringA::operator= (const chartype * cstr)

Assignment operator

am: null-terminated string

Reimplemented from [OSCL_String](#).

6.48.4.6 OSCL_IMPORT_REF OSCL_HeapStringA& OSCL_HeapStringA::operator= (const OSCL_String & src)

Assignment operator

Reimplemented from [OSCL_String](#).

6.48.4.7 OSCL_IMPORT_REF OSCL_HeapStringA& OSCL_HeapStringA::operator= (const OSCL_HeapStringA & src)

Assignment operators

6.48.4.8 OSCL_IMPORT_REF void OSCL_HeapStringA::set (const chartype * buf, uint32 length)

Set the contents of this string to a new string or character array.

Parameters:

buf: string or character array.

length: number of characters to copy.

6.48.5 Friends And Related Function Documentation

6.48.5.1 friend class OSCL_String [friend]

The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

6.49 Oscl_Int64_Utils Class Reference

The Oscl_Int64_Utils class provides a wrapper for commonly used int64/uint64 operations.

```
#include <oscl_int64_utils.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [set_int64](#) (int64 &input_value, const int32 upper, const int32 lower)
- OSCL_IMPORT_REF int32 [get_int64_upper32](#) (const int64 &input_value)
- OSCL_IMPORT_REF int32 [get_int64_lower32](#) (const int64 &input_value)
- OSCL_IMPORT_REF int32 [get_int64_middle32](#) (const int64 &input_value)
- OSCL_IMPORT_REF void [set_uint64](#) (uint64 &input_value, const uint32 upper, const uint32 lower)
- OSCL_IMPORT_REF uint32 [get_uint64_upper32](#) (const uint64 &input_value)
- OSCL_IMPORT_REF uint32 [get_uint64_lower32](#) (const uint64 &input_value)
- OSCL_IMPORT_REF uint32 [get_uint64_middle32](#) (const uint64 &input_value)

6.49.1 Detailed Description

The Oscl_Int64_Utils class provides a wrapper for commonly used int64/uint64 operations.

The Oscl_Int64_Utils class:

Provides a wrapper for commonly used operations to mask the differences between OSes that have an int64/uint64 class instead of a 64-bit integer.

6.49.2 Member Function Documentation

- 6.49.2.1 **OSCL_IMPORT_REF int32 Oscl_Int64_Utils::get_int64_lower32 (const int64 & *input_value*) [static]**
- 6.49.2.2 **OSCL_IMPORT_REF int32 Oscl_Int64_Utils::get_int64_middle32 (const int64 & *input_value*) [static]**
- 6.49.2.3 **OSCL_IMPORT_REF int32 Oscl_Int64_Utils::get_int64_upper32 (const int64 & *input_value*) [static]**
- 6.49.2.4 **OSCL_IMPORT_REF uint32 Oscl_Int64_Utils::get_uint64_lower32 (const uint64 & *input_value*) [static]**
- 6.49.2.5 **OSCL_IMPORT_REF uint32 Oscl_Int64_Utils::get_uint64_middle32 (const uint64 & *input_value*) [static]**
- 6.49.2.6 **OSCL_IMPORT_REF uint32 Oscl_Int64_Utils::get_uint64_upper32 (const uint64 & *input_value*) [static]**
- 6.49.2.7 **OSCL_IMPORT_REF void Oscl_Int64_Utils::set_int64 (int64 & *input_value*, const int32 *upper*, const int32 *lower*) [static]**
- 6.49.2.8 **OSCL_IMPORT_REF void Oscl_Int64_Utils::set_uint64 (uint64 & *input_value*, const uint32 *upper*, const uint32 *lower*) [static]**

The documentation for this class was generated from the following file:

- [oscl_int64_utils.h](#)

6.50 Oscl_Less< T > Struct Template Reference

```
#include <oscl_map.h>
```

Public Methods

- bool [operator\(\)](#) (const T &x, const T &y) const

```
template<class T> struct Oscl_Less< T >
```

6.50.1 Member Function Documentation

6.50.1.1 template<class T> bool Oscl_Less< T >::operator() (const T & x, const T & y) const [inline]

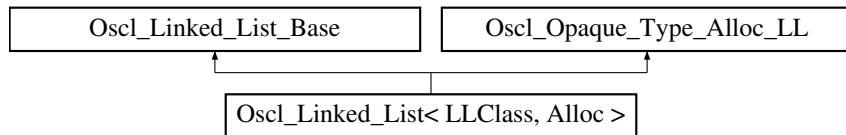
The documentation for this struct was generated from the following file:

- [oscl_map.h](#)

6.51 Oscl_Linked_List< LLClass, Alloc > Class Template Reference

```
#include <oscl_linked_list.h>
```

Inheritance diagram for Oscl_Linked_List< LLClass, Alloc >::



Public Methods

- [Oscl_Linked_List \(\)](#)
- [~Oscl_Linked_List \(\)](#)
- int32 [dequeue_element \(LLClass &element\)](#)
- int32 [get_first \(LLClass &ele\)](#)
- int32 [get_next \(LLClass &ele\)](#)
- int32 [check_list \(\)](#)
- int32 [get_num_elements \(\)](#)
- int32 [add_element \(LLClass &new_element\)](#)
- int32 [add_to_front \(const LLClass &new_element\)](#)
- int32 [get_element \(int32 index, LLClass &element\)](#)
- int32 [remove_element \(const LLClass &data_to_remove\)](#)
- int32 [get_index \(const LLClass &data\)](#)
- int32 [remove_element \(const int32 index_to_remove\)](#)
- int32 [move_to_end \(const LLClass &data_to_move\)](#)
- int32 [move_to_front \(const LLClass &data_to_move\)](#)

6.51.1 Detailed Description

`template<class LLClass, class Alloc> class Oscl_Linked_List< LLClass, Alloc >`

Oscl Linked List Class

6.51.2 Constructor & Destructor Documentation

6.51.2.1 `template<class LLClass, class Alloc> Oscl_Linked_List< LLClass, Alloc >::Oscl_Linked_List () [inline]`

Initialized the protected variables of list.

6.51.2.2 `template<class LLClass, class Alloc> Oscl_Linked_List< LLClass, Alloc >::~Oscl_Linked_List () [inline]`

The destructor.

6.51.3 Member Function Documentation

6.51.3.1 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::add_element (LLClass & *new_element*) [inline]

Adds new element to the list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

6.51.3.2 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::add_to_front (const LLClass & *new_element*) [inline]

Adds new element at the start of the list.if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

6.51.3.3 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::check_list () [inline]

Debug routine: Checks the list for elements.

Returns:

32-bit integer, if node count is equal to number of node added to the list then returns 1 otherwise returns 0.

Reimplemented from [Oscl_Linked_List_Base](#).

6.51.3.4 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::dequeue_element (LLClass & *element*) [inline]

6.51.3.5 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::get_element (int32 *index*, LLClass & *element*) [inline]

Search and returns the element in the list for passed index.

Parameters:

index, element The index is the count for the node.

Returns:

32-bit integer on success returns 1 otherwise returns 0.

6.51.3.6 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::get_first (LLClass & *ele*) [inline]

Return the first element of list in passed parameter,

Parameters:

ele return the value of first element of list in this parameter

Returns:

32-bit interger,If first element found, it returns 1 otherwise it returns 0

6.51.3.7 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::get_index (const LLClass & *data*) [inline]

Returns the index for requested element.

Parameters:

data the element for which index to be return.

Returns:

32-bit integer if data is found in the list it returns index otherwise it returns -1.

6.51.3.8 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::get_next (LLClass & *ele*) [inline]

Return the next element of list in passed parameter,

Parameters:

ele return the value of next element of list in this parameter

Returns:

32-bit interger ,if next element is found in list,it returns 1 otherwise it returns 0

6.51.3.9 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::get_num_elements () [inline]

Get number of elements in the list.

Returns:

32-bit integer, number of elements in list.

6.51.3.10 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::move_to_end (const LLClass & *data_to_move*) [inline]

Moves the element to end of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

6.51.3.11 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::move_to_front (const LLClass & *data_to_move*) [inline]

Moves the element to front of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

6.51.3.12 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::remove_element (const int32 *index_to_remove*) [inline]

Removes the element for requested index.

Parameters:

index_to_remove

Returns:

on success return 1 otherwise return 0.

Reimplemented from [Oscl_Linked_List_Base](#).

6.51.3.13 template<class LLClass, class Alloc> int32 Oscl_Linked_List< LLClass, Alloc >::remove_element (const LLClass & *data_to_remove*) [inline]

Removes the element from the list.

Parameters:

data_to_remove

Returns:

32-bit integer on if element fount in the list returns 1 otherwise returns 0.

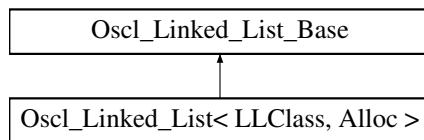
The documentation for this class was generated from the following file:

- [oscl_linked_list.h](#)

6.52 Oscl_Linked_List_Base Class Reference

```
#include <oscl_linked_list.h>
```

Inheritance diagram for Oscl_Linked_List_Base::



Protected Methods

- virtual ~[Oscl_Linked_List_Base](#) ()
- OSCL_IMPORT_REF void [construct](#) ([Oscl_Opaque_Type_Alloc_LL](#) *op)
- OSCL_IMPORT_REF void [destroy](#) ()
- OSCL_IMPORT_REF int32 [get_first](#) ([OsclAny](#) *ele)
- OSCL_IMPORT_REF int32 [get_next](#) ([OsclAny](#) *ele)
- OSCL_IMPORT_REF int32 [check_list](#) ()
- OSCL_IMPORT_REF int32 [add_element](#) ([OsclAny](#) *new_element)
- OSCL_IMPORT_REF int32 [add_to_front](#) (const [OsclAny](#) *new_element)
- OSCL_IMPORT_REF int32 [get_element](#) (int32 index, [OsclAny](#) *element)
- OSCL_IMPORT_REF int32 [remove_element](#) (const [OsclAny](#) *data_to_remove)
- OSCL_IMPORT_REF int32 [get_index](#) (const [OsclAny](#) *data)
- OSCL_IMPORT_REF int32 [remove_element](#) (const int32 index_to_remove)
- OSCL_IMPORT_REF int32 [move_to_end](#) (const [OsclAny](#) *data_to_move)
- OSCL_IMPORT_REF int32 [move_to_front](#) (const [OsclAny](#) *data_to_move)

Protected Attributes

- [OsclAny](#) * head
- [OsclAny](#) * tail
- [OsclAny](#) * iterator
- int32 [num_elements](#)
- uint32 [sizeof_T](#)

6.52.1 Detailed Description

Oscl Linked List Base Class. A non-templated base class is used to avoid large inline functions in the [Oscl_Linked_List](#) implementation.

6.52.2 Constructor & Destructor Documentation

6.52.2.1 `virtual Oscl_Linked_List_Base::~Oscl_Linked_List_Base ()` [inline, protected, virtual]

6.52.3 Member Function Documentation

6.52.3.1 `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::add_element (OsclAny * new_element)` [protected]

Adds new element to the list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

6.52.3.2 `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::add_to_front (const OsclAny * new_element)` [protected]

Adds new element at the start of the list.if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

6.52.3.3 `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::check_list ()` [protected]

Debug routine: Checks the list for elements.

Returns:

32-bit integer, if node count is equal to number of node added to the list then returns 1 otherwise returns 0.

Reimplemented in [Oscl_Linked_List< LLClass, Alloc >](#).

6.52.3.4 `OSCL_IMPORT_REF void Oscl_Linked_List_Base::construct (Oscl_Opaque_Type_Alloc_LL * op)` [protected]

6.52.3.5 `OSCL_IMPORT_REF void Oscl_Linked_List_Base::destroy ()` [protected]

6.52.3.6 `OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::get_element (int32 index, OsclAny * element)` [protected]

Search and returns the element in the list for passed index.

Parameters:

index, element The index is the count for the node.

Returns:

32-bit integer on success returns 1 otherwise returns 0.

**6.52.3.7 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::get_first (OsclAny * *ele*)
[protected]**

Return the first element of list in passed parameter,

Parameters:

ele return the value of first element of list in this parameter

Returns:

32-bit interger,If first element found, it returns 1 otherwise it returns 0

**6.52.3.8 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::get_index (const OsclAny * *data*)
[protected]**

Returns the index for requested element.

Parameters:

data the element for which index to be return.

Returns:

32-bit integer if data is found in the list it returns index otherwise it returns -1.

**6.52.3.9 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::get_next (OsclAny * *ele*)
[protected]**

Return the next element of list in passed parameter,

Parameters:

ele return the value of next element of list in this parameter

Returns:

32-bit interger ,if next element is found in list,it returns 1 otherwise it returns 0

**6.52.3.10 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::move_to_end (const OsclAny *
data_to_move) [protected]**

Moves the element to end of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

**6.52.3.11 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::move_to_front (const OsclAny *
data_to_move) [protected]**

Moves the element to front of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

**6.52.3.12 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::remove_element (const int32
index_to_remove) [protected]**

Removes the element for requested index.

Parameters:

index_to_remove

Returns:

on success return 1 otherwise return 0.

Reimplemented in [Oscl_Linked_List< LLClass, Alloc >](#).

**6.52.3.13 OSCL_IMPORT_REF int32 Oscl_Linked_List_Base::remove_element (const OsclAny *
data_to_remove) [protected]**

Removes the element from the list.

Parameters:

data_to_remove

Returns:

32-bit integer on if element fount in the list returns 1 otherwise returns 0.

6.52.4 Field Documentation

6.52.4.1 OsclAny* Oscl_Linked_List_Base::head [protected]**6.52.4.2 OsclAny* Oscl_Linked_List_Base::iterator [protected]****6.52.4.3 int32 Oscl_Linked_List_Base::num_elements [protected]****6.52.4.4 uint32 Oscl_Linked_List_Base::sizeof_T [protected]****6.52.4.5 OsclAny* Oscl_Linked_List_Base::tail [protected]**

The documentation for this class was generated from the following file:

- [oscl_linked_list.h](#)

6.53 Oscl_Map< Key, T, Alloc, Compare > Class Template Reference

```
#include <oscl_map.h>
```

Public Types

- `typedef Key key_type`
- `typedef Compare key_compare`
- `typedef Oscl_Pair< const Key, T > value_type`
- `typedef Oscl_Map< Key, T, Alloc, Compare > self`
- `typedef rep_type::pointer pointer`
- `typedef rep_type::reference reference`
- `typedef rep_type::const_reference const_reference`
- `typedef rep_type::iterator iterator`
- `typedef rep_type::const_iterator const_iterator`
- `typedef rep_type::size_type size_type`
- `typedef Oscl_Pair< iterator, bool > pair_iterator_bool`
- `typedef Oscl_Pair< iterator, iterator > pair_iterator_iterator`
- `typedef Oscl_Pair< const_iterator, const_iterator > pair_citerator_citerator`

Public Methods

- `Oscl_Map (const Compare &comp=Compare())`
- `Oscl_Map (const self &x)`
- `self & operator= (const self &x)`
- `key_compare key_comp () const`
- `value_compare value_comp () const`
- `iterator begin ()`
- `const_iterator begin () const`
- `iterator end ()`
- `const_iterator end () const`
- `bool empty () const`
- `size_type size () const`
- `size_type max_size () const`
- `T & operator[] (const key_type &k)`
- `pair_iterator_bool insert (const value_type &x)`
- `iterator insert (iterator position, const value_type &x)`
- `void insert (const value_type *first, const value_type *last)`
- `void erase (iterator position)`
- `size_type erase (const key_type &x)`
- `void erase (iterator first, iterator last)`
- `void clear ()`
- `iterator find (const key_type &x)`
- `const_iterator find (const key_type &x) const`
- `size_type count (const key_type &x) const`
- `iterator lower_bound (const key_type &x)`
- `const_iterator lower_bound (const key_type &x) const`
- `iterator upper_bound (const key_type &x)`

- `const_iterator upper_bound (const key_type &x) const`
- `pair_iterator iterator equal_range (const key_type &x)`
- `pair_citerator citerator equal_range (const key_type &x) const`

6.53.1 Detailed Description

template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> class Oscl_Map< Key, T, Alloc, Compare >

Oscl_Map Class. A subset of STL::Map methods. Oscl_Map is a sorted associative container that associates objects of type Key with objects of type T. It is also a unique associative container, meaning that no two elements have the same key. Oscl_Map uses the key to speed lookup, insertion, and deletion of elements. It is often superior to all other containers when you need to lookup an element by key value. Due to the underlying tree structure, inserts and erases can be performed in logarithmic time, where a vector would take linear time in some cases.

6.53.2 Member Typedef Documentation

- 6.53.2.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::const_iterator Oscl_Map< Key, T, Alloc, Compare >::const_iterator
- 6.53.2.2 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::const_reference Oscl_Map< Key, T, Alloc, Compare >::const_reference
- 6.53.2.3 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::iterator Oscl_Map< Key, T, Alloc, Compare >::iterator
- 6.53.2.4 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Compare Oscl_Map< Key, T, Alloc, Compare >::key_compare
- 6.53.2.5 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Key Oscl_Map< Key, T, Alloc, Compare >::key_type
- 6.53.2.6 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Oscl_Pair<const_iterator, const_iterator> Oscl_Map< Key, T, Alloc, Compare >::pair_citerator_citerator
- 6.53.2.7 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Oscl_Pair<iterator, bool> Oscl_Map< Key, T, Alloc, Compare >::pair_iterator_bool
- 6.53.2.8 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Oscl_Pair<iterator, iterator> Oscl_Map< Key, T, Alloc, Compare >::pair_iterator_iterator
- 6.53.2.9 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::pointer Oscl_Map< Key, T, Alloc, Compare >::pointer
- 6.53.2.10 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::reference Oscl_Map< Key, T, Alloc, Compare >::reference
- 6.53.2.11 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Oscl_Map<Key, T, Alloc, Compare> Oscl_Map< Key, T, Alloc, Compare >::self
- 6.53.2.12 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::size_type Oscl_Map< Key, T, Alloc, Compare >::size_type
- 6.53.2.13 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Oscl_Pair<const Key, T> Oscl_Map< Key, T, Alloc, Compare >::value_type

6.53.3 Constructor & Destructor Documentation

- 6.53.3.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Oscl_Map< Key, T, Alloc, Compare >::Oscl_Map (const Compare & comp = Compare()) [inline]

Creates an empty map using comp as the key compare object

6.53.3.2 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Oscl_Map< Key, T, Alloc, Compare >::Oscl_Map (const **self & *x*) [inline]**

Oscl_Map copy constructor

6.53.4 Member Function Documentation

6.53.4.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> const_iterator Oscl_Map< Key, T, Alloc, Compare >::begin () const [inline]

Returns a const iterator pointing to the beginning of the map

6.53.4.2 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator Oscl_Map< Key, T, Alloc, Compare >::begin () [inline]

Returns an iterator pointing to the beginning of the map

6.53.4.3 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> void Oscl_Map< Key, T, Alloc, Compare >::clear () [inline]

Erases all elements

6.53.4.4 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> size_type Oscl_Map< Key, T, Alloc, Compare >::count (const **key_type & *x*) const [inline]**

Returns the number of elements with key *x*. For map this will either be 0 or 1.

6.53.4.5 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> bool Oscl_Map< Key, T, Alloc, Compare >::empty () const [inline]

Returns true if map size is 0

6.53.4.6 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> const_iterator Oscl_Map< Key, T, Alloc, Compare >::end () const [inline]

Returns a const iterator pointing to the end of the map.

6.53.4.7 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator Oscl_Map< Key, T, Alloc, Compare >::end () [inline]

Returns an iterator pointing to the end of the map.

6.53.4.8 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> pair_citerator_citerator Oscl_Map< Key, T, Alloc, Compare >::equal_range (const **key_type & *x*) const [inline]**

Finds a range containing all elements whose key is *x*

**6.53.4.9 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
 pair_iterator_iterator Oscl_Map< Key, T, Alloc, Compare >::equal_range (const
 key_type & x) [inline]**

Finds a range containing all elements whose key is x

**6.53.4.10 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> void
 Oscl_Map< Key, T, Alloc, Compare >::erase (iterator first, iterator last) [inline]**

Erases all elements in the range [first,last)

**6.53.4.11 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> size_type
 Oscl_Map< Key, T, Alloc, Compare >::erase (const key_type & x) [inline]**

Erases the element with key x

**6.53.4.12 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> void
 Oscl_Map< Key, T, Alloc, Compare >::erase (iterator position) [inline]**

Erases the element pointed to by position

**6.53.4.13 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
 const_iterator Oscl_Map< Key, T, Alloc, Compare >::find (const key_type & x) const
 [inline]**

Finds an element whose key is x

**6.53.4.14 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator
 Oscl_Map< Key, T, Alloc, Compare >::find (const key_type & x) [inline]**

Finds an element whose key is x

**6.53.4.15 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> void
 Oscl_Map< Key, T, Alloc, Compare >::insert (const value_type *first, const value_type
 *last) [inline]**

Inserts the range [first,last) into the map

**6.53.4.16 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator
 Oscl_Map< Key, T, Alloc, Compare >::insert (iterator position, const value_type & x)
 [inline]**

Inserts x into the map using position as a hint as to where it should be inserted

**6.53.4.17 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
`pair_iterator_bool` Oscl_Map< Key, T, Alloc, Compare >::insert (const `value_type` & x) [inline]**

Inserts x into the map

**6.53.4.18 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
`key_compare` Oscl_Map< Key, T, Alloc, Compare >::key_comp () const [inline]**

Returns the key compare object used by the map

**6.53.4.19 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
`const_iterator` Oscl_Map< Key, T, Alloc, Compare >::lower_bound (const `key_type` & x) const [inline]**

Finds the first element whose key is not less than x

**6.53.4.20 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator
`Oscl_Map`< Key, T, Alloc, Compare >::lower_bound (const `key_type` & x) [inline]**

Finds the first element whose key is not less than x

**6.53.4.21 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> size_type
`Oscl_Map`< Key, T, Alloc, Compare >::max_size () const [inline]**

Returns the maximum possible size of the map

**6.53.4.22 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> self&
`Oscl_Map`< Key, T, Alloc, Compare >::operator= (const `self` & x) [inline]**

Oscl_Map assignment operator

6.53.4.23]

template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> T& Oscl_Map< Key, T, Alloc, Compare >::operator[] (const `key_type` & k) [inline]

Returns a reference to the object that is associated with a particular key. If the map does not already contain such an object, operator[] inserts the default object `value_type()`.

**6.53.4.24 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> size_type
`Oscl_Map`< Key, T, Alloc, Compare >::size () const [inline]**

Returns the size of the map

**6.53.4.25 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
const_iterator Oscl_Map< Key, T, Alloc, Compare >::upper_bound (const key_type &
x) const [inline]**

Finds the first element whose key is not greater than x

**6.53.4.26 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> iterator
Oscl_Map< Key, T, Alloc, Compare >::upper_bound (const key_type & x) [inline]**

Finds the first element whose key is not greater than x

**6.53.4.27 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>>
value_compare Oscl_Map< Key, T, Alloc, Compare >::value_comp () const
[inline]**

Returns the value compare object used by the map

The documentation for this class was generated from the following file:

- [oscl_map.h](#)

6.54 Oscl_Map< Key, T, Alloc, Compare >::value_compare Class Reference

```
#include <oscl_map.h>
```

Public Methods

- bool [operator\(\)](#) (const [value_type](#) &x, const [value_type](#) &y) const

Protected Methods

- [value_compare](#) (Compare c)

Protected Attributes

- Compare [comp](#)

Friends

- class [Oscl_Map< Key, T, Alloc, Compare >](#)

```
template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> class Oscl_Map< Key,  
T, Alloc, Compare >::value_compare
```

6.54.1 Constructor & Destructor Documentation

```
6.54.1.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::value\_compare (Compare c) [inline, protected]
```

6.54.2 Member Function Documentation

```
6.54.2.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> bool  
Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::operator\(\) (const value\_type &  
x, const value\_type &y) const [inline]
```

6.54.3 Friends And Related Function Documentation

```
6.54.3.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> friend class  
Oscl\_Map< Key, T, Alloc, Compare > [friend]
```

6.54.4 Field Documentation

```
6.54.4.1 template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Compare  
Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::comp [protected]
```

The documentation for this class was generated from the following file:

- [oscl_map.h](#)

6.55 Oscl_MTLinked_List< LLClass, Alloc, TheLock > Class Template Reference

```
#include <oscl_linked_list.h>
```

Public Methods

- [Oscl_MTLinked_List \(\)](#)
- [~Oscl_MTLinked_List \(\)](#)
- int32 [dequeue_element \(LLClass &element\)](#)
- int32 [add_element \(LLClass &new_element\)](#)
- int32 [add_to_front \(LLClass &new_element\)](#)
- uint32 [get_element \(int32 index, LLClass &element\)](#)
- int32 [remove_element \(const LLClass &data_to_remove\)](#)
- int32 [get_index \(const LLClass &data\)](#)
- int32 [remove_element \(const int32 index_to_remove\)](#)
- int32 [move_to_end \(const LLClass &data_to_move\)](#)
- int32 [move_to_front \(const LLClass &data_to_move\)](#)

Protected Attributes

- [Oscl_Linked_List< LLClass, Alloc > the_list](#)

6.55.1 Detailed Description

template<class LLClass, class Alloc, class TheLock> class Oscl_MTLinked_List< LLClass, Alloc, TheLock >

Oscl_MTLinked_List is a multi-threaded version of the LinkedList. It has mutex protection to allow access by different threads.

6.55.2 Constructor & Destructor Documentation

6.55.2.1 template<class LLClass, class Alloc, class TheLock> Oscl_MTLinked_List< LLClass, Alloc, TheLock >::Oscl_MTLinked_List () [inline]

Constructor for Oscl_MTLinked_List

6.55.2.2 template<class LLClass, class Alloc, class TheLock> Oscl_MTLinked_List< LLClass, Alloc, TheLock >::~Oscl_MTLinked_List () [inline]

Destructor for Oscl_MTLinked_List

6.55.3 Member Function Documentation

6.55.3.1 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::add_element (LLClass & *new_element*) [inline]

Adds new element to the Multi Threaded Linked list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

6.55.3.2 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::add_to_front (LLClass & *new_element*) [inline]

Adds new element at the start of the Multi Threaded Linked list. if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

Parameters:

new_element the element to be add in the list.

Returns:

32-bit integer on the success returns 1.

6.55.3.3 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::dequeue_element (LLClass & *element*) [inline]

6.55.3.4 template<class LLClass, class Alloc, class TheLock> uint32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::get_element (int32 *index*, LLClass & *element*) [inline]

Search and returns the element in the Multi Threaded Linked List for passed index.

Parameters:

index, element The index is the count for the node.

Returns:

32-bit integer on success returns 1 otherwise returns 0.

6.55.3.5 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::get_index (const LLClass & *data*) [inline]

Returns the index for requested element.

Parameters:

data the element for which index to be return.

Returns:

32-bit integer if data is found in the list it returns index otherwise it returns -1.

6.55.3.6 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::move_to_end (const LLClass & *data_to_move*) [inline]

Moves the element to end of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

6.55.3.7 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::move_to_front (const LLClass & *data_to_move*) [inline]

Moves the element to front of the list

Parameters:

data_to_move

Returns:

On success returns 1 otherwise returns 0.

6.55.3.8 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::remove_element (const int32 *index_to_remove*) [inline]

Removes the element for requested index.

Parameters:

index_to_remove

Returns:

on success return 1 otherwise return 0.

6.55.3.9 template<class LLClass, class Alloc, class TheLock> int32 Oscl_MTLinked_List< LLClass, Alloc, TheLock >::remove_element (const LLClass & *data_to_remove*) [inline]

Removes the element from the list.

Parameters:

data_to_remove

Returns:

32-bit integer on if element fount in the list returns 1 otherwise returns 0.

6.55.4 Field Documentation

6.55.4.1 template<class LLClass, class Alloc, class TheLock> Oscl_Linked_List<LLClass, Alloc> Oscl_MTLinked_List< LLClass, Alloc, TheLock >::the_list [protected]

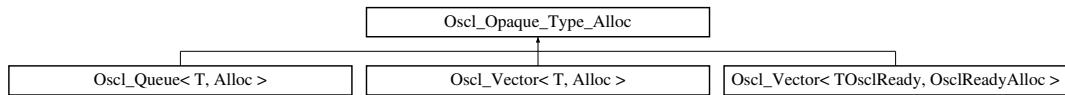
The documentation for this class was generated from the following file:

- [oscl_linked_list.h](#)

6.56 Oscl_Opaque_Type_Alloc Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl_Opaque_Type_Alloc::



Public Methods

- virtual void **construct** (**OsclAny** *p, const **OsclAny** *init_val)=0
- virtual void **destroy** (**OsclAny** *p)=0
- virtual **OsclAny** * **allocate** (const uint32 size)=0
- virtual void **deallocate** (**OsclAny** *p)=0

6.56.1 Detailed Description

This class combines opaque type operations with memory allocation operations.

6.56.2 Member Function Documentation

6.56.2.1 virtual **OsclAny* Oscl_Opaque_Type_Alloc::allocate (const uint32 size) [pure virtual]**

Allocate "size" bytes

6.56.2.2 virtual void Oscl_Opaque_Type_Alloc::construct (OsclAny** * p, const **OsclAny** * init_val) [pure virtual]**

Construct element at p using element at init_val as the initial value. Both pointers must be non-NULL.

6.56.2.3 virtual void Oscl_Opaque_Type_Alloc::deallocate (OsclAny** * p) [pure virtual]**

Deallocate memory previously allocated with "allocate"

6.56.2.4 virtual void Oscl_Opaque_Type_Alloc::destroy (OsclAny** * p) [pure virtual]**

Destroy element at p.

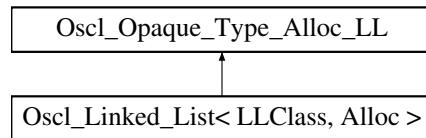
The documentation for this class was generated from the following file:

- **oscl_opaque_type.h**

6.57 Oscl_Opaque_Type_Alloc_LL Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl_Opaque_Type_Alloc_LL::



Public Methods

- virtual void `construct (OsclAny *p, const OsclAny *init_val)=0`
- virtual void `destroy (OsclAny *p)=0`
- virtual `OsclAny * allocate (const uint32 size)=0`
- virtual void `deallocate (OsclAny *p)=0`
- virtual `OsclAny * get_next (const OsclAny *elem) const=0`
- virtual void `set_next (OsclAny *elem, const OsclAny *nextelem)=0`
- virtual void `get_data (OsclAny *elem, OsclAny *data_val)=0`
- virtual bool `compare_data (const OsclAny *elem, const OsclAny *data_val) const=0`

6.57.1 Detailed Description

This class combines opaque type operations with memory allocation operations and linked list support

6.57.2 Member Function Documentation

6.57.2.1 virtual `OsclAny* Oscl_Opaque_Type_Alloc_LL::allocate (const uint32 size)` [pure virtual]

Allocate "size" bytes

6.57.2.2 virtual bool `Oscl_Opaque_Type_Alloc_LL::compare_data (const OsclAny * elem, const OsclAny * data_val) const` [pure virtual]

Compare data.

6.57.2.3 virtual void `Oscl_Opaque_Type_Alloc_LL::construct (OsclAny * p, const OsclAny * init_val)` [pure virtual]

Construct element at p using element at init_val as the initial value. Both pointers must be non-NULL.

6.57.2.4 virtual void `Oscl_Opaque_Type_Alloc_LL::deallocate (OsclAny * p)` [pure virtual]

Deallocate memory previously allocated with "allocate"

6.57.2.5 virtual void Oscl_Opaque_Type_Alloc_LL::destroy (OsclAny **p*) [pure virtual]

Destroy element at p.

6.57.2.6 virtual void Oscl_Opaque_Type_Alloc_LL::get_data (OsclAny **elem*, OsclAny **data_val*) [pure virtual]

Get data

6.57.2.7 virtual OsclAny* Oscl_Opaque_Type_Alloc_LL::get_next (const OsclAny **elem*) const [pure virtual]

Get next element in linked list.

6.57.2.8 virtual void Oscl_Opaque_Type_Alloc_LL::set_next (OsclAny **elem*, const OsclAny **nextelem*) [pure virtual]

Set next element in linked list.

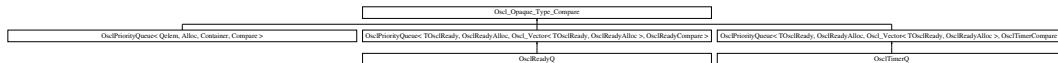
The documentation for this class was generated from the following file:

- [oscl_opaque_type.h](#)

6.58 Oscl_Opaque_Type_Compare Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl_Opaque_Type_Compare::



Public Methods

- virtual void `swap (OsclAny *a, const OsclAny *b)=0`
- virtual int `compare_LT (OsclAny *a, OsclAny *b) const=0`
- virtual int `compare_EQ (const OsclAny *a, const OsclAny *b) const=0`

6.58.1 Detailed Description

Opaque type operations with swap & comparisons.

6.58.2 Member Function Documentation

6.58.2.1 virtual int Oscl_Opaque_Type_Compare::compare_EQ (const OsclAny * a, const OsclAny * b) const [pure virtual]

Return a==b.

Implemented in `OsclPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >`.

6.58.2.2 virtual int Oscl_Opaque_Type_Compare::compare_LT (OsclAny * a, OsclAny * b) const [pure virtual]

Return a<b.

Implemented in `OsclPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >`.

6.58.2.3 virtual void Oscl_Opaque_Type_Compare::swap (OsclAny * a, const OsclAny * b) [pure virtual]

Swap element at "a" with element at "b". Both pointers must be non-NULL.

Implemented in `OsclPriorityQueue< Qelem, Alloc, Container, Compare >`, `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >`.

The documentation for this class was generated from the following file:

- [oscl_opaque_type.h](#)

6.59 Oscl_Pair< T1, T2 > Struct Template Reference

```
#include <oscl_tree.h>
```

Public Methods

- [Oscl_Pair \(\)](#)
- [Oscl_Pair \(const T1 &a, const T2 &b\)](#)

Data Fields

- [T1 first](#)
- [T2 second](#)

```
template<class T1, class T2> struct Oscl_Pair< T1, T2 >
```

6.59.1 Constructor & Destructor Documentation

6.59.1.1 template<class T1, class T2> Oscl_Pair< T1, T2 >::Oscl_Pair () [inline]

6.59.1.2 template<class T1, class T2> Oscl_Pair< T1, T2 >::Oscl_Pair (const T1 & a, const T2 & b) [inline]

6.59.2 Field Documentation

6.59.2.1 template<class T1, class T2> T1 Oscl_Pair< T1, T2 >::first

6.59.2.2 template<class T1, class T2> T2 Oscl_Pair< T1, T2 >::second

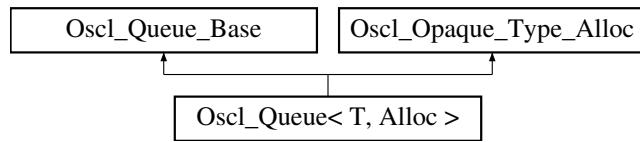
The documentation for this struct was generated from the following file:

- [oscl_tree.h](#)

6.60 Oscl_Queue< T, Alloc > Class Template Reference

```
#include <oscl_queue.h>
```

Inheritance diagram for Oscl_Queue< T, Alloc >::



Public Types

- typedef T [value_type](#)
- typedef T * [pointer](#)
- typedef T & [reference](#)
- typedef const T & [const_reference](#)
- typedef uint32 [size_type](#)

Public Methods

- [Oscl_Queue \(\)](#)
- [Oscl_Queue \(uint32 n\)](#)
- virtual [~Oscl_Queue \(\)](#)
- void [push \(const T &x\)](#)
- [reference front \(\)](#)
- [const_reference front \(\) const](#)
- void [pop \(\)](#)
- [reference back \(\)](#)
- [const_reference back \(\) const](#)
- void [clear \(\)](#)

6.60.1 Detailed Description

template<class T, class Alloc> class Oscl_Queue< T, Alloc >

Oscl_Queue Class. A subset of STL::Queue methods. Oscl_Queue supports constant time insertion (at the end) and removal of elements at the front of the queue. It does not support insertion or removal of elements at the other ends or middle of the queue, nor random access to elements. * No iteration capability is [currently] supplied. * No assignment or copy capability is [currently] supplied. The number of elements in a queue can vary dynamically, and memory management is performed automatically.

6.60.2 Member Typedef Documentation

- 6.60.2.1 `template<class T, class Alloc> typedef const T& Oscl_Queue< T, Alloc >::const_reference`
- 6.60.2.2 `template<class T, class Alloc> typedef T* Oscl_Queue< T, Alloc >::pointer`
- 6.60.2.3 `template<class T, class Alloc> typedef T& Oscl_Queue< T, Alloc >::reference`
- 6.60.2.4 `template<class T, class Alloc> typedef uint32 Oscl_Queue< T, Alloc >::size_type`
- 6.60.2.5 `template<class T, class Alloc> typedef T Oscl_Queue< T, Alloc >::value_type`

6.60.3 Constructor & Destructor Documentation

- 6.60.3.1 `template<class T, class Alloc> Oscl_Queue< T, Alloc >::Oscl_Queue () [inline]`

Creates an empty queue.

- 6.60.3.2 `template<class T, class Alloc> Oscl_Queue< T, Alloc >::Oscl_Queue (uint32 n) [inline]`

Creates an empty queue with capacity n.

Parameters:

n creates a queue with n elements. The main reason for specifying n is efficiency. If you know the capacity to which your queue must grow, then it is more efficient to allocate the queue all at once rather than rely on the automatic reallocation scheme.

- 6.60.3.3 `template<class T, class Alloc> virtual Oscl_Queue< T, Alloc >::~Oscl_Queue () [inline, virtual]`

The destructor.

6.60.4 Member Function Documentation

- 6.60.4.1 `template<class T, class Alloc> const_reference Oscl_Queue< T, Alloc >::back () const [inline]`

Returns the last element (const)

- 6.60.4.2 `template<class T, class Alloc> reference Oscl_Queue< T, Alloc >::back () [inline]`

Returns the last element: "back" (generally not too useful, but some debugging aids might want to find out what was just added)

- 6.60.4.3 `template<class T, class Alloc> void Oscl_Queue< T, Alloc >::clear () [inline]`

Removes all elements.

Reimplemented from [Oscl_Queue_Base](#).

6.60.4.4 template<class T, class Alloc> const_reference Oscl_Queue< T, Alloc >::front () const [inline]

Returns the first element (const)

6.60.4.5 template<class T, class Alloc> reference Oscl_Queue< T, Alloc >::front () [inline]

Returns the first element.

Reimplemented from [Oscl_Queue_Base](#).

6.60.4.6 template<class T, class Alloc> void Oscl_Queue< T, Alloc >::pop () [inline]

Removes the first element

Reimplemented from [Oscl_Queue_Base](#).

6.60.4.7 template<class T, class Alloc> void Oscl_Queue< T, Alloc >::push (const T & x) [inline]

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL_LEAVE will occur

Parameters:

x new element

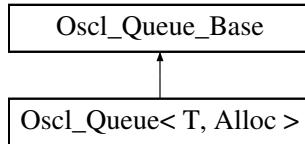
The documentation for this class was generated from the following file:

- [oscl_queue.h](#)

6.61 Oscl_Queue_Base Class Reference

```
#include <oscl_queue.h>
```

Inheritance diagram for Oscl_Queue_Base::



Public Methods

- uint32 `size () const`
- uint32 `capacity () const`
- bool `empty () const`
- OSCL_IMPORT_REF void `reserve (uint32 n)`

Protected Methods

- OSCL_IMPORT_REF void `construct (Oscl_Opaque_Type_Alloc *aType)`
- OSCL_IMPORT_REF void `construct (Oscl_Opaque_Type_Alloc *aType, uint32 n)`
- virtual `~Oscl_Queue_Base ()`
- OSCL_IMPORT_REF void `destroy ()`
- OSCL_IMPORT_REF void `push (const OsclAny *x)`
- OSCL_IMPORT_REF void `pop ()`
- OSCL_IMPORT_REF void `clear ()`

Protected Attributes

- uint32 `numelems`
- uint32 `bufsize`
- `OsclAny * elems`
- uint32 `sizeof_T`
- uint32 `ifront`
- uint32 `irear`

6.61.1 Detailed Description

`Oscl_Queue_Base` is a non-templatized base class for [Oscl_Queue](#). The purpose of this base class is to avoid large inline routines in the [Oscl_Queue](#) implementation. This class is not intended for direct instantiation except by [Oscl_Queue](#).

6.61.2 Constructor & Destructor Documentation

6.61.2.1 virtual Oscl_Queue_Base::~Oscl_Queue_Base () [inline, protected, virtual]

The destructor.

6.61.3 Member Function Documentation

6.61.3.1 **uint32 Oscl_Queue_Base::capacity () const [inline]**

Returns the allocated memory of the queue.

6.61.3.2 **OSCL_IMPORT_REF void Oscl_Queue_Base::clear () [protected]**

Removes all elements.

Reimplemented in [Oscl_Queue< T, Alloc >](#).

6.61.3.3 **OSCL_IMPORT_REF void Oscl_Queue_Base::construct (Oscl_Opaque_Type_Alloc * aType, uint32 n) [protected]**

6.61.3.4 **OSCL_IMPORT_REF void Oscl_Queue_Base::construct (Oscl_Opaque_Type_Alloc * aType) [protected]**

6.61.3.5 **OSCL_IMPORT_REF void Oscl_Queue_Base::destroy () [protected]**

Like an explicit destructor call.

6.61.3.6 **bool Oscl_Queue_Base::empty () const [inline]**

True if there are no elements in the queue

6.61.3.7 **OSCL_IMPORT_REF void Oscl_Queue_Base::pop () [protected]**

Removes the first element

Reimplemented in [Oscl_Queue< T, Alloc >](#).

6.61.3.8 **OSCL_IMPORT_REF void Oscl_Queue_Base::push (const OsclAny * x) [protected]**

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL_LEAVE will occur

Parameters:

x new element

6.61.3.9 **OSCL_IMPORT_REF void Oscl_Queue_Base::reserve (uint32 n)**

Reallocates memory if necessary to a capacity of *n* elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

Parameters:

n size of vector

6.61.3.10 uint32 Oscl_Queue_Base::size () const [inline]

Returns the size of the queue.

6.61.4 Field Documentation**6.61.4.1 uint32 Oscl_Queue_Base::bufsize [protected]****6.61.4.2 OsclAny* Oscl_Queue_Base::elems [protected]****6.61.4.3 uint32 Oscl_Queue_Base::ifront [protected]****6.61.4.4 uint32 Oscl_Queue_Base::irear [protected]****6.61.4.5 uint32 Oscl_Queue_Base::numelems [protected]****6.61.4.6 uint32 Oscl_Queue_Base::sizeof_T [protected]**

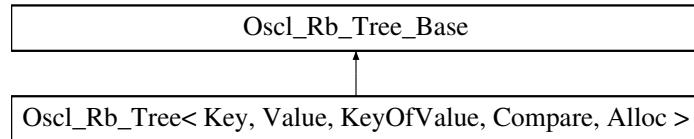
The documentation for this class was generated from the following file:

- [oscl_queue.h](#)

6.62 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::



Public Types

- `typedef Key key_type`
- `typedef Value value_type`
- `typedef value_type * pointer`
- `typedef const value_type * const_pointer`
- `typedef value_type & reference`
- `typedef const value_type & const_reference`
- `typedef Oscl_Rb_Tree_Node< Value >::link_type link_type`
- `typedef Oscl_Rb_Tree_Iterator< value_type > iterator`
- `typedef Oscl_Rb_Tree_Const_Iterator< value_type > const_iterator`
- `typedef uint32 size_type`
- `typedef int32 difference_type`

Public Methods

- `Oscl_Rb_Tree (const Compare &comp=Compare())`
- `Oscl_Rb_Tree (const Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > &x)`
- `~Oscl_Rb_Tree ()`
- `Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > & operator= (const Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > &x)`
- `iterator begin ()`
- `const_iterator begin () const`
- `iterator end ()`
- `const_iterator end () const`
- `bool empty () const`
- `size_type size () const`
- `size_type max_size () const`
- `Oscl_Pair< iterator, bool > insert_unique (const value_type &v)`
- `iterator insert_unique (iterator position, const value_type &v)`
- `void insert_unique (const iterator first, const iterator last)`
- `void insert_unique (const value_type *first, const value_type *last)`
- `void erase (iterator position)`
- `size_type erase (const key_type &x)`
- `void erase (iterator first, iterator last)`
- `void erase (const key_type *first, const key_type *last)`



6.62 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

- void [clear \(\)](#)
- [iterator find \(const Key &k\)](#)
- [const_iterator find \(const Key &k\) const](#)
- [size_type count \(const Key &k\) const](#)
- [iterator lower_bound \(const Key &k\)](#)
- [const_iterator lower_bound \(const Key &k\) const](#)
- [iterator upper_bound \(const Key &k\)](#)
- [const_iterator upper_bound \(const Key &k\) const](#)
- [Oscl_Pair< iterator, iterator > equal_range \(const Key &k\)](#)
- [Oscl_Pair< const_iterator, const_iterator > equal_range \(const Key &k\) const](#)

template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> class Oscl_Rb_-Tree< Key, Value, KeyOfValue, Compare, Alloc >

6.62.1 Member Typedef Documentation

- 6.62.1.1 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [Oscl_Rb_Tree_Const_Iterator<value_type>](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const_iterator
- 6.62.1.2 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef const [value_type*](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const_pointer
- 6.62.1.3 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef const [value_type&](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const_reference
- 6.62.1.4 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef int32 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::difference_type
- 6.62.1.5 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [Oscl_Rb_Tree_Iterator<value_type>](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::iterator
- 6.62.1.6 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Key Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::key_type
- 6.62.1.7 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [Oscl_Rb_Tree_Node<Value>::link_type](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::link_type
- 6.62.1.8 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [value_type*](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::pointer
- 6.62.1.9 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef [value_type&](#) Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::reference
- 6.62.1.10 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef uint32 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::size_type
- 6.62.1.11 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Value Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::value_type

6.62.2 Constructor & Destructor Documentation

- 6.62.2.1 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::Oscl_Rb_Tree (const Compare & *comp* = Compare() [inline])
- 6.62.2.2 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::Oscl_Rb_Tree (const Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > & *x*) [inline]
- 6.62.2.3 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::~Oscl_Rb_Tree () [inline]



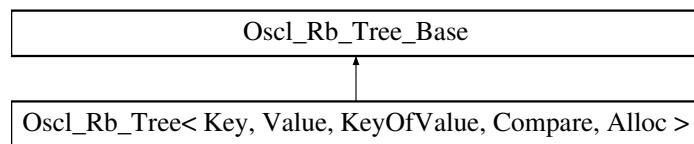
6.62 Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > Class Template Reference

- [oscl_tree.h](#)

6.63 Oscl_Rb_Tree_Base Class Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl_Rb_Tree_Base::



Public Types

- `typedef Oscl_Rb_Tree_Node_Base::base_link_type base_link_type`

Public Methods

- `OSCL_IMPORT_REF void rotate_left (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF void rotate_right (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF void rebalance (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF base_link_type rebalance_for_erase (base_link_type z, base_link_type &root, base_link_type &leftmost, base_link_type &rightmost)`

6.63.1 Member Typedef Documentation

6.63.1.1 `typedef Oscl_Rb_Tree_Node_Base::base_link_type Oscl_Rb_Tree_Base::base_link_type`

6.63.2 Member Function Documentation

6.63.2.1 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rebalance (base_link_type x, base_link_type &root)`

6.63.2.2 `OSCL_IMPORT_REF base_link_type Oscl_Rb_Tree_Base::rebalance_for_erase (base_link_type z, base_link_type &root, base_link_type &leftmost, base_link_type &rightmost)`

6.63.2.3 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rotate_left (base_link_type x, base_link_type &root)`

6.63.2.4 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rotate_right (base_link_type x, base_link_type &root)`

The documentation for this class was generated from the following file:

- `oscl_tree.h`

6.64 Oscl_Rb_Tree_Const_Iterator< Value > Struct Template Reference

```
#include <oscl_tree.h>
```

Public Types

- `typedef Value value_type`
- `typedef const value_type & reference`
- `typedef const value_type * pointer`
- `typedef Oscl_Rb_Tree_Const_Iterator< Value > const_iterator`
- `typedef Oscl_Rb_Tree_Const_Iterator< Value > self`
- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef Oscl_Rb_Tree_Node< Value > * link_type`

Public Methods

- `Oscl_Rb_Tree_Const_Iterator()`
- `Oscl_Rb_Tree_Const_Iterator(link_type x)`
- `Oscl_Rb_Tree_Const_Iterator(const const_iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator==(const self &x)`
- `bool operator!=(const self &x)`
- `self & operator++()`
- `self operator++(int)`
- `self & operator--()`
- `self operator--(int)`

Data Fields

- `base_link_type node`

```
template<class Value> struct Oscl_Rb_Tree_Const_Iterator< Value >
```

6.64.1 Member Typedef Documentation

- 6.64.1.1 `template<class Value> typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Const_Iterator< Value >::base_link_type`
- 6.64.1.2 `template<class Value> typedef Oscl_Rb_Tree_Const_Iterator<Value> Oscl_Rb_Tree_Const_Iterator< Value >::const_iterator`
- 6.64.1.3 `template<class Value> typedef Oscl_Rb_Tree_Node<Value>* Oscl_Rb_Tree_Const_Iterator< Value >::link_type`
- 6.64.1.4 `template<class Value> typedef const value_type* Oscl_Rb_Tree_Const_Iterator< Value >::pointer`
- 6.64.1.5 `template<class Value> typedef const value_type& Oscl_Rb_Tree_Const_Iterator< Value >::reference`
- 6.64.1.6 `template<class Value> typedef Oscl_Rb_Tree_Const_Iterator<Value> Oscl_Rb_Tree_Const_Iterator< Value >::self`
- 6.64.1.7 `template<class Value> typedef Value Oscl_Rb_Tree_Const_Iterator< Value >::value_type`

6.64.2 Constructor & Destructor Documentation

- 6.64.2.1 `template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator () [inline]`
- 6.64.2.2 `template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator (link_type x) [inline]`
- 6.64.2.3 `template<class Value> Oscl_Rb_Tree_Const_Iterator< Value >::Oscl_Rb_Tree_Const_Iterator (const const_iterator & it) [inline]`

6.64.3 Member Function Documentation

- 6.64.3.1 `template<class Value> reference Oscl_Rb_Tree_Const_Iterator< Value >::operator * () const [inline]`
- 6.64.3.2 `template<class Value> bool Oscl_Rb_Tree_Const_Iterator< Value >::operator!= (const self & x) [inline]`
- 6.64.3.3 `template<class Value> self Oscl_Rb_Tree_Const_Iterator< Value >::operator++ (int) [inline]`
- 6.64.3.4 `template<class Value> self& Oscl_Rb_Tree_Const_Iterator< Value >::operator++ () [inline]`
- 6.64.3.5 `template<class Value> self Oscl_Rb_Tree_Const_Iterator< Value >::operator- (int) [inline]`
- 6.64.3.6 `template<class Value> self& Oscl_Rb_Tree_Const_Iterator< Value >::operator- () [inline]`

-
- [oscl_tree.h](#)

6.65 Oscl_Rb_Tree_Iterator< Value > Struct Template Reference

```
#include <oscl_tree.h>
```

Public Types

- `typedef Value value_type`
- `typedef value_type & reference`
- `typedef value_type * pointer`
- `typedef Oscl_Rb_Tree_Iterator< Value > iterator`
- `typedef Oscl_Rb_Tree_Iterator< Value > self`
- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef Oscl_Rb_Tree_Node< Value > * link_type`

Public Methods

- `Oscl_Rb_Tree_Iterator ()`
- `Oscl_Rb_Tree_Iterator (link_type x)`
- `Oscl_Rb_Tree_Iterator (const iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator== (const self &x)`
- `bool operator!= (const self &x)`
- `self & operator++ ()`
- `self operator++ (int)`
- `self & operator– ()`
- `self operator– (int)`

Data Fields

- `base_link_type node`

template<class Value> struct Oscl_Rb_Tree_Iterator< Value >

6.65.1 Member Typedef Documentation

- 6.65.1.1 template<class Value> typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Iterator< Value >::base_link_type
- 6.65.1.2 template<class Value> typedef Oscl_Rb_Tree_Iterator<Value> Oscl_Rb_Tree_Iterator< Value >::iterator
- 6.65.1.3 template<class Value> typedef Oscl_Rb_Tree_Node<Value>* Oscl_Rb_Tree_Iterator< Value >::link_type
- 6.65.1.4 template<class Value> typedef value_type* Oscl_Rb_Tree_Iterator< Value >::pointer
- 6.65.1.5 template<class Value> typedef value_type& Oscl_Rb_Tree_Iterator< Value >::reference
- 6.65.1.6 template<class Value> typedef Oscl_Rb_Tree_Iterator<Value> Oscl_Rb_Tree_Iterator< Value >::self
- 6.65.1.7 template<class Value> typedef Value Oscl_Rb_Tree_Iterator< Value >::value_type

6.65.2 Constructor & Destructor Documentation

- 6.65.2.1 template<class Value> Oscl_Rb_Tree_Iterator< Value >::Oscl_Rb_Tree_Iterator () [inline]
- 6.65.2.2 template<class Value> Oscl_Rb_Tree_Iterator< Value >::Oscl_Rb_Tree_Iterator (link_type x) [inline]
- 6.65.2.3 template<class Value> Oscl_Rb_Tree_Iterator< Value >::Oscl_Rb_Tree_Iterator (const iterator & it) [inline]

6.65.3 Member Function Documentation

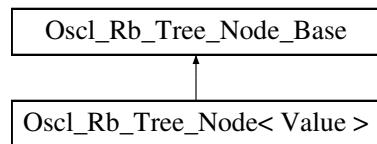
- 6.65.3.1 template<class Value> reference Oscl_Rb_Tree_Iterator< Value >::operator * () const [inline]
- 6.65.3.2 template<class Value> bool Oscl_Rb_Tree_Iterator< Value >::operator!= (const self & x) [inline]
- 6.65.3.3 template<class Value> self Oscl_Rb_Tree_Iterator< Value >::operator++ (int) [inline]
- 6.65.3.4 template<class Value> self& Oscl_Rb_Tree_Iterator< Value >::operator++ () [inline]
- 6.65.3.5 template<class Value> self Oscl_Rb_Tree_Iterator< Value >::operator- (int) [inline]
- 6.65.3.6 template<class Value> self& Oscl_Rb_Tree_Iterator< Value >::operator- () [inline]
- 6.65.3.7 template<class Value> pointer Oscl_Rb_Tree_Iterator< Value >::operator -> () const [inline]

- [oscl_tree.h](#)

6.66 Oscl_Rb_Tree_Node< Value > Struct Template Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl_Rb_Tree_Node< Value >::



Public Types

- [typedef Value value_type](#)
- [typedef Oscl_Rb_Tree_Node< Value > * link_type](#)

Data Fields

- [value_type value](#)

```
template<class Value> struct Oscl_Rb_Tree_Node< Value >
```

6.66.1 Member Typedef Documentation

6.66.1.1 template<class Value> typedef Oscl_Rb_Tree_Node<Value>* Oscl_Rb_Tree_Node< Value >::link_type

6.66.1.2 template<class Value> typedef Value Oscl_Rb_Tree_Node< Value >::value_type

6.66.2 Field Documentation

6.66.2.1 template<class Value> [value_type](#) Oscl_Rb_Tree_Node< Value >::value

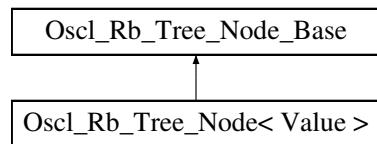
The documentation for this struct was generated from the following file:

- [oscl_tree.h](#)

6.67 Oscl_Rb_Tree_Node_Base Struct Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl_Rb_Tree_Node_Base::



Public Types

- `typedef Oscl_Rb_Tree_Node_Base * base_link_type`
- `typedef enum Oscl_Rb_Tree_Node_Base::RedBl color_type`
- `enum RedBl { red, black }`

Static Public Methods

- `base_link_type minimum (base_link_type x)`
- `base_link_type maximum (base_link_type x)`

Data Fields

- `color_type color`
- `base_link_type parent`
- `base_link_type left`
- `base_link_type right`

6.67.1 Member Typedef Documentation

6.67.1.1 `typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Node_Base::base_link_type`

6.67.1.2 `typedef enum Oscl_Rb_Tree_Node_Base::RedBl Oscl_Rb_Tree_Node_Base::color_type`

6.67.2 Member Enumeration Documentation

6.67.2.1 `enum Oscl_Rb_Tree_Node_Base::RedBl`

Enumeration values:

`red`

`black`

6.67.3 Member Function Documentation

6.67.3.1 **base_link_type** Oscl_Rb_Tree_Node_Base::maximum (**base_link_type** *x*) [inline, static]

6.67.3.2 **base_link_type** Oscl_Rb_Tree_Node_Base::minimum (**base_link_type** *x*) [inline, static]

6.67.4 Field Documentation

6.67.4.1 **color_type** Oscl_Rb_Tree_Node_Base::color

6.67.4.2 **base_link_type** Oscl_Rb_Tree_Node_Base::left

6.67.4.3 **base_link_type** Oscl_Rb_Tree_Node_Base::parent

6.67.4.4 **base_link_type** Oscl_Rb_Tree_Node_Base::right

The documentation for this struct was generated from the following file:

- [oscl_tree.h](#)

6.68 Oscl_Select1st< V, U > Struct Template Reference

```
#include <oscl_map.h>
```

Public Methods

- const U & [operator\(\)](#) (const V &x) const

```
template<class V, class U> struct Oscl_Select1st< V, U >
```

6.68.1 Member Function Documentation

**6.68.1.1 template<class V, class U> const U& Oscl_Select1st< V, U >::operator() (const V & x)
const [inline]**

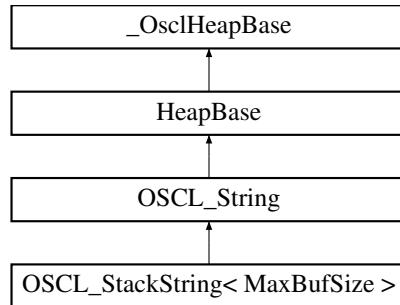
The documentation for this struct was generated from the following file:

- [oscl_map.h](#)

6.69 OSCL_StackString< MaxBufSize > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_StackString< MaxBufSize >::



Public Types

- `typedef OSCL_String::chartype chartype`

Public Methods

- `OSCL_StackString ()`
- `OSCL_StackString (const OSCL_StackString &src)`
- `OSCL_StackString (const OSCL_String &src)`
- `OSCL_StackString (const chartype *cstr)`
- `OSCL_StackString (const chartype *buf, uint32 length)`
- `~OSCL_StackString ()`
- `uint32 get_size () const`
- `uint32 get_maxsize () const`
- `const chartype * get_cstr () const`
- `chartype * get_str () const`
- `OSCL_StackString & operator= (const OSCL_StackString &src)`
- `OSCL_StackString & operator= (const OSCL_String &src)`
- `OSCL_StackString & operator= (const chartype *cstr)`
- `void set (const chartype *buf, uint32 length)`

Friends

- class `OSCL_String`

6.69.1 Detailed Description

`template<uint32 MaxBufSize> class OSCL_StackString< MaxBufSize >`

`OSCL_StackString` is a simple string class, compatible with regular character array strings.

The string array is fixed length, is allocated from the stack, and is modifiable. Operations that update the string will automatically truncate it to fit the fixed size storage. This is recommended for use for short strings (<255). Use `OSCL_HeapString` for very large strings to avoid stack overflow.

Parameters:

C: type of character.

MaxBufSize: maximum string length not including null terminator.

6.69.2 Member Typedef Documentation

6.69.2.1 template<uint32 MaxBufSize> typedef OSCL_String::chartype OSCL_StackString< MaxBufSize >::chartype

Reimplemented from [OSCL_String](#).

6.69.3 Friends And Related Function Documentation

6.69.3.1 template<uint32 MaxBufSize> friend class OSCL_String [friend]

The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

6.70 oscl_stat_buf Struct Reference

```
#include <oscl_file_dir_utils.h>
```

Data Fields

- uint32 [mode](#)
- uint32 [perms](#)

6.70.1 Field Documentation

6.70.1.1 uint32 oscl_stat_buf::mode

6.70.1.2 uint32 oscl_stat_buf::perms

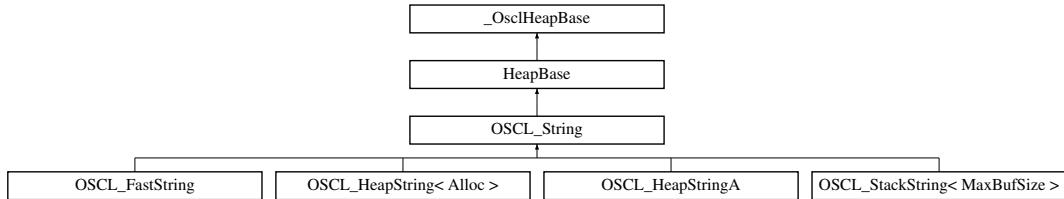
The documentation for this struct was generated from the following file:

- [oscl_file_dir_utils.h](#)

6.71 OSCL_String Class Reference

```
#include <oscl_string.h>
```

Inheritance diagram for OSCL_String::



Public Types

- `typedef char chartype`

Public Methods

- `virtual uint32 get_size () const=0`
- `virtual uint32 get_maxsize () const=0`
- `virtual const chartype * get_cstr () const=0`
- `virtual OSCL_IMPORT_REF bool is_writable () const`
- `virtual chartype * get_str () const=0`
- `OSCL_IMPORT_REF OSCL_String & operator= (const OSCL_String &src)`
- `OSCL_IMPORT_REF OSCL_String & operator= (const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_String & operator+= (const OSCL_String &src)`
- `OSCL_IMPORT_REF OSCL_String & operator+= (const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_String & operator+= (const chartype c)`
- `OSCL_IMPORT_REF bool operator== (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator!= (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator< (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator<= (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator> (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator>= (const OSCL_String &src) const`
- `OSCL_IMPORT_REF bool operator== (const chartype *cstr) const`
- `OSCL_IMPORT_REF chartype operator[] (uint32 index) const`
- `virtual OSCL_IMPORT_REF chartype read (uint32 index) const`
- `virtual OSCL_IMPORT_REF int8 hash () const`
- `virtual OSCL_IMPORT_REF void write (uint32 index, chartype c)`
- `virtual OSCL_IMPORT_REF void write (uint32 offset, uint32 length, const chartype *buf)`

Protected Methods

- `OSCL_IMPORT_REF OSCL_String ()`
- `virtual OSCL_IMPORT_REF ~OSCL_String ()`
- `virtual void set_rep (const chartype *cstr)=0`
- `virtual void append_rep (const chartype *cstr)=0`
- `virtual void set_rep (const OSCL_String &src)=0`
- `virtual void append_rep (const OSCL_String &src)=0`
- `virtual void set_len (uint32 len)=0`

6.71.1 Detailed Description

A common base class for string classes with "char" character format

6.71.2 Member Typedef Documentation

6.71.2.1 `typedef char OSCL_String::chartype`

Reimplemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsclMemAllocator >](#).

6.71.3 Constructor & Destructor Documentation

6.71.3.1 `OSCL_IMPORT_REF OSCL_String::OSCL_String()` [protected]

6.71.3.2 `virtual OSCL_IMPORT_REF OSCL_String::~OSCL_String()` [protected, virtual]

6.71.4 Member Function Documentation

6.71.4.1 `virtual void OSCL_String::append_rep(const OSCL_String & src)` [protected, pure virtual]

Append the input string to the current string. The string may be truncated to fit the available storage.

6.71.4.2 `virtual void OSCL_String::append_rep(const chartype * cstr)` [protected, pure virtual]

Append the input null-terminated string to the current string. The string may be truncated to fit the available storage.

6.71.4.3 `virtual const chartype* OSCL_String::get_cstr()` [pure virtual]

This function returns the C-style string for read access.

Implemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsclMemAllocator >](#).

6.71.4.4 `virtual uint32 OSCL_String::get_maxsize()` [pure virtual]

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsclMemAllocator >](#).

6.71.4.5 `virtual uint32 OSCL_String::get_size()` [pure virtual]

This function returns the string size not including the null-terminator.

Implemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsclMemAllocator >](#).

6.71.4.6 virtual **chartype*** OSCL_String::get_str () [pure virtual]

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsclMemAllocator >](#).

6.71.4.7 virtual OSCL_IMPORT_REF int8 OSCL_String::hash () [virtual]

This function performs a hash operation on the string. If the string is not writable, the function leaves.

6.71.4.8 virtual OSCL_IMPORT_REF bool OSCL_String::is_writable () [virtual]

This function returns true if the string is writable.

6.71.4.9 OSCL_IMPORT_REF bool OSCL_String::operator!= (const OSCL_String & src) const

6.71.4.10 OSCL_IMPORT_REF OSCL_String& OSCL_String::operator+= (const **chartype** c)

Append operator. This operator appends the input character to this object. The string may be truncated to fit available storage.

6.71.4.11 OSCL_IMPORT_REF OSCL_String& OSCL_String::operator+= (const **chartype** * cstr)

Append operator. This operator appends the input string to this object. The string may be truncated to fit available storage.

am: null-terminated string

6.71.4.12 OSCL_IMPORT_REF OSCL_String& OSCL_String::operator+= (const OSCL_String & src)

Append operator. This operator appends the input string to this object. The string may be truncated to fit available storage.

6.71.4.13 OSCL_IMPORT_REF bool OSCL_String::operator< (const OSCL_String & src) const

6.71.4.14 OSCL_IMPORT_REF bool OSCL_String::operator<= (const OSCL_String & src) const

6.71.4.15 OSCL_IMPORT_REF OSCL_String& OSCL_String::operator= (const **chartype** * cstr)

Assignment operator

am: null-terminated string

Reimplemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), [OSCL_FastString](#), and [OSCL_HeapString< OsclMemAllocator >](#).

6.71.4.16 **OSCL_IMPORT_REF OSCL_String& OSCL_String::operator= (const OSCL_String & src)**

Assignment operator

Reimplemented in [OSCL_HeapString< Alloc >](#), [OSCL_HeapStringA](#), [OSCL_StackString< MaxBufSize >](#), and [OSCL_HeapString< OsclMemAllocator >](#).

6.71.4.17 **OSCL_IMPORT_REF bool OSCL_String::operator== (const chartype * cstr) const**

Comparison operator

am: null-terminated string

6.71.4.18 **OSCL_IMPORT_REF bool OSCL_String::operator== (const OSCL_String & src) const**

Comparison operators

6.71.4.19 **OSCL_IMPORT_REF bool OSCL_String::operator> (const OSCL_String & src) const**

6.71.4.20 **OSCL_IMPORT_REF bool OSCL_String::operator>= (const OSCL_String & src) const**

6.71.4.21]

OSCL_IMPORT_REF chartype OSCL_String::operator[] (uint32 index) const

This is subscript notation to access a character at the given position. If the index is outside the current size range then the function leaves.

6.71.4.22 **virtual OSCL_IMPORT_REF chartype OSCL_String::read (uint32 index) const [virtual]**

This function returns the character at the given position. If the index is outside the current size range then the function leaves.

6.71.4.23 **virtual void OSCL_String::set_len (uint32 len) [protected, pure virtual]**

Update the length of the string. This function will only be called when the string is writable.

6.71.4.24 virtual void OSCL_String::set_rep (const OSCL_String & src) [protected, pure virtual]

Set string representation to input string.

6.71.4.25 virtual void OSCL_String::set_rep (const chartype * cstr) [protected, pure virtual]

Set string representation to input null-terminated string.

6.71.4.26 virtual OSCL_IMPORT_REF void OSCL_String::write (uint32 offset, uint32 length, const chartype * buf) [virtual]

This function replaces characters at the specified offset within the current string. If the string is not writable, the function leaves. The characters may be truncated to fit the current storage.

Parameters:

offset: the offset into the existing string buffer

length: number of characters to copy.

ptr: character buffer, not necessarily null-terminated.

6.71.4.27 virtual OSCL_IMPORT_REF void OSCL_String::write (uint32 index, chartype c) [virtual]

This function stores a character at the specified position. If the string is not writable, the function leaves. If the index is outside the current size range then the function leaves.

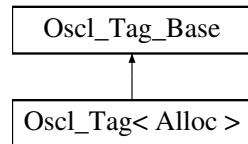
The documentation for this class was generated from the following file:

- [oscl_string.h](#)

6.72 Oscl_Tag< Alloc > Struct Template Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oscl_Tag< Alloc >::



Public Methods

- [Oscl_Tag](#) (const Oscl_Tag< Alloc > &t)
- [Oscl_Tag](#) (const [tag_base_type](#) &t)
- [~Oscl_Tag](#) ()
- bool [operator<](#) (const Oscl_Tag< Alloc > &x) const

Data Fields

- [Oscl_TAlloc< tag_base_unit, Alloc > tagAllocator](#)
- [tag_base_type tag](#)

```
template<class Alloc> struct Oscl_Tag< Alloc >
```

6.72.1 Constructor & Destructor Documentation

6.72.1.1 [template<class Alloc> Oscl_Tag< Alloc >::Oscl_Tag \(const Oscl_Tag< Alloc > & t\)](#) [inline]

6.72.1.2 [template<class Alloc> Oscl_Tag< Alloc >::Oscl_Tag \(const tag_base_type & t\)](#) [inline]

6.72.1.3 [template<class Alloc> Oscl_Tag< Alloc >::~Oscl_Tag \(\)](#) [inline]

6.72.2 Member Function Documentation

6.72.2.1 [template<class Alloc> bool Oscl_Tag< Alloc >::operator< \(const Oscl_Tag< Alloc > & x\) const](#) [inline]

6.72.3 Field Documentation

6.72.3.1 [template<class Alloc> tag_base_type Oscl_Tag< Alloc >::tag](#)

6.72.3.2 [template<class Alloc> Oscl_TAlloc<tag_base_unit, Alloc> Oscl_Tag< Alloc >::tagAllocator](#)

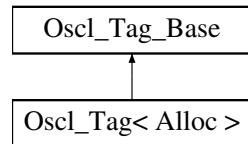
The documentation for this struct was generated from the following file:

- [oscl_tagtree.h](#)

6.73 Oscl_Tag_Base Struct Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oscl_Tag_Base::



Public Types

- `typedef char tag_base_unit`
- `typedef tag_base_unit * tag_base_type`
- `typedef uint32 size_type`

Public Methods

- `bool operator() (const tag_base_type &x, const tag_base_type &y) const`
- `size_type tag_len (const tag_base_type &t) const`
- `tag_base_type tag_copy (tag_base_type &dest, const tag_base_type &src) const`
- `int32 tag_cmp (const tag_base_type &x, const tag_base_type &y) const`
- `OSCL_IMPORT_REF tag_base_type tag_ancestor (tag_base_type &dest, const tag_base_type &src) const`
- `OSCL_IMPORT_REF size_type tag_depth (const tag_base_type &t) const`

6.73.1 Member Typedef Documentation

6.73.1.1 `typedef uint32 Oscl_Tag_Base::size_type`

6.73.1.2 `typedef tag_base_unit* Oscl_Tag_Base::tag_base_type`

6.73.1.3 `typedef char Oscl_Tag_Base::tag_base_unit`

6.73.2 Member Function Documentation

6.73.2.1 `bool Oscl_Tag_Base::operator() (const tag_base_type & x, const tag_base_type & y) const [inline]`

6.73.2.2 `OSCL_IMPORT_REF tag_base_type Oscl_Tag_Base::tag_ancestor (tag_base_type & dest, const tag_base_type & src) const`

6.73.2.3 `int32 Oscl_Tag_Base::tag_cmp (const tag_base_type & x, const tag_base_type & y) const [inline]`

6.73.2.4 `tag_base_type Oscl_Tag_Base::tag_copy (tag_base_type & dest, const tag_base_type & src) const [inline]`

6.73.2.5 `OSCL_IMPORT_REF size_type Oscl_Tag_Base::tag_depth (const tag_base_type & t) const`

6.73.2.6 `size_type Oscl_Tag_Base::tag_len (const tag_base_type & t) const [inline]`

The documentation for this struct was generated from the following file:

- [oscl_tagtree.h](#)

6.74 Oscl_TagTree< T, Alloc > Class Template Reference

```
#include <oscl_tagtree.h>
```

Public Types

- typedef [Oscl_Tag< Alloc > tag_type](#)
- typedef tag_type::tag_base_type [tag_base_type](#)
- typedef [Oscl_Vector< Node *, Alloc > children_type](#)
- typedef [Node node_type](#)
- typedef [node_type * node_ptr](#)
- typedef [Oscl_Map< const tag_base_type, node_ptr, Alloc, Oscl_Tag_Base > map_type](#)
- typedef map_type::size_type [size_type](#)
- typedef map_type::value_type [value_type](#)
- typedef [Oscl_Pair< iterator, bool > pair_iterator_bool](#)

Public Methods

- [Oscl_TagTree \(size_type max_depth=0\)](#)
- [Oscl_TagTree \(const Oscl_TagTree< T, Alloc > &x\)](#)
- [Oscl_TagTree< T, Alloc > & operator= \(const Oscl_TagTree< T, Alloc > &x\)](#)
- [~Oscl_TagTree \(\)](#)
- [iterator begin \(\)](#)
- [const_iterator begin \(\) const](#)
- [iterator end \(\)](#)
- [const_iterator end \(\) const](#)
- [bool empty \(\) const](#)
- [size_type size \(\) const](#)
- [T & operator\[\] \(const tag_base_type &t\)](#)
- [pair_iterator_bool insert \(const tag_base_type &t, const T &x\)](#)
- [void erase \(iterator position\)](#)
- [size_type erase \(const tag_base_type &x\)](#)
- [void clear \(\)](#)
- [iterator find \(const tag_base_type &x\)](#)
- [size_type count \(const tag_base_type &x\) const](#)

6.74.1 Detailed Description

template<class T, class Alloc> class Oscl_TagTree< T, Alloc >

Oscl_TagTree Class.

6.74.2 Member Typedef Documentation

- 6.74.2.1 template<class T, class Alloc> typedef **Oscl_Vector<Node*, Alloc>** Oscl_TagTree< T, Alloc >::children_type
- 6.74.2.2 template<class T, class Alloc> typedef **Oscl_Map<const tag_base_type, node_ptr, Alloc, Oscl_Tag_Base>** Oscl_TagTree< T, Alloc >::map_type
- 6.74.2.3 template<class T, class Alloc> typedef **node_type*** Oscl_TagTree< T, Alloc >::node_ptr
- 6.74.2.4 template<class T, class Alloc> typedef **Node** Oscl_TagTree< T, Alloc >::node_type
- 6.74.2.5 template<class T, class Alloc> typedef **Oscl_Pair<iterator, bool>** Oscl_TagTree< T, Alloc >::pair_iterator_bool
- 6.74.2.6 template<class T, class Alloc> typedef map_type::size_type Oscl_TagTree< T, Alloc >::size_type
- 6.74.2.7 template<class T, class Alloc> typedef tag_type::tag_base_type Oscl_TagTree< T, Alloc >::tag_base_type
- 6.74.2.8 template<class T, class Alloc> typedef **Oscl_Tag<Alloc>** Oscl_TagTree< T, Alloc >::tag_type
- 6.74.2.9 template<class T, class Alloc> typedef map_type::value_type Oscl_TagTree< T, Alloc >::value_type

6.74.3 Constructor & Destructor Documentation

- 6.74.3.1 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::Oscl_TagTree (**size_type max_depth = 0**) [inline]

Creates a tag tree with only a root node with tag ""

- 6.74.3.2 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::Oscl_TagTree (const Oscl_TagTree< T, Alloc > & x) [inline]

Copy constructor

- 6.74.3.3 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::~Oscl_TagTree () [inline]

Destructor

6.74.4 Member Function Documentation

- 6.74.4.1 template<class T, class Alloc> const_iterator Oscl_TagTree< T, Alloc >::begin () const [inline]

Returns an iterator pointing to the first node in the tree.

6.74.4.2 template<class T, class Alloc> iterator Oscl_TagTree< T, Alloc >::begin () [inline]

Returns an iterator pointing to the first node in the tree.

6.74.4.3 template<class T, class Alloc> void Oscl_TagTree< T, Alloc >::clear () [inline]

Erases the entire tag tree.

6.74.4.4 template<class T, class Alloc> size_type Oscl_TagTree< T, Alloc >::count (const tag_base_type & x) const [inline]

Returns the number of elements with key x. This can only be 0 or 1..

6.74.4.5 template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::empty () const [inline]

Returns true if tree size is 0

6.74.4.6 template<class T, class Alloc> const_iterator Oscl_TagTree< T, Alloc >::end () const [inline]

Returns a const iterator pointing to the end of the tree.

6.74.4.7 template<class T, class Alloc> iterator Oscl_TagTree< T, Alloc >::end () [inline]

Returns an iterator pointing to the end of the tree.

6.74.4.8 template<class T, class Alloc> size_type Oscl_TagTree< T, Alloc >::erase (const tag_base_type & x) [inline]

Erases the node with tag x. If the node has children, then the node will not be erased from the tree. It will be replaced with the default node value

Parameters:

x Tag of node to erase

Returns:

Returns the number of nodes erased. Since one-to-one mapping between nodes and tags, this will be either 0 or 1

6.74.4.9 template<class T, class Alloc> void Oscl_TagTree< T, Alloc >::erase (iterator position) [inline]

Erases the element pointed to by the iterator. If the node has children, then the node will not be erased from the tree. It will be replaced with the default node value.

Parameters:

position Iterator pointing to the node to be erased

6.74.4.10 template<class T, class Alloc> iterator Oscl_TagTree< T, Alloc >::find (const tag_base_type & x) [inline]

Finds an element whose key is x

Returns:

returns an iterator to the element with key x. If no element is found, returns [end\(\)](#)

6.74.4.11 template<class T, class Alloc> pair_iterator_bool Oscl_TagTree< T, Alloc >::insert (const tag_base_type & t, const T & x) [inline]

Inserts x into the tree and associates it with tag t. If the tag already exists x will not be inserted, and an iterator pointing to the existing node with tag t is returned.

Parameters:

t tag to use

x element to insert

Returns:

Returns a pair of parameters, iterator and bool. The iterator points to the inserted node containing x. If the tag t already existed, then the iterator points to the node associated with tag t. The bool is true if x was inserted and false if it was not inserted due to an existing node with tag t.

6.74.4.12 template<class T, class Alloc> Oscl_TagTree<T, Alloc>& Oscl_TagTree< T, Alloc >::operator= (const Oscl_TagTree< T, Alloc > & x) [inline]

Assignment operator

6.74.4.13]

template<class T, class Alloc> T& Oscl_TagTree< T, Alloc >::operator[] (const tag_base_type & t) [inline]

Returns a reference to the object that is associated with a particular tag. If the map does not already contain such an object, operator[] inserts the default object T().

6.74.4.14 template<class T, class Alloc> size_type Oscl_TagTree< T, Alloc >::size () const [inline]

Returns the number of nodes stored in the tree

The documentation for this class was generated from the following file:

- [oscl_tagtree.h](#)

6.75 Oscl_TagTree< T, Alloc >::const_iterator Struct Reference

```
#include <oscl_tagtree.h>
```

Public Types

- `typedef const node_type & reference`
- `typedef const node_type * pointer`
- `typedef map_type::const_iterator mapiter`
- `typedef const_iterator self`

Public Methods

- `const_iterator()`
- `const_iterator(mapiter x)`
- `const_iterator(const const_iterator &it)`
- `reference operator*() const`
- `pointer operator->() const`
- `bool operator==(const self &x)`
- `bool operator!=(const self &x)`
- `self & operator++()`
- `self operator++(int)`
- `self & operator--()`
- `self operator--(int)`

Data Fields

- `mapiter mapit`

template<class T, class Alloc> struct Oscl_TagTree< T, Alloc >::const_iterator

6.75.1 Member Typedef Documentation

- 6.75.1.1 template<class T, class Alloc> typedef map_type::const_iterator Oscl_TagTree< T, Alloc >::const_iterator::mapiter
- 6.75.1.2 template<class T, class Alloc> typedef const node_type* Oscl_TagTree< T, Alloc >::const_iterator::pointer
- 6.75.1.3 template<class T, class Alloc> typedef const node_type& Oscl_TagTree< T, Alloc >::const_iterator::reference
- 6.75.1.4 template<class T, class Alloc> typedef const_iterator Oscl_TagTree< T, Alloc >::const_iterator::self

6.75.2 Constructor & Destructor Documentation

- 6.75.2.1 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::const_iterator::const_iterator() [inline]
- 6.75.2.2 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::const_iterator::const_iterator(mapiter x) [inline]
- 6.75.2.3 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::const_iterator::const_iterator(const const_iterator & it) [inline]

6.75.3 Member Function Documentation

- 6.75.3.1 template<class T, class Alloc> reference Oscl_TagTree< T, Alloc >::const_iterator::operator *() const [inline]
- 6.75.3.2 template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::const_iterator::operator!= (const self & x) [inline]
- 6.75.3.3 template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::const_iterator::operator++(int) [inline]
- 6.75.3.4 template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::const_iterator::operator++() [inline]
- 6.75.3.5 template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::const_iterator::operator-(int) [inline]
- 6.75.3.6 template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::const_iterator::operator-() [inline]
- 6.75.3.7 template<class T, class Alloc> pointer Oscl_TagTree< T, Alloc >::const_iterator::operator-> () const [inline]
- 6.75.3.8 template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::const_iterator::operator==(const self & x) [inline]

6.75.4 Field Documentation

- [oscl_tagtree.h](#)

6.76 Oscl_TagTree< T, Alloc >::iterator Struct Reference

```
#include <oscl_tagtree.h>
```

Public Types

- `typedef node_type & reference`
- `typedef node_type * pointer`
- `typedef map_type::iterator mapiter`
- `typedef iterator self`

Public Methods

- `iterator ()`
- `iterator (mapiter x)`
- `iterator (const iterator &it)`
- `reference operator * () const`
- `pointer operator → () const`
- `bool operator== (const self &x)`
- `bool operator!= (const self &x)`
- `self & operator++ ()`
- `self operator++ (int)`
- `self & operator– ()`
- `self operator– (int)`

Data Fields

- `mapiter mapit`

template<class T, class Alloc> struct Oscl_TagTree< T, Alloc >::iterator

6.76.1 Member Typedef Documentation

- 6.76.1.1 template<class T, class Alloc> typedef map_type::iterator Oscl_TagTree< T, Alloc >::iterator::mapiter
- 6.76.1.2 template<class T, class Alloc> typedef node_type* Oscl_TagTree< T, Alloc >::iterator::pointer
- 6.76.1.3 template<class T, class Alloc> typedef node_type& Oscl_TagTree< T, Alloc >::iterator::reference
- 6.76.1.4 template<class T, class Alloc> typedef iterator Oscl_TagTree< T, Alloc >::iterator::self

6.76.2 Constructor & Destructor Documentation

- 6.76.2.1 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::iterator::iterator () [inline]
- 6.76.2.2 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::iterator::iterator (mapiter x) [inline]
- 6.76.2.3 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::iterator::iterator (const iterator & it) [inline]

6.76.3 Member Function Documentation

- 6.76.3.1 template<class T, class Alloc> reference Oscl_TagTree< T, Alloc >::iterator::operator * () const [inline]
- 6.76.3.2 template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::iterator::operator!= (const self & x) [inline]
- 6.76.3.3 template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::iterator::operator++ (int) [inline]
- 6.76.3.4 template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::iterator::operator++ () [inline]
- 6.76.3.5 template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::iterator::operator- (int) [inline]
- 6.76.3.6 template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::iterator::operator- () [inline]
- 6.76.3.7 template<class T, class Alloc> pointer Oscl_TagTree< T, Alloc >::iterator::operator → () const [inline]
- 6.76.3.8 template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::iterator::operator== (const self & x) [inline]

6.76.4 Field Documentation

- [oscl_tagtree.h](#)

6.77 Oscl_TagTree< T, Alloc >::Node Struct Reference

```
#include <oscl_tagtree.h>
```

Public Types

- `typedef Oscl_Vector< Node *, Alloc > children_type`

Public Methods

- `Node()`
- `void sort_children()`
- `tag_type::size_type depth()`

Data Fields

- `tag_type tag`
- `T value`
- `Node * parent`
- `children_type children`

template<class T, class Alloc> struct Oscl_TagTree< T, Alloc >::Node

6.77.1 Member Typedef Documentation

6.77.1.1 template<class T, class Alloc> typedef Oscl_Vector<Node*, Alloc> Oscl_TagTree< T, Alloc >::Node::children_type

6.77.2 Constructor & Destructor Documentation

6.77.2.1 template<class T, class Alloc> Oscl_TagTree< T, Alloc >::Node::Node () [inline]

6.77.3 Member Function Documentation

6.77.3.1 template<class T, class Alloc> tag_type::size_type Oscl_TagTree< T, Alloc >::Node::depth () [inline]

6.77.3.2 template<class T, class Alloc> void Oscl_TagTree< T, Alloc >::Node::sort_children () [inline]

6.77.4 Field Documentation

6.77.4.1 template<class T, class Alloc> children_type Oscl_TagTree< T, Alloc >::Node::children

6.77.4.2 template<class T, class Alloc> Node* Oscl_TagTree< T, Alloc >::Node::parent

6.77.4.3 template<class T, class Alloc> tag_type Oscl_TagTree< T, Alloc >::Node::tag

6.77.4.4 template<class T, class Alloc> T Oscl_TagTree< T, Alloc >::Node::value

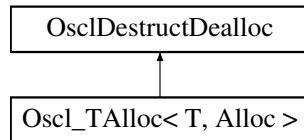
The documentation for this struct was generated from the following file:

- [oscl_tagtree.h](#)

6.78 Oscl_TAlloc< T, Alloc > Class Template Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oscl_TAlloc< T, Alloc >::



Public Types

- typedef T [value_type](#)
- typedef T * [pointer](#)
- typedef const T * [const_pointer](#)
- typedef uint32 [size_type](#)
- typedef T & [reference](#)
- typedef const T & [const_reference](#)

Public Methods

- virtual [~Oscl_TAlloc \(\)](#)
- [pointer allocate_fl \(uint32 size, const char *file_name, const int line_num\)](#)
- [pointer allocate \(uint32 size\)](#)
- [pointer alloc_and_construct_fl \(const_reference val, const char *file_name, const int line_num\)](#)
- [pointer alloc_and_construct \(const_reference val\)](#)
- void [deallocate \(OsclAny *p\)](#)
- void [deallocate \(OsclAny *p, size_type n\)](#)
- void [destruct_and_dealloc \(OsclAny *p\)](#)
- [pointer address \(reference r\)](#)
- [const_pointer address \(const_reference r\) const](#)
- void [construct \(pointer p, const_reference val\)](#)
- void [destroy \(pointer p\)](#)

```
template<class T, class Alloc> class Oscl_TAlloc< T, Alloc >
```

6.78.1 Member Typedef Documentation

6.78.1.1 `template<class T, class Alloc> typedef const T* Oscl_TAlloc< T, Alloc >::const_pointer`

6.78.1.2 `template<class T, class Alloc> typedef const T& Oscl_TAlloc< T, Alloc >::const_reference`

6.78.1.3 `template<class T, class Alloc> typedef T* Oscl_TAlloc< T, Alloc >::pointer`

6.78.1.4 `template<class T, class Alloc> typedef T& Oscl_TAlloc< T, Alloc >::reference`

6.78.1.5 `template<class T, class Alloc> typedef uint32 Oscl_TAlloc< T, Alloc >::size_type`

6.78.1.6 `template<class T, class Alloc> typedef T Oscl_TAlloc< T, Alloc >::value_type`

6.78.2 Constructor & Destructor Documentation

6.78.2.1 `template<class T, class Alloc> virtual Oscl_TAlloc< T, Alloc >::~Oscl_TAlloc () [inline, virtual]`

6.78.3 Member Function Documentation

6.78.3.1 `template<class T, class Alloc> const_pointer Oscl_TAlloc< T, Alloc >::address (const_reference r) const [inline]`

6.78.3.2 `template<class T, class Alloc> pointer Oscl_TAlloc< T, Alloc >::address (reference r) [inline]`

6.78.3.3 `template<class T, class Alloc> pointer Oscl_TAlloc< T, Alloc >::alloc_and_construct (const_reference val) [inline]`

6.78.3.4 `template<class T, class Alloc> pointer Oscl_TAlloc< T, Alloc >::alloc_and_construct_file (const_reference val, const char *file_name, const int line_num) [inline]`

6.78.3.5 `template<class T, class Alloc> pointer Oscl_TAlloc< T, Alloc >::allocate (uint32 size) [inline]`

6.78.3.6 `template<class T, class Alloc> pointer Oscl_TAlloc< T, Alloc >::allocate_file (uint32 size, const char *file_name, const int line_num) [inline]`

6.78.3.7 `template<class T, class Alloc> void Oscl_TAlloc< T, Alloc >::construct (pointer p, const_reference val) [inline]`

6.78.3.8 `template<class T, class Alloc> void Oscl_TAlloc< T, Alloc >::deallocate (OsclAny *p, size_type n) [inline]`

6.78.3.9 `template<class T, class Alloc> void Oscl_TAlloc< T, Alloc >::deallocate (OsclAny *p) [inline]`

6.78.3.10 `template<class T, class Alloc> void Oscl_TAlloc< T, Alloc >::destroy (pointer p) [inline]`

6.78.3.11 `template<class T, class Alloc> void Oscl_TAlloc< T, Alloc >::destruct_and_dealloc (OsclAny *p) [inline, virtual]`

The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

6.79 Oscl_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference

```
#include <oscl_defalloc.h>
```

Public Types

- `typedef Oscl_TAlloc< U, V > other`

```
template<class T, class Alloc>template<class U, class V> struct Oscl_TAlloc< T, Alloc >::rebind< U, V >
```

6.79.1 Member Typedef Documentation

6.79.1.1 `template<class T, class Alloc> template<class U, class V> typedef Oscl_TAlloc<U, V> Oscl_TAlloc< T, Alloc >::rebind< U, V >::other`

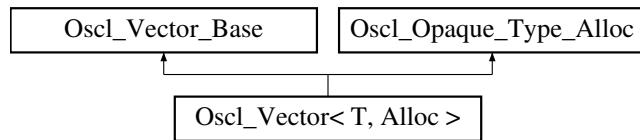
The documentation for this struct was generated from the following file:

- `oscl_defalloc.h`

6.80 Oscl_Vector< T, Alloc > Class Template Reference

```
#include <oscl_vector.h>
```

Inheritance diagram for Oscl_Vector< T, Alloc >::



Public Types

- `typedef T value_type`
- `typedef T * pointer`
- `typedef T & reference`
- `typedef const T & const_reference`
- `typedef T * iterator`
- `typedef const T * const_iterator`

Public Methods

- `Oscl_Vector()`
- `Oscl_Vector(uint32 n)`
- `Oscl_Vector(const Oscl_Vector< T, Alloc > &x)`
- `virtual ~Oscl_Vector()`
- `Oscl_Vector< T, Alloc > & operator=(const Oscl_Vector< T, Alloc > &x)`
- `void push_back(const T &x)`
- `void push_front(const T &x)`
- `iterator insert(iterator pos, const T &x)`
- `T & operator[](uint32 n)`
- `const T & operator[](uint32 n) const`
- `T & front()`
- `const T & front() const`
- `T & back()`
- `const T & back() const`
- `void pop_back()`
- `void clear()`
- `void destroy()`
- `iterator begin() const`
- `iterator end() const`
- `iterator erase(iterator pos)`
- `iterator erase(iterator first, iterator last)`

6.80.1 Detailed Description

template<class T, class Alloc> class Oscl_Vector< T, Alloc >

Oscl_Vector Class. A subset of STL::Vector methods. Oscl_Vector supports random access to elements, constant time insertion and removal of elements at the end of the vector, and linear time insertion and removal of elements at the beginning or middle of the vector. The number of elements in a vector can vary dynamically, and memory management is performed automatically.

6.80.2 Member Typedef Documentation

6.80.2.1 template<class T, class Alloc> typedef const T* Oscl_Vector< T, Alloc >::const_iterator

6.80.2.2 template<class T, class Alloc> typedef const T& Oscl_Vector< T, Alloc >::const_reference

6.80.2.3 template<class T, class Alloc> typedef T* Oscl_Vector< T, Alloc >::iterator

6.80.2.4 template<class T, class Alloc> typedef T* Oscl_Vector< T, Alloc >::pointer

6.80.2.5 template<class T, class Alloc> typedef T& Oscl_Vector< T, Alloc >::reference

6.80.2.6 template<class T, class Alloc> typedef T Oscl_Vector< T, Alloc >::value_type

6.80.3 Constructor & Destructor Documentation

6.80.3.1 template<class T, class Alloc> Oscl_Vector< T, Alloc >::Oscl_Vector () [inline]

Creates an empty vector.

6.80.3.2 template<class T, class Alloc> Oscl_Vector< T, Alloc >::Oscl_Vector (uint32 n) [inline]

Creates an empty vector with capacity n.

Parameters:

n creates a vector with n elements. The main reason for specifying n is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

6.80.3.3 template<class T, class Alloc> Oscl_Vector< T, Alloc >::Oscl_Vector (const Oscl_Vector< T, Alloc > & x) [inline]

Copy Constructor.

Parameters:

x vector class to copy.

**6.80.3.4 template<class T, class Alloc> virtual Oscl_Vector< T, Alloc >::~Oscl_Vector ()
[inline, virtual]**

The destructor.

6.80.4 Member Function Documentation

**6.80.4.1 template<class T, class Alloc> const T& Oscl_Vector< T, Alloc >::back () const
[inline]**

Returns the last element.

6.80.4.2 template<class T, class Alloc> T& Oscl_Vector< T, Alloc >::back () [inline]

Returns the last element.

**6.80.4.3 template<class T, class Alloc> iterator Oscl_Vector< T, Alloc >::begin () const
[inline]**

Returns an iterator pointing to the beginning of the vector.

Reimplemented from [Oscl_Vector_Base](#).

6.80.4.4 template<class T, class Alloc> void Oscl_Vector< T, Alloc >::clear () [inline]

Removes all elements.

6.80.4.5 template<class T, class Alloc> void Oscl_Vector< T, Alloc >::destroy () [inline]

Destroy – this is like an explicit destructor call.

Reimplemented from [Oscl_Vector_Base](#).

**6.80.4.6 template<class T, class Alloc> iterator Oscl_Vector< T, Alloc >::end () const
[inline]**

Returns an iterator pointing to the end of the vector..

Reimplemented from [Oscl_Vector_Base](#).

**6.80.4.7 template<class T, class Alloc> iterator Oscl_Vector< T, Alloc >::erase (iterator *first*,
iterator *last*) [inline]**

Erases elements in range [*first*, *last*). Erasing an element invalidates all iterators pointing to elements following the deletion point.

Parameters:

first starting position

last ending position, this position is not erased

6.80.4.8 template<class T, class Alloc> iterator Oscl_Vector< T, Alloc >::erase (iterator pos) [inline]

Erases the element pointed to by iterator pos. Erasing an element invalidates all iterators pointing to elements following the deletion point.

Parameters:

pos iterator at erase position

6.80.4.9 template<class T, class Alloc> const T& Oscl_Vector< T, Alloc >::front () const [inline]

Returns the first element.

6.80.4.10 template<class T, class Alloc> T& Oscl_Vector< T, Alloc >::front () [inline]

Returns the first element.

6.80.4.11 template<class T, class Alloc> iterator Oscl_Vector< T, Alloc >::insert (iterator pos, const T & x) [inline]

Inserts a new element before the one at pos.

Parameters:

pos position at which to insert the new element.

x new element

6.80.4.12 template<class T, class Alloc> Oscl_Vector<T, Alloc>& Oscl_Vector< T, Alloc >::operator= (const Oscl_Vector< T, Alloc > & x) [inline]

The assignment operator

6.80.4.13]

template<class T, class Alloc> const T& Oscl_Vector< T, Alloc >::operator[] (uint32 n) const [inline]

Returns the n'th element.

Parameters:

n element position to return

6.80.4.14]

template<class T, class Alloc> T& Oscl_Vector< T, Alloc >::operator[] (uint32 n) [inline]

Returns the n'th element.

Parameters:

n element position to return

6.80.4.15 template<class T, class Alloc> void Oscl_Vector< T, Alloc >::pop_back () [inline]

Removes the last element.

Reimplemented from [Oscl_Vector_Base](#).

6.80.4.16 template<class T, class Alloc> void Oscl_Vector< T, Alloc >::push_back (const T & x) [inline]

Inserts a new element at the end. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

Parameters:

x new element

6.80.4.17 template<class T, class Alloc> void Oscl_Vector< T, Alloc >::push_front (const T & x) [inline]

Inserts a new element at the front. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

Parameters:

x new element

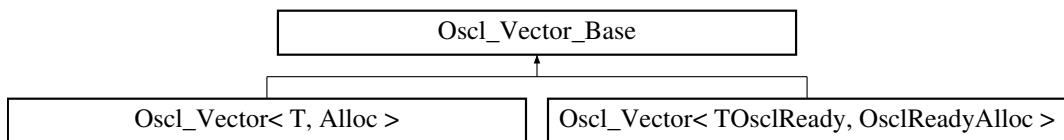
The documentation for this class was generated from the following file:

- [oscl_vector.h](#)

6.81 Oscl_Vector_Base Class Reference

```
#include <oscl_vector.h>
```

Inheritance diagram for Oscl_Vector_Base::



Public Methods

- uint32 [size \(\) const](#)
- uint32 [capacity \(\) const](#)
- bool [empty \(\) const](#)
- OSCL_IMPORT_REF void [reserve \(uint32 n\)](#)

Protected Methods

- OSCL_IMPORT_REF void [construct \(Oscl_Opaque_Type_Alloc *aType\)](#)
- OSCL_IMPORT_REF void [construct \(Oscl_Opaque_Type_Alloc *aType, uint32 n\)](#)
- OSCL_IMPORT_REF void [construct \(Oscl_Opaque_Type_Alloc *aType, const Oscl_Vector_Base &x\)](#)
- virtual [~Oscl_Vector_Base \(\)](#)
- OSCL_IMPORT_REF void [push_back \(const OsclAny *x\)](#)
- OSCL_IMPORT_REF void [pop_back \(\)](#)
- OSCL_IMPORT_REF void [push_front \(const OsclAny *x\)](#)
- OSCL_IMPORT_REF OsclAny * [insert \(OsclAny *pos, const OsclAny *x\)](#)
- OSCL_IMPORT_REF OsclAny * [erase \(OsclAny *pos\)](#)
- OSCL_IMPORT_REF OsclAny * [erase \(OsclAny *first, OsclAny *last\)](#)
- OSCL_IMPORT_REF void [assign_vector \(const Oscl_Vector_Base &x\)](#)
- OSCL_IMPORT_REF void [destroy \(\)](#)

Protected Attributes

- uint32 [numelems](#)
- uint32 [bufsize](#)
- [OsclAny * elems](#)
- uint32 [sizeof_T](#)

Friends

- class [OsclPriorityQueueBase](#)

6.81.1 Detailed Description

Oscl_Vector_Base is a non-templatized base class for [Oscl_Vector](#). The purpose of this base class is to avoid large inline routines in the [Oscl_Vector](#) implementation. This class is not intended for direct instantiation except by [Oscl_Vector](#).

6.81.2 Constructor & Destructor Documentation

6.81.2.1 `virtual Oscl_Vector_Base::~Oscl_Vector_Base () [inline, protected, virtual]`

The destructor.

6.81.3 Member Function Documentation

6.81.3.1 `OSCL_IMPORT_REF void Oscl_Vector_Base::assign_vector (const Oscl_Vector_Base & x) [protected]`

6.81.3.2 `uint32 Oscl_Vector_Base::capacity () const [inline]`

Returns the allocated memory of the vector in units of number of elements.

6.81.3.3 `OSCL_IMPORT_REF void Oscl_Vector_Base::construct (Oscl_Opaque_Type_Alloc * aType, const Oscl_Vector_Base & x) [protected]`

6.81.3.4 `OSCL_IMPORT_REF void Oscl_Vector_Base::construct (Oscl_Opaque_Type_Alloc * aType, uint32 n) [protected]`

6.81.3.5 `OSCL_IMPORT_REF void Oscl_Vector_Base::construct (Oscl_Opaque_Type_Alloc * aType) [protected]`

6.81.3.6 `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< OsclSocketRequest *, OsclMemAllocator >](#), [Oscl_Vector< entry_type *, Alloc >](#), [Oscl_Vector< OSCL_HeapString< OsclMemAllocator >, OsclMemAllocator >](#), [Oscl_Vector< OsclAsyncFileBuffer *, OsclMemAllocator >](#), [Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator >](#), [Oscl_Vector< OsclSharedPtr< PVLoggerFilter >, alloc_type >](#), [Oscl_Vector< TOsclReady, OsclReadyAlloc >](#), [Oscl_Vector< OsclSharedPtr< PVLoggerAppender >, alloc_type >](#), [Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >](#), and [Oscl_Vector< OsclAny *, OsclMemAllocator >](#).

6.81.3.7 `bool Oscl_Vector_Base::empty () const [inline]`

True if the vector's size is 0.

6.81.3.8 OSCL_IMPORT_REF `OsclAny*` Oscl_Vector_Base::erase (`OsclAny *first, OsclAny *last`) [protected]

Erases elements in range [first, last). Erasing an element invalidates all iterators pointing to elements following the deletion point.

Parameters:

first starting position

last ending position, this position is not erased

6.81.3.9 OSCL_IMPORT_REF `OsclAny*` Oscl_Vector_Base::erase (`OsclAny *pos`) [protected]

Erases the element pointed to by iterator pos. Erasing an element invalidates all iterators pointing to elements following the deletion point.

Parameters:

pos iterator at erase position

6.81.3.10 OSCL_IMPORT_REF `OsclAny*` Oscl_Vector_Base::insert (`OsclAny *pos, const OsclAny *x`) [protected]

Inserts a new element at a specific position.

Parameters:

pos iterator at insert position.

x pointer to new element

6.81.3.11 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< OsclSocketRequest *, OsclMemAllocator >`, `Oscl_Vector< entry_type *, Alloc >`, `Oscl_Vector< OSCL_HeapString< OsclMemAllocator >, OsclMemAllocator >`, `Oscl_Vector< OsclAsyncFileBuffer *, OsclMemAllocator >`, `Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator >`, `Oscl_Vector< OsclSharedPtr< PVLoggerFilter >, alloc_type >`, `Oscl_Vector< TOsclReady, OsclReadyAlloc >`, `Oscl_Vector< OsclSharedPtr< PVLoggerAppender >, alloc_type >`, `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >`, and `Oscl_Vector< OsclAny *, OsclMemAllocator >`.

6.81.3.12 OSCL_IMPORT_REF void Oscl_Vector_Base::push_back (const `OsclAny *x`) [protected]

Inserts a new element at the end. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

Parameters:

x pointer to the new element

**6.81.3.13 OSCL_IMPORT_REF void Oscl_Vector_Base::push_front (const OsclAny * *x*)
[protected]**

Inserts a new element at the front. Inserting an element invalidates all iterators if memory reallocation occurs as a result of the insertion.

Parameters:

x pointer to new element

6.81.3.14 OSCL_IMPORT_REF void Oscl_Vector_Base::reserve (uint32 *n*)

Reallocates memory if necessary to a capacity of *n* elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

Parameters:

n size of vector

6.81.3.15 uint32 Oscl_Vector_Base::size () const [inline]

Returns the size of the vector in units of number of elements.

6.81.4 Friends And Related Function Documentation

6.81.4.1 friend class OsclPriorityQueueBase [friend]

6.81.5 Field Documentation

6.81.5.1 uint32 Oscl_Vector_Base::bufsize [protected]**6.81.5.2 OsclAny* Oscl_Vector_Base::elems [protected]****6.81.5.3 uint32 Oscl_Vector_Base::numelems [protected]****6.81.5.4 uint32 Oscl_Vector_Base::sizeof_T [protected]**

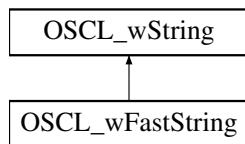
The documentation for this class was generated from the following file:

- [oscl_vector.h](#)

6.82 OSCL_wFastString Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wFastString::



Public Types

- `typedef OSCL_wString::chartype chartype`

Public Methods

- `OSCL_IMPORT_REF OSCL_wFastString()`
- `OSCL_IMPORT_REF OSCL_wFastString(const OSCL_wFastString &src)`
- `OSCL_IMPORT_REF OSCL_wFastString(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_wFastString(chartype *buf, uint32 maxlen)`
- `OSCL_IMPORT_REF ~OSCL_wFastString()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_wFastString & operator=(const OSCL_wFastString &src)`
- `OSCL_IMPORT_REF OSCL_wFastString & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(chartype *cstr, uint32 maxlen)`
- `OSCL_IMPORT_REF void set_length()`

Friends

- class `OSCL_wString`

6.82.1 Detailed Description

OSCL_wFastString is identical to `OSCL_FastString` except that it uses wide-character format. For descriptions, see `OSCL_FastString`.

6.82.2 Member Typedef Documentation

6.82.2.1 `typedef OSCL_wString::chartype OSCL_wFastString::chartype`

Reimplemented from `OSCL_wString`.

6.82.3 Constructor & Destructor Documentation

- 6.82.3.1 **OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString ()**
- 6.82.3.2 **OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString (const OSCL_wFastString & src)**
- 6.82.3.3 **OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString (const chartype * cstr)**
- 6.82.3.4 **OSCL_IMPORT_REF OSCL_wFastString::OSCL_wFastString (chartype * buf, uint32 maxlen)**
- 6.82.3.5 **OSCL_IMPORT_REF OSCL_wFastString::~OSCL_wFastString ()**

6.82.4 Member Function Documentation

- 6.82.4.1 **OSCL_IMPORT_REF const chartype* OSCL_wFastString::get_cstr () [virtual]**

Implements [OSCL_wString](#).

- 6.82.4.2 **OSCL_IMPORT_REF uint32 OSCL_wFastString::get_maxsize () [virtual]**

Implements [OSCL_wString](#).

- 6.82.4.3 **OSCL_IMPORT_REF uint32 OSCL_wFastString::get_size () [virtual]**

Implements [OSCL_wString](#).

- 6.82.4.4 **OSCL_IMPORT_REF chartype* OSCL_wFastString::get_str () [virtual]**

Implements [OSCL_wString](#).

- 6.82.4.5 **OSCL_IMPORT_REF OSCL_wFastString& OSCL_wFastString::operator= (const chartype * cstr)**

Reimplemented from [OSCL_wString](#).

- 6.82.4.6 **OSCL_IMPORT_REF OSCL_wFastString& OSCL_wFastString::operator= (const OSCL_wFastString & src)**

- 6.82.4.7 **OSCL_IMPORT_REF void OSCL_wFastString::set (chartype * cstr, uint32 maxlen)**

- 6.82.4.8 **OSCL_IMPORT_REF void OSCL_wFastString::set_length ()**

6.82.5 Friends And Related Function Documentation

- 6.82.5.1 **friend class OSCL_wString [friend]**

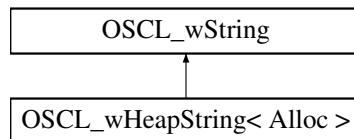
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

6.83 OSCL_wHeapString< Alloc > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wHeapString< Alloc >::



Public Types

- `typedef OSCL_wString::chartype chartype`

Public Methods

- `OSCL_wHeapString()`
- `OSCL_wHeapString(const OSCL_wHeapString &src)`
- `OSCL_wHeapString(const OSCL_wString &src)`
- `OSCL_wHeapString(const chartype *cstr)`
- `OSCL_wHeapString(const chartype *buf, uint32 length)`
- `~OSCL_wHeapString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype * get_cstr() const`
- `chartype * get_str() const`
- `OSCL_wHeapString & operator=(const OSCL_wHeapString &src)`
- `OSCL_wHeapString & operator=(const OSCL_wString &src)`
- `OSCL_wHeapString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`

Friends

- class `OSCL_wString`

6.83.1 Detailed Description

```
template<class Alloc> class OSCL_wHeapString< Alloc >
```

`OSCL_wHeapString` is identical to `OSCL_HeapString` except that it uses wide-character format. For descriptions, see `OSCL_HeapString`.

6.83.2 Member Typedef Documentation

6.83.2.1 template<class Alloc> typedef OSCL_wString::chartype OSCL_wHeapString< Alloc >::chartype

Reimplemented from `OSCL_wString`.

6.83.3 Friends And Related Function Documentation

6.83.3.1 template<class Alloc> friend class OSCL_wString [friend]

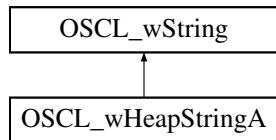
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

6.84 OSCL_wHeapStringA Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wHeapStringA::



Public Types

- `typedef OSCL_wString::chartype chartype`

Public Methods

- `OSCL_IMPORT_REF OSCL_wHeapStringA()`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wHeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wHeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wString &src, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const chartype *cstr, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const chartype *buf, uint32 length, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF ~OSCL_wHeapStringA()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const OSCL_wHeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(const chartype *buf, uint32 length)`

Friends

- class `OSCL_wString`

6.84.1 Detailed Description

OSCL_wHeapStringA is identical to `OSCL_HeapStringA` except that it uses wide-character format. For descriptions, see `OSCL_HeapStringA`.

6.84.2 Member Typedef Documentation

6.84.2.1 `typedef OSCL_wString::chartype OSCL_wHeapStringA::chartype`

Reimplemented from [OSCL_wString](#).

6.84.3 Constructor & Destructor Documentation

6.84.3.1 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA()`

6.84.3.2 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (Oscl_DefAlloc * alloc, OsclRefCounter *ref = NULL)`

6.84.3.3 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL_wHeapStringA &src)`

6.84.3.4 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL_wHeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref = NULL)`

6.84.3.5 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL_wString &src, Oscl_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

6.84.3.6 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const chartype * cstr, Oscl_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

6.84.3.7 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const chartype * buf, uint32 length, Oscl_DefAlloc *alloc = NULL, OsclRefCounter *ref = NULL)`

6.84.3.8 `OSCL_IMPORT_REF OSCL_wHeapStringA::~OSCL_wHeapStringA()`

6.84.4 Member Function Documentation

6.84.4.1 `OSCL_IMPORT_REF const chartype* OSCL_wHeapStringA::get_cstr () [virtual]`

Implements [OSCL_wString](#).

6.84.4.2 `OSCL_IMPORT_REF uint32 OSCL_wHeapStringA::get_maxsize () [virtual]`

Implements [OSCL_wString](#).

6.84.4.3 `OSCL_IMPORT_REF uint32 OSCL_wHeapStringA::get_size () [virtual]`

Implements [OSCL_wString](#).

6.84.4.4 `OSCL_IMPORT_REF chartype* OSCL_wHeapStringA::get_str () [virtual]`

Implements [OSCL_wString](#).

6.84.4.5 OSCL_IMPORT_REF OSCL_wHeapStringA& OSCL_wHeapStringA::operator= (const chartype * *cstr*)

Reimplemented from [OSCL_wString](#).

6.84.4.6 OSCL_IMPORT_REF OSCL_wHeapStringA& OSCL_wHeapStringA::operator= (const OSCL_wString & *src*)

Reimplemented from [OSCL_wString](#).

6.84.4.7 OSCL_IMPORT_REF OSCL_wHeapStringA& OSCL_wHeapStringA::operator= (const OSCL_wHeapStringA & *src*)

6.84.4.8 OSCL_IMPORT_REF void OSCL_wHeapStringA::set (const chartype * *buf*, uint32 *length*)

6.84.5 Friends And Related Function Documentation

6.84.5.1 friend class OSCL_wString [friend]

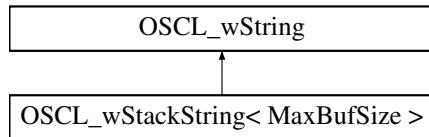
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

6.85 OSCL_wStackString< MaxBufSize > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL_wStackString< MaxBufSize >::



Public Types

- `typedef OSCL_wString::chartype chartype`

Public Methods

- `OSCL_wStackString()`
- `OSCL_wStackString(const OSCL_wStackString &src)`
- `OSCL_wStackString(const OSCL_wString &src)`
- `OSCL_wStackString(const chartype *cstr)`
- `OSCL_wStackString(const chartype *buf, uint32 length)`
- `~OSCL_wStackString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype * get_cstr() const`
- `chartype * get_str() const`
- `OSCL_wStackString & operator=(const OSCL_wStackString &src)`
- `OSCL_wStackString & operator=(const OSCL_wString &src)`
- `OSCL_wStackString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`

Friends

- class `OSCL_wString`

6.85.1 Detailed Description

`template<uint32 MaxBufSize> class OSCL_wStackString< MaxBufSize >`

`OSCL_wStackString` is identical to `OSCL_StackString` except that it uses wide-character format. For descriptions, see `OSCL_StackString`.

6.85.2 Member Typedef Documentation

6.85.2.1 `template<uint32 MaxBufSize> typedef OSCL_wString::chartype OSCL_wStackString<MaxBufSize >::chartype`

Reimplemented from [OSCL_wString](#).

6.85.3 Friends And Related Function Documentation

6.85.3.1 `template<uint32 MaxBufSize> friend class OSCL_wString [friend]`

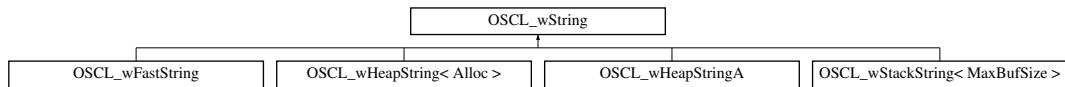
The documentation for this class was generated from the following file:

- [oscl_string_containers.h](#)

6.86 OSCL_wString Class Reference

```
#include <oscl_string.h>
```

Inheritance diagram for OSCL_wString::



Public Types

- `typedef oscl_wchar chartype`

Public Methods

- `virtual uint32 get_size () const=0`
- `virtual uint32 get_maxsize () const=0`
- `virtual const chartype * get_cstr () const=0`
- `virtual OSCL_IMPORT_REF bool is_writable () const`
- `virtual chartype * get_str () const=0`
- `OSCL_IMPORT_REF OSCL_wString & operator=(const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wString & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_wString & operator+=(const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wString & operator+=(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_wString & operator+=(const chartype c)`
- `OSCL_IMPORT_REF bool operator==(const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator!=(const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator< (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator<= (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator> (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator>= (const OSCL_wString &src) const`
- `OSCL_IMPORT_REF bool operator==(const chartype *cstr) const`
- `OSCL_IMPORT_REF chartype operator[] (uint32 index) const`
- `virtual OSCL_IMPORT_REF chartype read (uint32 index) const`
- `virtual OSCL_IMPORT_REF int8 hash () const`
- `virtual OSCL_IMPORT_REF void write (uint32 index, chartype c)`
- `virtual OSCL_IMPORT_REF void write (uint32 offset, uint32 length, const chartype *buf)`

Protected Methods

- `OSCL_IMPORT_REF OSCL_wString ()`
- `virtual OSCL_IMPORT_REF ~OSCL_wString ()`
- `virtual void set_rep (const chartype *cstr)=0`
- `virtual void append_rep (const chartype *cstr)=0`
- `virtual void set_rep (const OSCL_wString &src)=0`
- `virtual void append_rep (const OSCL_wString &src)=0`
- `virtual void set_len (uint32 len)=0`

6.86.1 Detailed Description

A common base class for string classes with wide character (oscl_wchar) format. OSCL_wString and [OSCL_String](#) are identical except for the character format. For descriptions, see [OSCL_String](#).

6.86.2 Member Typedef Documentation

6.86.2.1 `typedef oscl_wchar OSCL_wString::chartype`

Reimplemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

6.86.3 Constructor & Destructor Documentation

6.86.3.1 `OSCL_IMPORT_REF OSCL_wString::OSCL_wString () [protected]`

6.86.3.2 `virtual OSCL_IMPORT_REF OSCL_wString::~OSCL_wString () [protected, virtual]`

6.86.4 Member Function Documentation

6.86.4.1 `virtual void OSCL_wString::append_rep (const OSCL_wString & src) [protected, pure virtual]`

6.86.4.2 `virtual void OSCL_wString::append_rep (const chartype * cstr) [protected, pure virtual]`

6.86.4.3 `virtual const chartype* OSCL_wString::get_cstr () [pure virtual]`

Implemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

6.86.4.4 `virtual uint32 OSCL_wString::get_maxsize () [pure virtual]`

Implemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

6.86.4.5 `virtual uint32 OSCL_wString::get_size () [pure virtual]`

Implemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

6.86.4.6 `virtual chartype* OSCL_wString::get_str () [pure virtual]`

Implemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

- 6.86.4.7 **virtual OSCL_IMPORT_REF int8 OSCL_wString::hash () [virtual]**
- 6.86.4.8 **virtual OSCL_IMPORT_REF bool OSCL_wString::is_writable () [virtual]**
- 6.86.4.9 **OSCL_IMPORT_REF bool OSCL_wString::operator!= (const OSCL_wString & src) const**
- 6.86.4.10 **OSCL_IMPORT_REF OSCL_wString& OSCL_wString::operator+= (const chartype c)**
- 6.86.4.11 **OSCL_IMPORT_REF OSCL_wString& OSCL_wString::operator+= (const chartype * cstr)**
- 6.86.4.12 **OSCL_IMPORT_REF OSCL_wString& OSCL_wString::operator+= (const OSCL_wString & src)**
- 6.86.4.13 **OSCL_IMPORT_REF bool OSCL_wString::operator< (const OSCL_wString & src) const**
- 6.86.4.14 **OSCL_IMPORT_REF bool OSCL_wString::operator<= (const OSCL_wString & src) const**
- 6.86.4.15 **OSCL_IMPORT_REF OSCL_wString& OSCL_wString::operator= (const chartype * cstr)**

Reimplemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), [OSCL_wStackString< MaxBufSize >](#), and [OSCL_wFastString](#).

- 6.86.4.16 **OSCL_IMPORT_REF OSCL_wString& OSCL_wString::operator= (const OSCL_wString & src)**

Reimplemented in [OSCL_wHeapString< Alloc >](#), [OSCL_wHeapStringA](#), and [OSCL_wStackString< MaxBufSize >](#).

- 6.86.4.17 **OSCL_IMPORT_REF bool OSCL_wString::operator== (const chartype * cstr) const**
- 6.86.4.18 **OSCL_IMPORT_REF bool OSCL_wString::operator== (const OSCL_wString & src) const**
- 6.86.4.19 **OSCL_IMPORT_REF bool OSCL_wString::operator> (const OSCL_wString & src) const**
- 6.86.4.20 **OSCL_IMPORT_REF bool OSCL_wString::operator>= (const OSCL_wString & src) const**
- 6.86.4.21 **]**

OSCL_IMPORT_REF chartype OSCL_wString::operator[] (uint32 index) const

- 6.86.4.22 **virtual OSCL_IMPORT_REF chartype OSCL_wString::read (uint32 *index*) const [virtual]**
- 6.86.4.23 **virtual void OSCL_wString::set_len (uint32 *len*) [protected, pure virtual]**
- 6.86.4.24 **virtual void OSCL_wString::set_rep (const OSCL_wString & *src*) [protected, pure virtual]**
- 6.86.4.25 **virtual void OSCL_wString::set_rep (const chartype * *cstr*) [protected, pure virtual]**
- 6.86.4.26 **virtual OSCL_IMPORT_REF void OSCL_wString::write (uint32 *offset*, uint32 *length*, const chartype * *buf*) [virtual]**
- 6.86.4.27 **virtual OSCL_IMPORT_REF void OSCL_wString::write (uint32 *index*, chartype *c*) [virtual]**

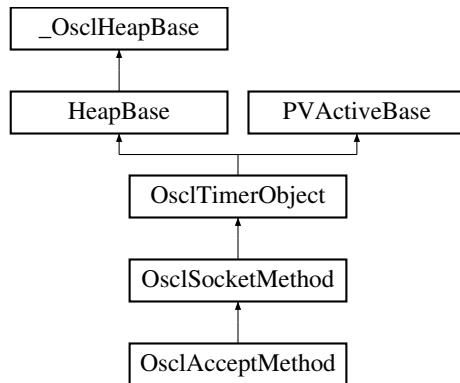
The documentation for this class was generated from the following file:

- [oscl_string.h](#)

6.87 OsclAcceptMethod Class Reference

```
#include <oscl_socket_accept.h>
```

Inheritance diagram for OsclAcceptMethod::



Public Methods

- [~OsclAcceptMethod \(\)](#)
- [TPVSocketEvent Accept \(int32 aTimeout\)](#)
- [void DiscardAcceptedSocket \(\)](#)
- [OsclSocketI * GetAcceptedSocket \(\)](#)
- [OsclAcceptRequest * AcceptRequest \(\)](#)

Static Public Methods

- [OsclAcceptMethod * NewL \(OsclIPSocketI &c\)](#)

6.87.1 Constructor & Destructor Documentation

6.87.1.1 OsclAcceptMethod::~OsclAcceptMethod ()

6.87.2 Member Function Documentation

6.87.2.1 TPVSocketEvent OsclAcceptMethod::Accept (int32 aTimeout)

6.87.2.2 OsclAcceptRequest* OsclAcceptMethod::AcceptRequest () [inline]

6.87.2.3 void OsclAcceptMethod::DiscardAcceptedSocket ()

6.87.2.4 OsclSocketI* OsclAcceptMethod::GetAcceptedSocket ()

6.87.2.5 OsclAcceptMethod* OsclAcceptMethod::NewL (OsclIPSocketI &c) [static]

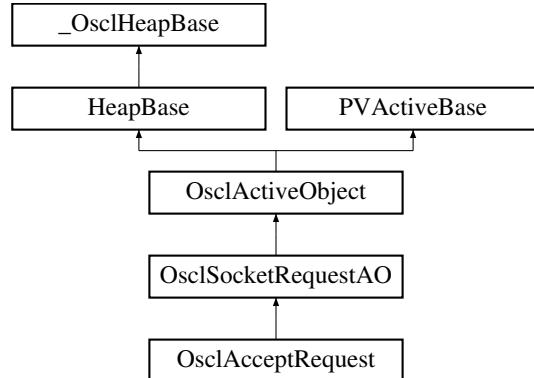
The documentation for this class was generated from the following file:

- [oscl_socket_accept.h](#)

6.88 OsclAcceptRequest Class Reference

```
#include <oscl_socket_accept.h>
```

Inheritance diagram for OsclAcceptRequest::



Public Methods

- [OsclAcceptRequest \(OsclSocketMethod &c\)](#)
- [void Accept \(OsclSocketI &aSocket\)](#)

6.88.1 Constructor & Destructor Documentation

6.88.1.1 OsclAcceptRequest::OsclAcceptRequest ([OsclSocketMethod & c](#)) [inline]

6.88.2 Member Function Documentation

6.88.2.1 void OsclAcceptRequest::Accept ([OsclSocketI & aSocket](#))

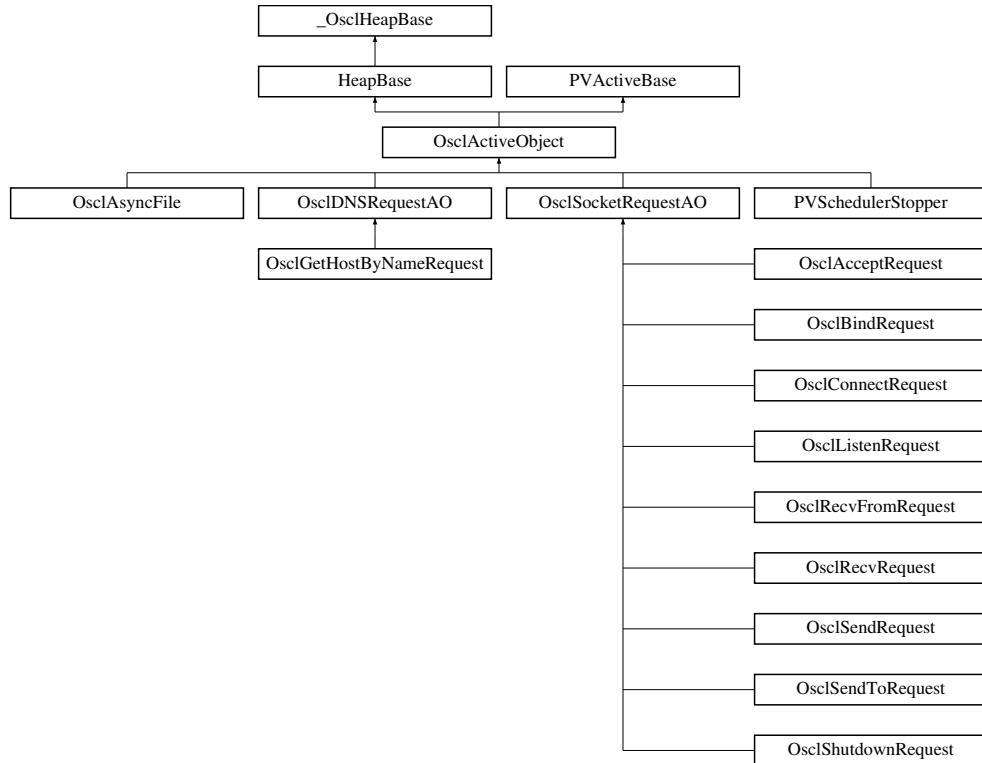
The documentation for this class was generated from the following file:

- [oscl_socket_accept.h](#)

6.89 OsclActiveObject Class Reference

```
#include <oscl_scheduler_ao.h>
```

Inheritance diagram for OsclActiveObject::



Public Types

- enum **OsclActivePriority** { **EPriorityIdle** = -100, **EPriorityLow** = -20, **EPriorityNominal** = 0, **EPriorityHigh** = 10, **EPriorityHighest** = 20 }

Public Methods

- OSCL_IMPORT_REF **OsclActiveObject** (int32 aPriority, const char name[])
- virtual OSCL_IMPORT_REF ~**OsclActiveObject** ()
- OSCL_IMPORT_REF void **SetBusy** ()
- OSCL_IMPORT_REF bool **IsBusy** () const
- OSCL_IMPORT_REF void **PendForExec** ()
- OSCL_IMPORT_REF void **PendComplete** (int32 aStatus)
- OSCL_IMPORT_REF void **AddToScheduler** ()
- OSCL_IMPORT_REF void **RemoveFromScheduler** ()
- OSCL_IMPORT_REF void **RunIfNotReady** ()
- OSCL_IMPORT_REF void **Cancel** ()
- OSCL_IMPORT_REF int32 **Priority** () const
- OSCL_IMPORT_REF int32 **Status** () const
- OSCL_IMPORT_REF void **SetStatus** (int32)
- OSCL_IMPORT_REF **OsclAOStatus & StatusRef** ()

Protected Methods

- virtual OSCL_IMPORT_REF void [DoCancel \(\)](#)
- virtual OSCL_IMPORT_REF int32 [RunError \(int32 aError\)](#)

6.89.1 Detailed Description

User base class for execution objects. OsclActiveObject defines an execution object without any timer. This AO can be used across threads, i.e. the request can be activated in one thread and completed in another.

6.89.2 Member Enumeration Documentation

6.89.2.1 enum OsclActiveObject::OsclActivePriority

Scheduling priorities.

Enumeration values:

- EPriorityIdle** A low priority, useful for execution objects representing background processing.
EPriorityLow A priority higher than EPriorityIdle but lower than EPriorityNominal.
EPriorityNominal Most exec objects will have this priority.
EPriorityHigh A priority higher than EPriorityNominal; useful for execution objects handling user input.
EPriorityHighest A priority higher than EPriorityHighest.

6.89.3 Constructor & Destructor Documentation

6.89.3.1 OSCL_IMPORT_REF OsclActiveObject::OsclActiveObject (int32 *aPriority*, const char *name*[])

Constructor.

Parameters:

- aPriority* (input param): scheduling priority
name (input param): optional name for this AO.

6.89.3.2 virtual OSCL_IMPORT_REF OsclActiveObject::~OsclActiveObject () [virtual]

Destructor.

6.89.4 Member Function Documentation

6.89.4.1 OSCL_IMPORT_REF void OsclActiveObject::AddToScheduler ()

Add this exec object to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

6.89.4.2 OSCL_IMPORT_REF void OsclActiveObject::Cancel ()

Cancel any pending request. If the request is readied, this will call the DoCancel routine, wait for the request to cancel, then set the request idle. The AO will not run. If the request is not readied, it does nothing. Request must be canceled from the same thread in which it is scheduled.

Reimplemented from [PVActiveBase](#).

6.89.4.3 virtual OSCL_IMPORT_REF void OsclActiveObject::DoCancel () [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Implements [PVActiveBase](#).

Reimplemented in [OsclDNSRequestAO](#), and [OsclSocketRequestAO](#).

6.89.4.4 OSCL_IMPORT_REF bool OsclActiveObject::IsBusy ()

Return true if this AO is pending, false otherwise.

6.89.4.5 OSCL_IMPORT_REF void OsclActiveObject::PendComplete (int32 *aStatus*)

Complete the active request for the AO. This API is thread-safe. If the request is not pending, this call will leave.

Parameters:

aStatus: request completion status.

6.89.4.6 OSCL_IMPORT_REF void OsclActiveObject::PendForExec ()

Set request active for this AO and set the status to pending. PendForExec is identical to SetActive, but it additionally sets the request status to OSCL_REQUEST_PENDING.

6.89.4.7 OSCL_IMPORT_REF int32 OsclActiveObject::Priority ()

Return scheduling priority of this exec object.

6.89.4.8 OSCL_IMPORT_REF void OsclActiveObject::RemoveFromScheduler ()

Remove this AO from its scheduler. Will leave if the calling thread context does not match the scheduling thread. Cancels any readied request before removing.

Reimplemented from [PVActiveBase](#).

**6.89.4.9 virtual OSCL_IMPORT_REF int32 OsclActiveObject::RunError (int32 *aError*)
[protected, virtual]**

Run Error handler. This gets called by scheduler when the Run routine leaves. The default implementation simply returns the leave code. If the derived class wants to handle errors from Run, it may override this. The RunError should return OsclErrNone if it handles the error, otherwise it should return the input error code.

Parameters:

aError: the leave code generated by the Run.

Implements [PVActiveBase](#).

6.89.4.10 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 acced to any scheduler, or if the calling thread context does not match the scheduling thread.

6.89.4.11 OSCL_IMPORT_REF void OsclActiveObject::SetBusy ()

Set object ready for this AO, additionally sets the request status to OSCL_REQUEST_PENDING. Will leave if the request is already readied, or the execution object is not added to any scheduler, or the calling thread context does not match the scheduler thread.

6.89.4.12 OSCL_IMPORT_REF void OsclActiveObject::SetStatus (int32)**6.89.4.13 OSCL_IMPORT_REF int32 OsclActiveObject::Status ()**

Request status access

6.89.4.14 OSCL_IMPORT_REF OsclAOStatus& OsclActiveObject::StatusRef ()

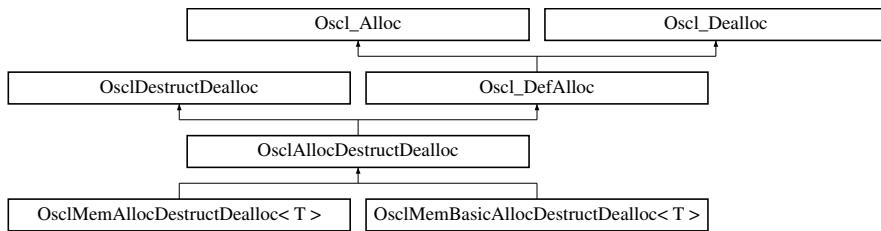
The documentation for this class was generated from the following file:

- [oscl_scheduler_ao.h](#)

6.90 OsclAllocDestructDealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for OsclAllocDestructDealloc::



Public Methods

- virtual ~OsclAllocDestructDealloc ()

6.90.1 Constructor & Destructor Documentation

6.90.1.1 virtual OsclAllocDestructDealloc::~OsclAllocDestructDealloc () [inline, virtual]

The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

6.91 OsclAOStatus Class Reference

```
#include <oscl_aostatus.h>
```

Public Methods

- OSCL_INLINE OsclAOStatus ()
- OSCL_INLINE OsclAOStatus (int32 aStatus)
- OSCL_INLINE int32 operator= (int32 aStatus)
- OSCL_INLINE int32 operator== (int32 aStatus) const
- OSCL_INLINE int32 operator!= (int32 aStatus) const
- OSCL_INLINE int32 operator>= (int32 aStatus) const
- OSCL_INLINE int32 operator<= (int32 aStatus) const
- OSCL_INLINE int32 operator> (int32 aStatus) const
- OSCL_INLINE int32 operator< (int32 aStatus) const
- OSCL_INLINE int32 Value () const

6.91.1 Constructor & Destructor Documentation

6.91.1.1 OSCL_INLINE OsclAOStatus::OsclAOStatus ()

6.91.1.2 OSCL_INLINE OsclAOStatus::OsclAOStatus (int32 *aStatus*)

6.91.2 Member Function Documentation

6.91.2.1 OSCL_INLINE int32 OsclAOStatus::operator!= (int32 *aStatus*) const

6.91.2.2 OSCL_INLINE int32 OsclAOStatus::operator< (int32 *aStatus*) const

6.91.2.3 OSCL_INLINE int32 OsclAOStatus::operator<= (int32 *aStatus*) const

6.91.2.4 OSCL_INLINE int32 OsclAOStatus::operator= (int32 *aStatus*)

6.91.2.5 OSCL_INLINE int32 OsclAOStatus::operator== (int32 *aStatus*) const

6.91.2.6 OSCL_INLINE int32 OsclAOStatus::operator> (int32 *aStatus*) const

6.91.2.7 OSCL_INLINE int32 OsclAOStatus::operator>= (int32 *aStatus*) const

6.91.2.8 OSCL_INLINE int32 OsclAOStatus::Value ()

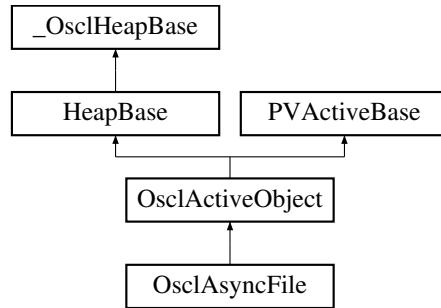
The documentation for this class was generated from the following file:

- [oscl_aostatus.h](#)

6.92 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 \(int32 offset, Oscl_File::seek_type origin\)`](#)
- [`int32 Tell \(\)`](#)
- [`uint32 Read \(OsclAny *aBuffer1, uint32 aDataSize, uint32 aNumElements\)`](#)
- [`int32 EndOfFile \(\)`](#)
- [`int32 Size \(\)`](#)
- [`int32 Close \(\)`](#)
- [`uint32 Write \(const OsclAny *aBuffer1, uint32 aDataSize, uint32 aNumElements\)`](#)
- [`uint32 Flush \(\)`](#)

Static Public Methods

- [`OsclAsyncFile * NewL \(OsclNativeFile &aAsyncFile, int32 aCacheSize, PVLogger *\)`](#)
- [`void Delete \(OsclAsyncFile *\)`](#)

Data Fields

- [`uint32 iNumOfRun`](#)
- [`uint32 iNumOfRunErr`](#)

6.92.1 Detailed Description

OsclAsyncFile

6.92.2 Constructor & Destructor Documentation

6.92.2.1 OsclAsyncFile::~OsclAsyncFile ()

Destructor.

6.92.3 Member Function Documentation

6.92.3.1 int32 OsclAsyncFile::Close ()

6.92.3.2 void OsclAsyncFile::Delete (OsclAsyncFile *) [static]

6.92.3.3 int32 OsclAsyncFile::EndOfFile ()

6.92.3.4 uint32 OsclAsyncFile::Flush () [inline]

6.92.3.5 OsclAsyncFile* OsclAsyncFile::NewL (OsclNativeFile & aSyncFile, int32 aCacheSize, PVLogger *) [static]

Two-phased constructor.

Parameters:

aSyncFile: open handle for async file read. Note: it is the caller's job to open/close this file handle.

aSyncFile: duplicate open handle for sync file read. Note: it is the caller's job to open this file handle, but this class will close the handle.

aCacheSize: size of one of the individual cache buffers. The total cached data size will be larger, since multiple buffers are used.

aStartAsyncRead: When true, async file read will start immediately. When false, read will not begin until StartAsyncRead is called.

- 6.92.3.6 int32 OsclAsyncFile::Open (const char **filename*, uint32 *mode*, const OsclNativeFileParams & *params*, Oscl_FileServer & *fileserv*)
- 6.92.3.7 int32 OsclAsyncFile::Open (const oscl_wchar **filename*, uint32 *mode*, const OsclNativeFileParams & *params*, Oscl_FileServer & *fileserv*)
- 6.92.3.8 uint32 OsclAsyncFile::Read (OsclAny **aBuffer1*, uint32 *aDataSize*, uint32 *aNumElements*)
- 6.92.3.9 int32 OsclAsyncFile::Seek (int32 *offset*, Oscl_File::seek_type *origin*)
- 6.92.3.10 int32 OsclAsyncFile::Size ()
- 6.92.3.11 int32 OsclAsyncFile::Tell ()
- 6.92.3.12 uint32 OsclAsyncFile::Write (const OsclAny **aBuffer1*, uint32 *aDataSize*, uint32 *aNumElements*) [inline]

6.92.4 Field Documentation

- 6.92.4.1 uint32 OsclAsyncFile::iNumOfRun

- 6.92.4.2 uint32 OsclAsyncFile::iNumOfRunErr

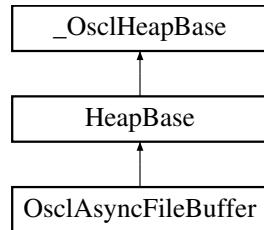
The documentation for this class was generated from the following file:

- [oscl_file_async_read.h](#)

6.93 OsclAsyncFileBuffer Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclAsyncFileBuffer::



Public Methods

- [`~OsclAsyncFileBuffer \(\)`](#)
- [`void CleanInUse \(\)`](#)
- [`void SetInUse \(\)`](#)
- [`bool IsInUse \(\)`](#)
- [`bool IsValid \(\)`](#)
- [`int32 Offset \(\)`](#)
- [`void SetOffset \(int32 aOffset\)`](#)
- [`int32 Length \(\)`](#)
- [`bool HasThisOffset \(int32 aOffset\)`](#)
- [`int32 Id \(\)`](#)
- [`OsclBuf * Buffer \(\)`](#)
- [`void UpdateData \(\)`](#)
- [`void StartAsyncRead \(bool aStartAsyncRead\)`](#)

Static Public Methods

- [`OsclAsyncFileBuffer * NewL \(int32 aBufferSize, int32 aId\)`](#)

6.93.1 Detailed Description

Buffer class used with async read. We keep an array of these, covering consecutive areas of the file. This allows for some seeking without requiring a full flush & refill each time.

6.93.2 Constructor & Destructor Documentation

6.93.2.1 `OsclAsyncFileBuffer::~OsclAsyncFileBuffer ()`

6.93.3 Member Function Documentation

6.93.3.1 `OsclBuf* OsclAsyncFileBuffer::Buffer ()`

6.93.3.2 `void OsclAsyncFileBuffer::CleanInUse () [inline]`

6.93.3.3 `bool OsclAsyncFileBuffer::HasThisOffset (int32 aOffset)`

6.93.3.4 `int32 OsclAsyncFileBuffer::Id () [inline]`

6.93.3.5 `bool OsclAsyncFileBuffer::IsInUse () [inline]`

6.93.3.6 `bool OsclAsyncFileBuffer::IsValid () [inline]`

6.93.3.7 `int32 OsclAsyncFileBuffer::Length () [inline]`

6.93.3.8 `OsclAsyncFileBuffer* OsclAsyncFileBuffer::NewL (int32 aBufferSize, int32 aId) [static]`

6.93.3.9 `int32 OsclAsyncFileBuffer::Offset () [inline]`

6.93.3.10 `void OsclAsyncFileBuffer::SetInUse () [inline]`

6.93.3.11 `void OsclAsyncFileBuffer::SetOffset (int32 aOffset) [inline]`

6.93.3.12 `void OsclAsyncFileBuffer::StartAsyncRead (bool aStartAsyncRead)`

6.93.3.13 `void OsclAsyncFileBuffer::UpdateData ()`

The documentation for this class was generated from the following file:

- `oscl_file_async_read.h`

6.94 OsclAuditCB Class Reference

```
#include <oscl_mem.h>
```

Public Methods

- [OsclAuditCB \(\)](#)
- [OsclAuditCB \(const OsclMemStatsNode *myStatsNode, OsclMemAudit *ptr\)](#)

Data Fields

- [const OsclMemStatsNode * pStatsNode](#)
- [OsclMemAudit * pAudit](#)

6.94.1 Constructor & Destructor Documentation

6.94.1.1 OsclAuditCB::OsclAuditCB () [inline]

6.94.1.2 OsclAuditCB::OsclAuditCB (const OsclMemStatsNode * *myStatsNode*, OsclMemAudit * *ptr*) [inline]

6.94.2 Field Documentation

6.94.2.1 OsclMemAudit* OsclAuditCB::pAudit

6.94.2.2 const OsclMemStatsNode* OsclAuditCB::pStatsNode

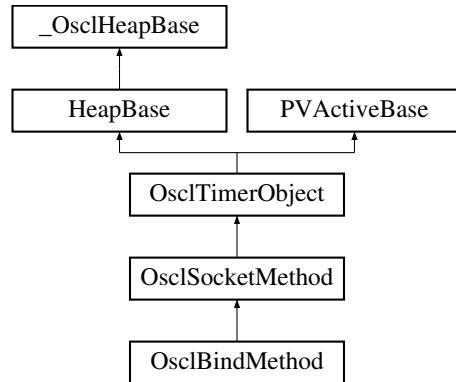
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

6.95 OsclBindMethod Class Reference

```
#include <oscl_socket_bind.h>
```

Inheritance diagram for OsclBindMethod::



Public Methods

- [~OsclBindMethod \(\)](#)
- [TPVSocketEvent Bind \(OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsclBindRequest * BindRequest \(\)](#)

Static Public Methods

- [OsclBindMethod * NewL \(OsclIPSocketI &c\)](#)

6.95.1 Constructor & Destructor Documentation

6.95.1.1 OsclBindMethod::~OsclBindMethod ()

6.95.2 Member Function Documentation

6.95.2.1 TPVSocketEvent OsclBindMethod::Bind (OsclNetworkAddress & aAddress, int32 aTimeout)

6.95.2.2 OsclBindRequest* OsclBindMethod::BindRequest () [inline]

6.95.2.3 OsclBindMethod* OsclBindMethod::NewL (OsclIPSocketI &c) [static]

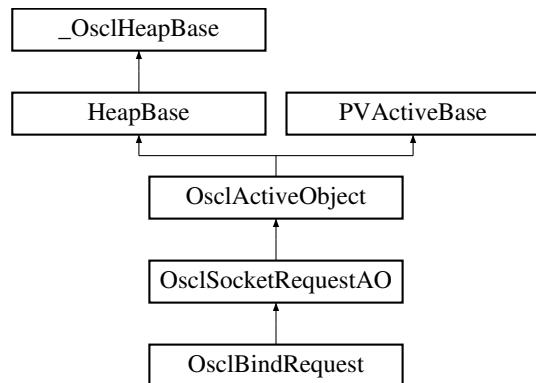
The documentation for this class was generated from the following file:

- [oscl_socket_bind.h](#)

6.96 OsclBindRequest Class Reference

```
#include <oscl_socket_bind.h>
```

Inheritance diagram for OsclBindRequest::



Public Methods

- [OsclBindRequest \(OsclSocketMethod &c\)](#)
- [void Bind \(OsclNetworkAddress &aAddress\)](#)

6.96.1 Detailed Description

This is the AO that interacts with the socket server

6.96.2 Constructor & Destructor Documentation

6.96.2.1 OsclBindRequest::OsclBindRequest ([OsclSocketMethod & c](#)) [inline]

6.96.3 Member Function Documentation

6.96.3.1 void OsclBindRequest::Bind ([OsclNetworkAddress & aAddress](#))

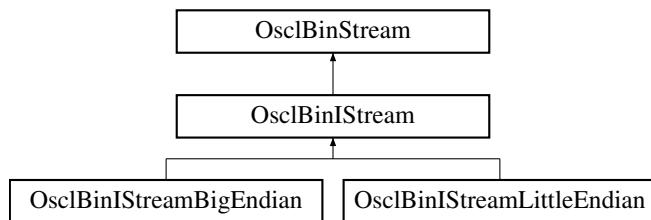
The documentation for this class was generated from the following file:

- [oscl_socket_bind.h](#)

6.97 OsclBinIStream Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStream::



Public Methods

- [OsclBinIStream \(\)](#)
- [~OsclBinIStream \(\)](#)
- uint8 [Read_uint8 \(\)](#)

This method reads an unsigned short from the stream.

- OsclBinIStream & [get \(int8 *data, int32 size\)](#)

This method reads 'length' number of bytes from the stream and places them in 'data'.

6.97.1 Constructor & Destructor Documentation

6.97.1.1 OsclBinIStream::OsclBinIStream () [inline]

6.97.1.2 OsclBinIStream::~OsclBinIStream () [inline]

6.97.2 Member Function Documentation

6.97.2.1 OsclBinIStream& OsclBinIStream::get (int8 * data, int32 size)

This method reads 'length' number of bytes from the stream and places them in 'data'.

Parameters:

data is a pointer to the place to store the bytes read

size is the number of bytes to read

6.97.2.2 uint8 OsclBinIStream::Read_uint8 ()

This method reads an unsigned short from the stream.

Returns:

Unsigned short read from the stream.

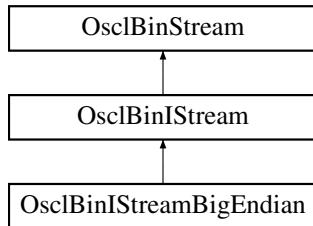
The documentation for this class was generated from the following file:

- [oscl_bin_stream.h](#)

6.98 OsclBinIStreamBigEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStreamBigEndian::



Public Methods

- [OsclBinIStreamBigEndian \(\)](#)
This method reads a int8 from the stream and stores it in 'data'.
- [void Read \(int8 &data\)](#)
This method reads a uint8 from the stream and stores it in 'data'.
- [void Read \(uint8 &data\)](#)
This method reads a int16 from the stream and stores it in 'data'.
- [void Read \(int16 &data\)](#)
This method reads a uint16 from the stream and stores it in 'data'.
- [void Read \(uint16 &data\)](#)
This method reads a int32 from the stream and stores it in 'data'.
- [void Read \(int32 &data\)](#)
This method reads a uint32 from the stream and stores it in 'data'.
- [OsclBinIStream & operator>> \(int8 &data\)](#)
This method reads a uint16 from the stream and stores it in 'data'.
- [OsclBinIStream & operator>> \(uint8 &data\)](#)
This method reads a int16 from the stream and stores it in 'data'.
- [OsclBinIStream & operator>> \(int16 &data\)](#)
This method reads a uint16 from the stream and stores it in 'data'.
- [OsclBinIStream & operator>> \(uint16 &data\)](#)
This method reads a int32 from the stream and stores it in 'data'.
- [OsclBinIStream & operator>> \(int32 &data\)](#)
This method reads a uint32 from the stream and stores it in 'data'.
- [OsclBinIStream & operator>> \(uint32 &data\)](#)
This method reads a uint32 from the stream and stores it in 'data'.
- [uint16 Read_uint16 \(\)](#)
This method reads an unsigned short from the stream.
- [uint32 Read_uint32 \(\)](#)
This method reads an unsigned long from the stream.

6.98.1 Constructor & Destructor Documentation

6.98.1.1 OsclBinIStreamBigEndian::OsclBinIStreamBigEndian () [inline]

6.98.2 Member Function Documentation

6.98.2.1 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (uint32 & data)

This method reads a uint32 from the stream and stores it in 'data'.

6.98.2.2 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int32 & data)

This method reads a int32 from the stream and stores it in 'data'.

6.98.2.3 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (uint16 & data)

This method reads a uint16 from the stream and stores it in 'data'.

6.98.2.4 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int16 & data)

This method reads a int16 from the stream and stores it in 'data'.

6.98.2.5 OsclBinIStream& OsclBinIStreamBigEndian::operator>> (uint8 & data)

This method reads a uint8 from the stream and stores it in 'data'.

6.98.2.6 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int8 & data)

This method reads a int8 from the stream and stores it in 'data'.

6.98.2.7 void OsclBinIStreamBigEndian::Read (uint32 & data)

6.98.2.8 void OsclBinIStreamBigEndian::Read (int32 & data)

6.98.2.9 void OsclBinIStreamBigEndian::Read (uint16 & data)

6.98.2.10 void OsclBinIStreamBigEndian::Read (int16 & data)

6.98.2.11 void OsclBinIStreamBigEndian::Read (uint8 & data)

6.98.2.12 void OsclBinIStreamBigEndian::Read (int8 & data)

6.98.2.13 uint16 OsclBinIStreamBigEndian::Read_uint16 ()

This method reads an unsigned short from the stream.

Returns:

Unsigned short read from the stream.

6.98.2.14 uint32 OsclBinIStreamBigEndian::Read_uint32 ()

This method reads an unsigned long from the stream.

Returns:

unsigned long read from the stream.

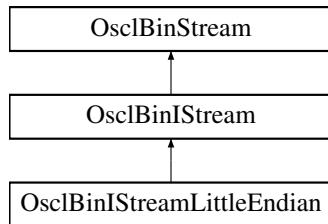
The documentation for this class was generated from the following file:

- [oscl_bin_stream.h](#)

6.99 OsclBinIStreamLittleEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStreamLittleEndian::



Public Methods

- [OsclBinIStreamLittleEndian \(\)](#)
- [OsclBinIStreamLittleEndian & operator>> \(int8 &data\)](#)
This method reads a int8 from the stream and stores it in 'data'.
- [OsclBinIStreamLittleEndian & operator>> \(uint8 &data\)](#)
This method reads a uint8 from the stream and stores it in 'data'.
- [OsclBinIStreamLittleEndian & operator>> \(int16 &data\)](#)
This method reads a int16 from the stream and stores it in 'data'.
- [OsclBinIStreamLittleEndian & operator>> \(uint16 &data\)](#)
This method reads a uint16 from the stream and stores it in 'data'.
- [OsclBinIStreamLittleEndian & operator>> \(int32 &data\)](#)
This method reads a int32 from the stream and stores it in 'data'.
- [OsclBinIStreamLittleEndian & operator>> \(uint32 &data\)](#)
This method reads a uint32 from the stream and stores it in 'data'.

Protected Methods

- uint16 [Read_uint16 \(\)](#)
- uint32 [Read_uint32 \(\)](#)

6.99.1 Constructor & Destructor Documentation

6.99.1.1 OsclBinIStreamLittleEndian::OsclBinIStreamLittleEndian () [inline]

6.99.2 Member Function Documentation

6.99.2.1 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint32 & data)

This method reads a uint32 from the stream and stores it in 'data'.

6.99.2.2 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int32 & data)

This method reads a int32 from the stream and stores it in 'data'.

6.99.2.3 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint16 & data)

This method reads a uint16 from the stream and stores it in 'data'.

6.99.2.4 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int16 & data)

This method reads a int16 from the stream and stores it in 'data'.

6.99.2.5 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint8 & data)

This method reads a uint8 from the stream and stores it in 'data'.

6.99.2.6 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int8 & data)

This method reads a int8 from the stream and stores it in 'data'.

6.99.2.7 uint16 OsclBinIStreamLittleEndian::Read_uint16 () [protected]

6.99.2.8 uint32 OsclBinIStreamLittleEndian::Read_uint32 () [protected]

The documentation for this class was generated from the following file:

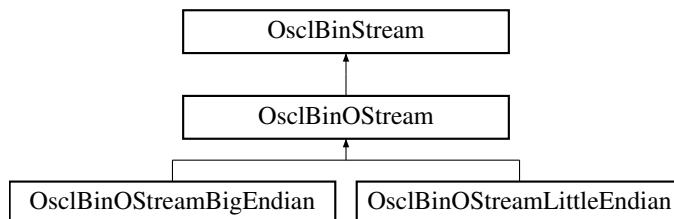
- [oscl_bin_stream.h](#)

6.100 OsclBinOStream Class Reference

Class OsclBinOStream implements the basic stream functions for an output stream.

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStream:::



Public Methods

- [OsclBinOStream \(\)](#)
- virtual [~OsclBinOStream \(\)](#)
- [OsclBinOStream & write \(const int8 *data, int32 size\)](#)

This method writes 'length' number of bytes stored in 'data' to the stream.

6.100.1 Detailed Description

Class OsclBinOStream implements the basic stream functions for an output stream.

6.100.2 Constructor & Destructor Documentation

6.100.2.1 [OsclBinOStream::OsclBinOStream \(\) \[inline\]](#)

6.100.2.2 [virtual OsclBinOStream::~OsclBinOStream \(\) \[inline, virtual\]](#)

6.100.3 Member Function Documentation

6.100.3.1 [OsclBinOStream& OsclBinOStream::write \(const int8 * data, int32 size\)](#)

This method writes 'length' number of bytes stored in 'data' to the stream.

The documentation for this class was generated from the following file:

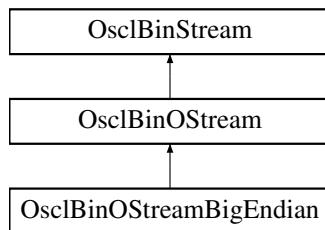
- [oscl_bin_stream.h](#)

6.101 OsclBinOStreamBigEndian Class Reference

Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering.

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStreamBigEndian::



Public Methods

- [OsclBinOStreamBigEndian \(\)](#)
- [OsclBinOStreamBigEndian & operator<< \(const int8 &data\)](#)
This method writes a int8 from 'data' to the stream.
- [OsclBinOStreamBigEndian & operator<< \(const uint8 &data\)](#)
This method writes a uint8 from 'data' to the stream.
- [OsclBinOStreamBigEndian & operator<< \(const int16 &data\)](#)
This method writes a int16 from 'data' to the stream.
- [OsclBinOStreamBigEndian & operator<< \(const uint16 &data\)](#)
This method writes a uint16 from 'data' to the stream.
- [OsclBinOStreamBigEndian & operator<< \(const int32 &data\)](#)
This method writes a int32 from 'data' to the stream.
- [OsclBinOStreamBigEndian & operator<< \(const uint32 &data\)](#)
This method writes a uint32 from 'data' to the stream.

Protected Methods

- [void WriteUnsignedShort \(const uint16 data\)](#)
- [void WriteUnsignedLong \(const uint32 data\)](#)

6.101.1 Detailed Description

Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering.

6.101.2 Constructor & Destructor Documentation

6.101.2.1 OsclBinOStreamBigEndian::OsclBinOStreamBigEndian () [inline]

6.101.3 Member Function Documentation

6.101.3.1 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint32 & data)

This method writes a uint32 from 'data' to the stream.

6.101.3.2 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int32 & data)

This method writes a int32 from 'data' to the stream.

6.101.3.3 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint16 & data)

This method writes a uint16 from 'data' to the stream.

6.101.3.4 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int16 & data)

This method writes a int16 from 'data' to the stream.

6.101.3.5 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint8 & data)

This method writes a uint8 from 'data' to the stream.

6.101.3.6 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int8 & data)

This method writes a int8 from 'data' to the stream.

6.101.3.7 void OsclBinOStreamBigEndian::WriteUnsignedLong (const uint32 data) [protected]

6.101.3.8 void OsclBinOStreamBigEndian::WriteUnsignedShort (const uint16 data) [protected]

The documentation for this class was generated from the following file:

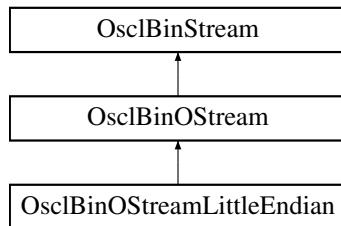
- [oscl_bin_stream.h](#)

6.102 OsclBinOStreamLittleEndian Class Reference

Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStreamLittleEndian::



Public Methods

- [OsclBinOStreamLittleEndian \(\)](#)
- [OsclBinOStreamLittleEndian & operator<< \(const int8 &data\)](#)
This method writes a int8 from 'data' to the stream.
- [OsclBinOStreamLittleEndian & operator<< \(const uint8 &data\)](#)
This method writes a uint8 from 'data' to the stream.
- [OsclBinOStreamLittleEndian & operator<< \(const int16 &data\)](#)
This method writes a int16 from 'data' to the stream.
- [OsclBinOStreamLittleEndian & operator<< \(const uint16 &data\)](#)
This method writes a uint16 from 'data' to the stream.
- [OsclBinOStreamLittleEndian & operator<< \(const int32 &data\)](#)
This method writes a int32 from 'data' to the stream.
- [OsclBinOStreamLittleEndian & operator<< \(const uint32 &data\)](#)
This method writes a uint32 from 'data' to the stream.

Protected Methods

- [void WriteUnsignedShort \(const uint16 data\)](#)
This method writes 'data' (unsigned short) to the stream.
- [void WriteUnsignedLong \(const uint32 data\)](#)
This method writes 'data' (unsigned long) to the stream.

6.102.1 Detailed Description

Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.

6.102.2 Constructor & Destructor Documentation

6.102.2.1 OsclBinOStreamLittleEndian::OsclBinOStreamLittleEndian () [inline]

6.102.3 Member Function Documentation

6.102.3.1 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint32 & data)

This method writes a uint32 from 'data' to the stream.

6.102.3.2 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int32 & data)

This method writes a int32 from 'data' to the stream.

6.102.3.3 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint16 & data)

This method writes a uint16 from 'data' to the stream.

6.102.3.4 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int16 & data)

This method writes a int16 from 'data' to the stream.

6.102.3.5 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint8 & data)

This method writes a uint8 from 'data' to the stream.

6.102.3.6 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int8 & data)

This method writes a int8 from 'data' to the stream.

6.102.3.7 void OsclBinOStreamLittleEndian::WriteUnsignedLong (const uint32 data) [protected]

This method writes 'data' (unsigned long) to the stream.

6.102.3.8 void OsclBinOStreamLittleEndian::WriteUnsignedShort (const uint16 data) [protected]

This method writes 'data' (unsigned short) to the stream.

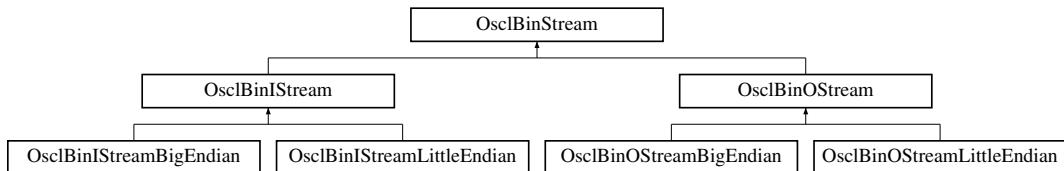
The documentation for this class was generated from the following file:

- [oscl_bin_stream.h](#)

6.103 OsclBinStream Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinStream::



Public Methods

- [OsclBinStream \(\)](#)
- [bool good \(\)](#)

This method determines if the stream is ok.

- [bool eof \(\)](#)

This method determines if end of stream has been reached.

- [bool fail \(\)](#)

This method determines if an error has occurred in the stream.

- [void Attach \(void *buffer, uint32 l_length\)](#)

This method specifies the data buffer to attach to the stream.

- [void Attach \(const uint32 numFragments, const OsclMemoryFragment *fragPtr\)](#)

This method specifies the memory fragment array to use for input.

- [uint32 tellg \(\)](#)

This method returns the current stream position.

- [void Seek \(uint32 absPosition\)](#)

This method seeks to the specified stream position.

- [uint32 PositionInBlock \(\)](#)

This method returns the current stream position.

- [void seekFromCurrentPosition \(int32 offset\)](#)

This method seeks to the specified offset from the current location.

Protected Types

- enum [state_t](#) { [GOOD_STATE](#), [EOF_STATE](#), [FAIL_STATE](#) }

Protected Methods

- bool [ReserveSpace](#) (uint32 size)
- bool [HaveRoomInCurrentBlock](#) (uint32 size)

Protected Attributes

- [state_t state](#)
- uint8 * [pBasePosition](#)
- uint8 * [pPosition](#)
- uint32 [length](#)
- const [OsclMemoryFragment](#) * [nextFragPtr](#)
- int [fragsLeft](#)
- const [OsclMemoryFragment](#) * [firstFragPtr](#)
- int [numFrags](#)
- [OsclMemoryFragment](#) [specialFragBuffer](#)

6.103.1 Member Enumeration Documentation

6.103.1.1 enum OsclBinStream::state_t [protected]

Enumeration values:

- GOOD_STATE**
- EOF_STATE**
- FAIL_STATE**

6.103.2 Constructor & Destructor Documentation

6.103.2.1 OsclBinStream::OsclBinStream () [inline]

6.103.3 Member Function Documentation

6.103.3.1 void OsclBinStream::Attach (const uint32 *numFragments*, const [OsclMemoryFragment](#) **fragPtr*)

This method specifies the memory fragment array to use for input.

This array should remain static while the stream refers to it.

Parameters:

numFragments is the number of elements in the array

fragPtr is the pointer to the MemoryFragment array

6.103.3.2 void OsclBinStream::Attach (void **buffer*, uint32 *l_length*)

This methods specifies the data buffer to attach to the stream.

Parameters:

buffer will provide the input

length of the buffer

6.103.3.3 bool OsclBinStream::eof ()

This method determines if end of stream has been reached.

Returns:

true if end of stream has been reached.

6.103.3.4 bool OsclBinStream::fail ()

This method determines if an error has occurred in the stream.

Returns:

true if an error occurred in the stream.

6.103.3.5 bool OsclBinStream::good ()

This method determines if the stream is ok.

Returns:

true if stream is ok.

6.103.3.6 bool OsclBinStream::HaveRoomInCurrentBlock (uint32 *size*) [protected]**6.103.3.7 uint32 OsclBinStream::PositionInBlock ()**

This method returns the current stream position.

Returns:

stream position.

6.103.3.8 bool OsclBinStream::ReserveSpace (uint32 *size*) [protected]**6.103.3.9 void OsclBinStream::Seek (uint32 *absPosition*)**

This method seeks to the specified stream position.

Returns:

Stream position.

6.103.3.10 void OsclBinStream::seekFromCurrentPosition (int32 *offset*)

This method seeks to the specified offset from the current location.

Parameters:

offset from current stream location

6.103.3.11 uint32 OsclBinStream::tellg ()

This method returns the current stream position.

This method is to be used if the input stream is a pointer to the MemoryFragment array

Returns:

Stream position.

6.103.4 Field Documentation

6.103.4.1 const OsclMemoryFragment* OsclBinStream::firstFragPtr [protected]

6.103.4.2 int OsclBinStream::fragsLeft [protected]

6.103.4.3 uint32 OsclBinStream::length [protected]

6.103.4.4 const OsclMemoryFragment* OsclBinStream::nextFragPtr [protected]

6.103.4.5 int OsclBinStream::numFrags [protected]

6.103.4.6 uint8* OsclBinStream::pBasePosition [protected]

6.103.4.7 uint8* OsclBinStream::pPosition [protected]

6.103.4.8 OsclMemoryFragment OsclBinStream::specialFragBuffer [protected]

6.103.4.9 state_t OsclBinStream::state [protected]

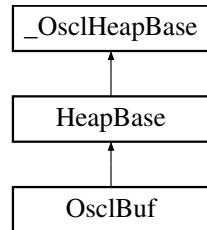
The documentation for this class was generated from the following file:

- [oscl_bin_stream.h](#)

6.104 OsclBuf Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclBuf::



Public Methods

- [OsclBuf](#) (int32 size)
- int32 [Length](#) ()
- [OsclPtr Des](#) ()
- [OsclPtrC DesC](#) ()

Static Public Methods

- OsclBuf * [NewL](#) (int32 size)
- void [Delete](#) (OsclBuf *a)

Data Fields

- uint8 * [iBuffer](#)
- int32 [iMaxLength](#)
- int32 [iLength](#)

6.104.1 Constructor & Destructor Documentation

6.104.1.1 **OsclBuf::OsclBuf (int32 *size*)** [inline]

6.104.2 Member Function Documentation

6.104.2.1 **void OsclBuf::Delete (OsclBuf * *a*)** [inline, static]

6.104.2.2 **OsclPtr OsclBuf::Des ()** [inline]

6.104.2.3 **OsclPtrC OsclBuf::DesC ()** [inline]

6.104.2.4 **int32 OsclBuf::Length ()** [inline]

6.104.2.5 **OsclBuf* OsclBuf::NewL (int32 *size*)** [inline, static]

6.104.3 Field Documentation

6.104.3.1 **uint8* OsclBuf::iBuffer**

6.104.3.2 **int32 OsclBuf::iLength**

6.104.3.3 **int32 OsclBuf::iMaxLength**

The documentation for this class was generated from the following file:

- [oscl_file_async_read.h](#)

6.105 OsclCompareLess< T > Class Template Reference

```
#include <oscl_priqueue.h>
```

Public Methods

- int [compare](#) (T &a, T &b) const

```
template<class T> class OsclCompareLess< T >
```

6.105.1 Member Function Documentation

**6.105.1.1 template<class T> int OsclCompareLess< T >::compare (T & a, T & b) const
[inline]**

The documentation for this class was generated from the following file:

- [oscl_priqueue.h](#)

6.106 OsclComponentRegistry Class Reference

```
#include <oscl_registry_serv_impl.h>
```

Public Methods

- [OsclComponentRegistry \(\)](#)
- [~OsclComponentRegistry \(\)](#)
- [int32 Register \(uint32 &aId, OSCL_String &, OsclComponentFactory\)](#)
- [int32 Unregister \(OSCL_String &\)](#)
- [int32 Unregister \(uint32\)](#)
- [OsclComponentFactory FindExact \(OSCL_String &\)](#)
- [void FindHierarchical \(OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &\)](#)
- [void OpenSession \(\)](#)
- [void CloseSession \(\)](#)

Data Fields

- [OsclComponentRegistryData iData](#)
- [OsclMutex iMutex](#)
- [uint32 iComponentIdCounter](#)
- [uint32 iNumSessions](#)

6.106.1 Detailed Description

Thread-safe singleton registry object.

6.106.2 Constructor & Destructor Documentation

6.106.2.1 OsclComponentRegistry::OsclComponentRegistry ()

6.106.2.2 OsclComponentRegistry::~OsclComponentRegistry ()

6.106.3 Member Function Documentation

6.106.3.1 void OsclComponentRegistry::CloseSession ()

6.106.3.2 **OsclComponentFactory** OsclComponentRegistry::FindExact (**OSCL_String** &)

6.106.3.3 void OsclComponentRegistry::FindHierarchical (**OSCL_String** &, **Oscl_Vector**<
OsclRegistryAccessElement, **OsclMemAllocator** > &)

6.106.3.4 void OsclComponentRegistry::OpenSession ()

6.106.3.5 int32 OsclComponentRegistry::Register (uint32 & *aId*, **OSCL_String** &,
OsclComponentFactory)

6.106.3.6 int32 OsclComponentRegistry::Unregister (uint32)

6.106.3.7 int32 OsclComponentRegistry::Unregister (**OSCL_String** &)

6.106.4 Field Documentation

6.106.4.1 uint32 OsclComponentRegistry::iComponentIdCounter

6.106.4.2 **OsclComponentRegistryData** OsclComponentRegistry::iData

6.106.4.3 **OsclMutex** OsclComponentRegistry::iMutex

6.106.4.4 uint32 OsclComponentRegistry::iNumSessions

The documentation for this class was generated from the following file:

- [oscl_registry_serv_impl.h](#)

6.107 OsclComponentRegistryData Class Reference

```
#include <oscl_registry_serv_impl.h>
```

Public Methods

- [OsclComponentRegistryElement * Find \(OSCL_String &, bool aExact\)](#)

Data Fields

- [Oscl_Vector< OsclComponentRegistryElement, OsclMemAllocator > iVec](#)

6.107.1 Detailed Description

Registry

6.107.2 Member Function Documentation

6.107.2.1 [OsclComponentRegistryElement* OsclComponentRegistryData::Find \(OSCL_String &, bool aExact\)](#)

6.107.3 Field Documentation

6.107.3.1 [Oscl_Vector<OsclComponentRegistryElement, OsclMemAllocator> OsclComponentRegistryData::iVec](#)

The documentation for this class was generated from the following file:

- [oscl_registry_serv_impl.h](#)

6.108 OsclComponentRegistryElement Class Reference

```
#include <oscl_registry_serv_impl.h>
```

Public Methods

- [OsclComponentRegistryElement \(OSCL_String &, OsclComponentFactory\)](#)
- [OsclComponentRegistryElement \(const OsclComponentRegistryElement &\)](#)
- [OsclComponentRegistryElement & operator= \(const OsclComponentRegistryElement &src\)](#)
- [~OsclComponentRegistryElement \(\)](#)
- [bool Match \(OSCL_String &aStr, bool aExact\)](#)

Data Fields

- [OSCL_String * iId](#)
- [OsclComponentFactory iFactory](#)
- [uint32 iComponentId](#)

6.108.1 Detailed Description

Data for each registered component.

6.108.2 Constructor & Destructor Documentation

6.108.2.1 OsclComponentRegistryElement::OsclComponentRegistryElement (OSCL_String &, OsclComponentFactory)

6.108.2.2 OsclComponentRegistryElement::OsclComponentRegistryElement (const OsclComponentRegistryElement &)

6.108.2.3 OsclComponentRegistryElement::~OsclComponentRegistryElement ()

6.108.3 Member Function Documentation

6.108.3.1 bool OsclComponentRegistryElement::Match (OSCL_String & aStr, bool aExact)

6.108.3.2 OsclComponentRegistryElement& OsclComponentRegistryElement::operator= (const OsclComponentRegistryElement & src)

6.108.4 Field Documentation

6.108.4.1 uint32 OsclComponentRegistryElement::iComponentId

6.108.4.2 OsclComponentFactory OsclComponentRegistryElement::iFactory

6.108.4.3 OSCL_String* OsclComponentRegistryElement::iId

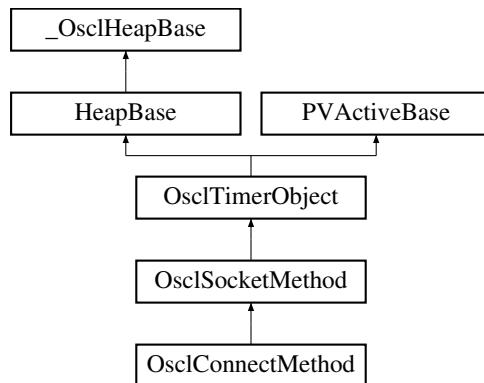
The documentation for this class was generated from the following file:

- [oscl_registry_serv_impl.h](#)

6.109 OsclConnectMethod Class Reference

```
#include <oscl_socket_connect.h>
```

Inheritance diagram for OsclConnectMethod::



Public Methods

- [~OsclConnectMethod \(\)](#)
- [TPVSocketEvent Connect \(OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsclConnectRequest * ConnectRequest \(\)](#)

Static Public Methods

- [OsclConnectMethod * NewL \(OsclIPSocketI &c\)](#)

6.109.1 Constructor & Destructor Documentation

6.109.1.1 OsclConnectMethod::~OsclConnectMethod ()

6.109.2 Member Function Documentation

6.109.2.1 TPVSocketEvent OsclConnectMethod::Connect (OsclNetworkAddress & aAddress, int32 aTimeout)

6.109.2.2 OsclConnectRequest* OsclConnectMethod::ConnectRequest () [inline]

6.109.2.3 OsclConnectMethod* OsclConnectMethod::NewL (OsclIPSocketI & c) [static]

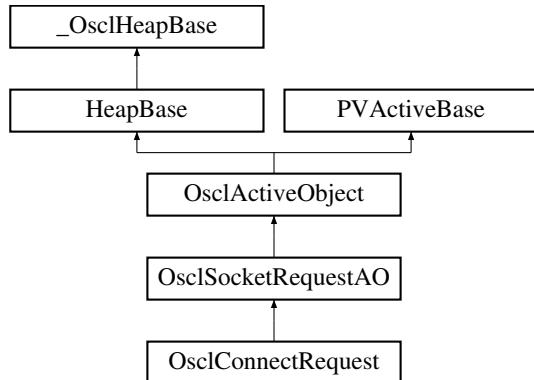
The documentation for this class was generated from the following file:

- [oscl_socket_connect.h](#)

6.110 OsclConnectRequest Class Reference

```
#include <oscl_socket_connect.h>
```

Inheritance diagram for OsclConnectRequest::



Public Methods

- [OsclConnectRequest \(OsclSocketMethod &c\)](#)
- [void Connect \(OsclNetworkAddress &aAddress\)](#)

6.110.1 Detailed Description

This is the AO that interacts with the socket server

6.110.2 Constructor & Destructor Documentation

6.110.2.1 OsclConnectRequest::OsclConnectRequest ([OsclSocketMethod & c](#)) [inline]

6.110.3 Member Function Documentation

6.110.3.1 void OsclConnectRequest::Connect ([OsclNetworkAddress & aAddress](#))

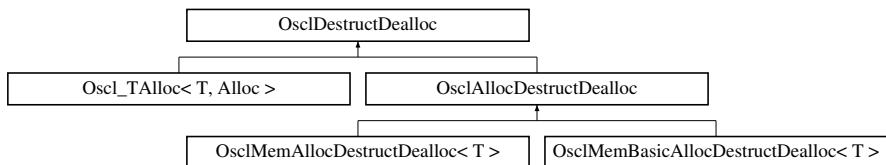
The documentation for this class was generated from the following file:

- [oscl_socket_connect.h](#)

6.111 OsclDestructDealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for OsclDestructDealloc::



Public Methods

- virtual void [destruct_and_dealloc \(OsclAny *ptr\)=0](#)

6.111.1 Member Function Documentation

6.111.1.1 virtual void OsclDestructDealloc::destruct_and_dealloc (OsclAny * ptr) [pure virtual]

Implemented in [Oscl_TAlloc< T, Alloc >](#), [OsclMemAllocDestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [Oscl_TAlloc< entry_type, Alloc >](#), [Oscl_TAlloc< node_type, TagTree_Allocator >](#), [Oscl_TAlloc< node_type, alloc_type >](#), [Oscl_TAlloc< MM_StatsNodeTagTreeType, OsclMemBasicAllocator >](#), [Oscl_TAlloc< char, alloc_type >](#), [Oscl_TAlloc< tag_base_unit, Alloc >](#), [Oscl_TAlloc< PVLogger, alloc_type >](#), and [Oscl_TAlloc< node_type, Alloc >](#).

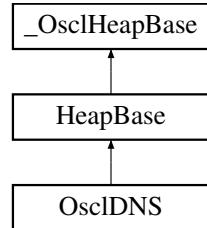
The documentation for this class was generated from the following file:

- [oscl_defalloc.h](#)

6.112 OsclDNS Class Reference

```
#include <oscl_dns.h>
```

Inheritance diagram for OsclDNS::



Public Methods

- OSCL_IMPORT_REF ~OsclDNS ()
- OSCL_IMPORT_REF TPVDNSEvent GetHostByName (char *name, OsclNetworkAddress &addr, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelGetHostByName ()

Static Public Methods

- OSCL_IMPORT_REF OsclDNS * NewL (Oscl_DefAlloc &alloc, OsclSocketServ &aServ, OsclDNSObserver &aObserver, uint32 aId)

Friends

- class OsclDNSRequestAO

6.112.1 Detailed Description

The DNS class

6.112.2 Constructor & Destructor Documentation

6.112.2.1 OSCL_IMPORT_REF OsclDNS::~OsclDNS ()

Destructor.

Note: the application must de-allocate the DNS object using the same allocator that was passed in the NewL object creation call.

6.112.3 Member Function Documentation

6.112.3.1 OSCL_IMPORT_REF void OsclDNS::CancelGetHostByName ()

Cancel GetHostByName

This method will cancel any pending GetHostByName operation on the current object, causing the GetHostByName to complete with error EPVDNSCancel. If there is no pending GetHostByName operation, this method will have no effect.

6.112.3.2 OSCL_IMPORT_REF TPVDNSEvent OsclDNS::GetHostByName (char * *name*, OsclNetworkAddress & *addr*, int32 *aTimeoutMsec* = -1)

GetHostByName. This is an asynchronous method.

Parameters:

name: Null-terminated string containing the host name.

addr: The output address. The ipAddr field will contain the network address of the host in dotted decimal notation.

aTimeoutMsec: A timeout for the request in milliseconds, or (-1) to indicate infinite wait. @returns: EPVDNSPending for success, EPVDNSFailure for failure.

6.112.3.3 OSCL_IMPORT_REF OsclDNS* OsclDNS::NewL (Oscl_DefAlloc & *alloc*, OsclSocketServ & *aServ*, OsclDNSObserver & *aObserver*, uint32 *aId*) [static]

DNS object creation.

Parameters:

alloc: Memory allocator

aServ: Socket server.

aObserver: DNS Event observer

aId: Unique ID for this DNS object. This ID will be included in all callbacks associated with this DNS object.

6.112.4 Friends And Related Function Documentation

6.112.4.1 friend class OsclDNSRequestAO [friend]

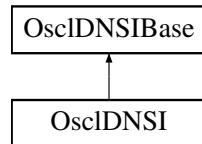
The documentation for this class was generated from the following file:

- [oscl_dns.h](#)

6.113 OsclDNSI Class Reference

```
#include <oscl_dns_imp_pv.h>
```

Inheritance diagram for OsclDNSI::



Public Methods

- [~OsclDNSI \(\)](#)
- int32 [Open \(OsclSocketServI &aServer\)](#)
- int32 [Close \(\)](#)
- void [GetHostByName \(GetHostNameParam &, OsclDNSRequestAO &\)](#)
- void [GetHostByNameSuccess \(GetHostNameParam &\)](#)

Static Public Methods

- OsclDNSI * [NewL \(Oscl_DefAlloc &a\)](#)

Friends

- class [OsclDNSRequest](#)
- class [DNSRequestParam](#)

6.113.1 Detailed Description

OsclDNSI, non-Symbian implementation

6.113.2 Constructor & Destructor Documentation

6.113.2.1 OsclDNSI::~OsclDNSI ()

6.113.3 Member Function Documentation

6.113.3.1 int32 OsclDNSI::Close () [virtual]

Implements [OsclDNSIBase](#).

6.113.3.2 void OsclDNSI::GetHostByName (GetHostNameParam &, OsclDNSRequestAO &) [virtual]

Implements [OsclDNSIBase](#).

6.113.3.3 void OsclDNSI::GetHostByNameSuccess ([GetHostNameParam](#) &) [virtual]

Implements [OsclDNSIBase](#).

6.113.3.4 OsclDNSI* OsclDNSI::NewL ([Oscl_DefAlloc](#) & *a*) [static]

6.113.3.5 int32 OsclDNSI::Open ([OsclSocketServI](#) & *aServer*) [virtual]

Implements [OsclDNSIBase](#).

6.113.4 Friends And Related Function Documentation

6.113.4.1 friend class DNSRequestParam [friend]

6.113.4.2 friend class OsclDNSRequest [friend]

Reimplemented from [OsclDNSIBase](#).

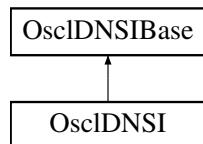
The documentation for this class was generated from the following file:

- [oscl_dns_imp_pv.h](#)

6.114 OsclDNSIBase Class Reference

```
#include <oscl_dns_imp_base.h>
```

Inheritance diagram for OsclDNSIBase::



Public Methods

- virtual ~OsclDNSIBase ()
- virtual int32 Open (OsclSocketServI &aServer)=0
- virtual int32 Close ()=0
- virtual void GetHostByName (GetHostNameParam &, OsclDNSRequestAO &)=0
- virtual void GetHostByNameSuccess (GetHostNameParam &)=0
- void CancelFxn (TPVDNSFxn)

Protected Methods

- OsclDNSIBase (Oscl_DefAlloc &a)
- virtual bool IsReady (OsclDNSRequestAO &aObject)=0
- virtual void CancelGetHostName ()=0

Protected Attributes

- Oscl_DefAlloc & iAlloc
- OsclSocketServI * iSocketServ

Friends

- class OsclDNSRequest
- class OsclGetHostNameRequest

6.114.1 Detailed Description

OsclDNSIBase is a common base class for all implementations.

6.114.2 Constructor & Destructor Documentation

6.114.2.1 `virtual OsclDNSIBase::~OsclDNSIBase () [virtual]`

6.114.2.2 `OsclDNSIBase::OsclDNSIBase (Oscl_DefAlloc & a) [protected]`

6.114.3 Member Function Documentation

6.114.3.1 `void OsclDNSIBase::CancelFxn (TPVDNSFxn)`

6.114.3.2 `virtual void OsclDNSIBase::CancelGetHostByName () [protected, pure virtual]`

6.114.3.3 `virtual int32 OsclDNSIBase::Close () [pure virtual]`

Implemented in [OsclDNSI](#).

6.114.3.4 `virtual void OsclDNSIBase::GetHostByName (GetHostNameParam &, OsclDNSRequestAO &) [pure virtual]`

Implemented in [OsclDNSI](#).

6.114.3.5 `virtual void OsclDNSIBase::GetHostByNameSuccess (GetHostNameParam &) [pure virtual]`

Implemented in [OsclDNSI](#).

6.114.3.6 `virtual bool OsclDNSIBase::IsReady (OsclDNSRequestAO & aObject) [protected, pure virtual]`

6.114.3.7 `virtual int32 OsclDNSIBase::Open (OsclSocketServI & aServer) [pure virtual]`

Implemented in [OsclDNSI](#).

6.114.4 Friends And Related Function Documentation

6.114.4.1 `friend class OsclDNSRequest [friend]`

Reimplemented in [OsclDNSI](#).

6.114.4.2 `friend class OsclGetHostByNameRequest [friend]`

6.114.5 Field Documentation

6.114.5.1 `Oscl_DefAlloc& OsclDNSIBase::iAlloc [protected]`

6.114.5.2 `OsclSocketServI* OsclDNSIBase::iSocketServ [protected]`

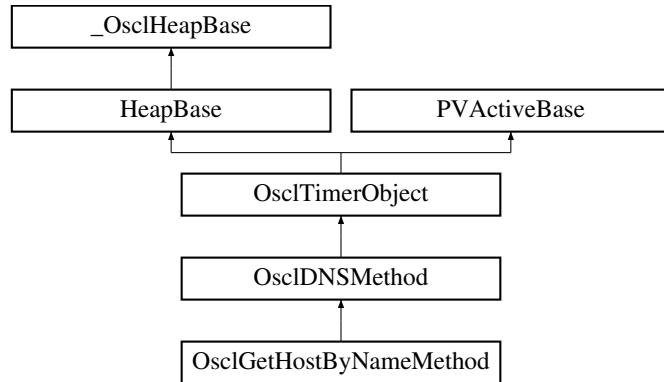
The documentation for this class was generated from the following file:

- [oscl_dns_imp_base.h](#)

6.115 OsclDNSMethod Class Reference

```
#include <oscl_dns_method.h>
```

Inheritance diagram for OsclDNSMethod::



Public Methods

- [OsclDNSMethod \(Oscl_DefAlloc &a, const char *name, TPVDNSFxn fxn\)](#)
- void [Abort \(\)](#)
- void [AbortAll \(\)](#)
- void [CancelMethod \(\)](#)
- void [Run \(\)](#)

Data Fields

- [OsclDNSObserver * iDNSObserver](#)
- uint32 [iId](#)
- [Oscl_DefAlloc & iAlloc](#)
- [TPVDNSFxn iDNSFxn](#)
- [PVLogger * iLogger](#)

Protected Methods

- void [ConstructL \(OsclDNSObserver *aObserver, OsclDNSRequestAO *aAO, uint32 aId\)](#)
- bool [StartMethod \(int32 aTimeoutMsec\)](#)
- void [MethodDone \(\)](#)

Protected Attributes

- [OsclDNSRequestAO * iDNSRequestAO](#)

6.115.1 Detailed Description

This is the base class for all socket methods. It provides the timeout on socket requests.

6.115.2 Constructor & Destructor Documentation

6.115.2.1 OsclDNSMethod::OsclDNSMethod ([Oscl_DefAlloc](#) & *a*, const char * *name*, [TPVDNSFxn](#) *fxn*) [inline]

6.115.3 Member Function Documentation

6.115.3.1 void OsclDNSMethod::Abort ()

6.115.3.2 void OsclDNSMethod::AbortAll ()

6.115.3.3 void OsclDNSMethod::CancelMethod ()

6.115.3.4 void OsclDNSMethod::ConstructL ([OsclDNSObserver](#) * *aObserver*, [OsclDNSRequestAO](#) * *aAO*, uint32 *aId*) [protected]

6.115.3.5 void OsclDNSMethod::MethodDone () [protected]

6.115.3.6 void OsclDNSMethod::Run () [virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request

2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implements [PVActiveBase](#).

6.115.3.7 **bool OsclDNSMethod::StartMethod (int32 *aTimeoutMsec*)** [protected]

6.115.4 Field Documentation

6.115.4.1 **Oscl_DefAlloc& OsclDNSMethod::iAlloc**

6.115.4.2 **TPVDNSFxn OsclDNSMethod::iDNSFxn**

6.115.4.3 **OsclDNSObserver* OsclDNSMethod::iDNSObserver**

6.115.4.4 **OsclDNSRequestAO* OsclDNSMethod::iDNSRequestAO** [protected]

6.115.4.5 **uint32 OsclDNSMethod::iId**

6.115.4.6 **PVLogger* OsclDNSMethod::iLogger**

The documentation for this class was generated from the following file:

- [oscl_dns_method.h](#)

6.116 OsclDNSObserver Class Reference

```
#include <oscl_dns.h>
```

Public Methods

- virtual OSCL_IMPORT_REF void [HandleDNSEvent](#) (int32 aId, [TPVDNSFxn](#) aFxn, [TPVDNSEvent](#) aEvent, int32 aError)=0
- virtual [~OsclDNSObserver](#) ()

6.116.1 Detailed Description

DNS event observer. The client implements this to get asynchronous command completion.

6.116.2 Constructor & Destructor Documentation

6.116.2.1 virtual OsclDNSObserver::~OsclDNSObserver () [inline, virtual]

6.116.3 Member Function Documentation

**6.116.3.1 virtual OSCL_IMPORT_REF void OsclDNSObserver::HandleDNSEvent (int32 *aId*,
[TPVDNSFxn](#) *aFxn*, [TPVDNSEvent](#) *aEvent*, int32 *aError*) [pure virtual]**

DNS Event callback.

Parameters:

***aId*:** The ID that was supplied when the DNS object was created.

***aEvent*:** Function completion event. Will be EPVDNSSuccess, EPVDNSTimeout, or EPVDNSFailure.

***aError*:** When the event is EPVDNSFailure, this may contain a platform-specific error code, or zero if none is available.

The documentation for this class was generated from the following file:

- [oscl_dns.h](#)

6.117 OsclDNSRequest Class Reference

```
#include <oscl_dns_request.h>
```

Public Methods

- [OsclDNSRequest \(\)](#)
- [~OsclDNSRequest \(\)](#)
- [void CancelRequest \(\)](#)
- [void Complete \(bool, int32 aStatus, int32 aSockErr\)](#)
- [void Activate \(DNSRequestParam *iParam, OsclDNSRequestAO &a\)](#)

Data Fields

- [OsclDNSRequestAO * iDNSRequestAO](#)
- [DNSRequestParam * iDNSRequestParam](#)
- [bool iActive](#)

6.117.1 Detailed Description

This class defines the interface to the dns implementation threads.

6.117.2 Constructor & Destructor Documentation

6.117.2.1 [OsclDNSRequest::OsclDNSRequest \(\) \[inline\]](#)

6.117.2.2 [OsclDNSRequest::~OsclDNSRequest \(\) \[inline\]](#)

6.117.3 Member Function Documentation

6.117.3.1 [void OsclDNSRequest::Activate \(DNSRequestParam * iParam, OsclDNSRequestAO & a\)](#)

6.117.3.2 [void OsclDNSRequest::CancelRequest \(\)](#)

6.117.3.3 [void OsclDNSRequest::Complete \(bool, int32 aStatus, int32 aSockErr\)](#)

6.117.4 Field Documentation

6.117.4.1 [bool OsclDNSRequest::iActive](#)

6.117.4.2 [OsclDNSRequestAO* OsclDNSRequest::iDNSRequestAO](#)

6.117.4.3 [DNSRequestParam* OsclDNSRequest::iDNSRequestParam](#)

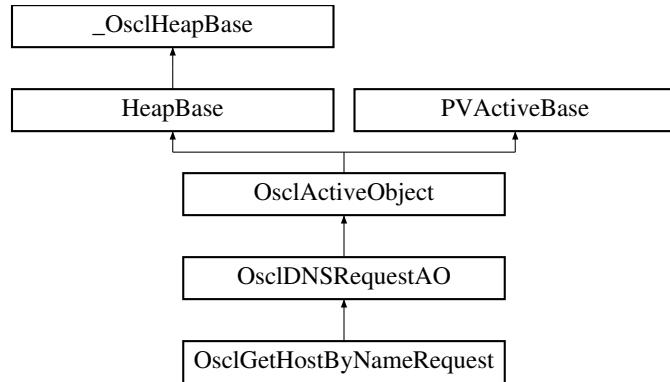
The documentation for this class was generated from the following file:

- [oscl_dns_request.h](#)

6.118 OsclDNSRequestAO Class Reference

```
#include <oscl_dns_method.h>
```

Inheritance diagram for OsclDNSRequestAO::



Protected Methods

- [OsclDNSRequestAO](#) (const char *name)
- void [ConstructL](#) ([OsclDNSI](#) *aDNS, [OsclDNSMethod](#) *aMethod)
- void [Abort](#) ()
- void [NewRequest](#) ()
- void [RequestDone](#) ()
- int [GetSocketError](#) ()
- [OsclSocketServI](#) * [Serv](#) ()
- void [DoCancel](#) ()
- void [Run](#) ()
- virtual void [Success](#) ()

Protected Attributes

- [OsclDNSI](#) * [iDNSI](#)
- [OsclDNSMethod](#) * [iDNSMethod](#)
- int32 [iSocketError](#)
- [PVLogger](#) * [iLogger](#)

Friends

- class [OsclDNSI](#)
- class [OsclDNSMethod](#)
- class [OsclDNSRequest](#)
- class [DNSRequestParam](#)

6.118.1 Detailed Description

This is the base class for all requests to the socket server.

6.118.2 Constructor & Destructor Documentation

6.118.2.1 `OsclDNSRequestAO::OsclDNSRequestAO (const char * name)` [inline, protected]

6.118.3 Member Function Documentation

6.118.3.1 `void OsclDNSRequestAO::Abort ()` [inline, protected]

6.118.3.2 `void OsclDNSRequestAO::ConstructL (OsclDNSI * aDNS, OsclDNSMethod * aMethod)` [inline, protected]

6.118.3.3 `void OsclDNSRequestAO::DoCancel ()` [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Reimplemented from [OsclActiveObject](#).

6.118.3.4 `int OsclDNSRequestAO::GetSocketError ()` [protected]

6.118.3.5 `void OsclDNSRequestAO::NewRequest ()` [protected]

6.118.3.6 `void OsclDNSRequestAO::RequestDone ()` [protected]

6.118.3.7 `void OsclDNSRequestAO::Run ()` [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implements [PVActiveBase](#).

6.118.3.8 **OsclSocketServI* OsclDNSRequestAO::Serv ()** [protected]

6.118.3.9 **virtual void OsclDNSRequestAO::Success ()** [inline, protected, virtual]

6.118.4 Friends And Related Function Documentation

6.118.4.1 **friend class DNSRequestParam** [friend]

6.118.4.2 **friend class OsclDNSI** [friend]

6.118.4.3 **friend class OsclDNSMethod** [friend]

6.118.4.4 **friend class OsclDNSRequest** [friend]

6.118.5 Field Documentation

6.118.5.1 **OsclDNSI* OsclDNSRequestAO::iDNSI** [protected]

6.118.5.2 **OsclDNSMethod* OsclDNSRequestAO::iDNSMethod** [protected]

6.118.5.3 **PVLogger* OsclDNSRequestAO::iLogger** [protected]

6.118.5.4 **int32 OsclDNSRequestAO::iSocketError** [protected]

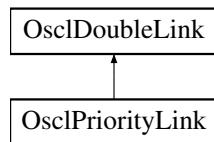
The documentation for this class was generated from the following file:

- [oscl_dns_method.h](#)

6.119 OsclDoubleLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleLink::



Public Methods

- [OsclDoubleLink \(\)](#)
- void [Remove \(\)](#)
- void [InsertAfter \(OsclDoubleLink *aLink\)](#)
- void [InsertBefore \(OsclDoubleLink *aLink\)](#)

Data Fields

- OsclDoubleLink * [iNext](#)
- OsclDoubleLink * [iPrev](#)

6.119.1 Constructor & Destructor Documentation

6.119.1.1 OsclDoubleLink::OsclDoubleLink () [inline]

6.119.2 Member Function Documentation

6.119.2.1 void OsclDoubleLink::InsertAfter (OsclDoubleLink * *aLink*)

6.119.2.2 void OsclDoubleLink::InsertBefore (OsclDoubleLink * *aLink*)

6.119.2.3 void OsclDoubleLink::Remove ()

6.119.3 Field Documentation

6.119.3.1 OsclDoubleLink* OsclDoubleLink::iNext

6.119.3.2 OsclDoubleLink* OsclDoubleLink::iPrev

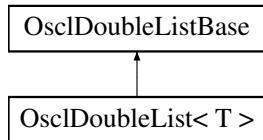
The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

6.120 OsclDoubleList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleList< T >::



Public Methods

- OSCL_INLINE OsclDoubleList()
- OSCL_INLINE OsclDoubleList(int32 anOffset)
- OSCL_INLINE void InsertHead(T &aRef)
- OSCL_INLINE void InsertTail(T &aRef)
- OSCL_INLINE bool IsHead(const T *aPtr) const
- OSCL_INLINE bool IsTail(const T *aPtr) const
- OSCL_INLINE T * Head() const
- OSCL_INLINE T * Tail() const

```
template<class T> class OsclDoubleList< T >
```

6.120.1 Constructor & Destructor Documentation

6.120.1.1 template<class T> OSCL_INLINE OsclDoubleList< T >::OsclDoubleList()

**6.120.1.2 template<class T> OSCL_INLINE OsclDoubleList< T >::OsclDoubleList(int32
anOffset)**

6.120.2 Member Function Documentation

6.120.2.1 template<class T> OSCL_INLINE T* OsclDoubleList< T >::Head()

6.120.2.2 template<class T> OSCL_INLINE void OsclDoubleList< T >::InsertHead(T & aRef)

6.120.2.3 template<class T> OSCL_INLINE void OsclDoubleList< T >::InsertTail(T & aRef)

**6.120.2.4 template<class T> OSCL_INLINE bool OsclDoubleList< T >::IsHead(const T * aPtr)
const**

**6.120.2.5 template<class T> OSCL_INLINE bool OsclDoubleList< T >::IsTail(const T * aPtr)
const**

6.120.2.6 template<class T> OSCL_INLINE T* OsclDoubleList< T >::Tail()

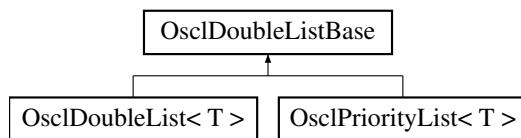
The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

6.121 OsclDoubleListBase Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleListBase::



Public Methods

- bool [IsEmpty \(\) const](#)
- void [SetOffset \(int32 anOffset\)](#)
- void [Reset \(\)](#)
- [OsclDoubleLink * getHead \(\)](#)
- int32 [getOffset \(\)](#)

Protected Methods

- [OsclDoubleListBase \(\)](#)
- [OsclDoubleListBase \(int32 anOffset\)](#)
- void [InsertHead \(OsclAny *aPtr\)](#)
- void [InsertTail \(OsclAny *aPtr\)](#)
- void [Insert \(OsclAny *aPtr\)](#)

Protected Attributes

- [OsclDoubleLink iHead](#)
- int32 [iOffset](#)

6.121.1 Constructor & Destructor Documentation

6.121.1.1 **OsclDoubleListBase::OsclDoubleListBase ()** [protected]

6.121.1.2 **OsclDoubleListBase::OsclDoubleListBase (int32 *anOffset*)** [protected]

6.121.2 Member Function Documentation

6.121.2.1 **OsclDoubleLink* OsclDoubleListBase::getHead ()** [inline]

6.121.2.2 **int32 OsclDoubleListBase::getOffset ()** [inline]

6.121.2.3 **void OsclDoubleListBase::Insert (OsclAny * *aPtr*)** [protected]

6.121.2.4 **void OsclDoubleListBase::InsertHead (OsclAny * *aPtr*)** [protected]

6.121.2.5 **void OsclDoubleListBase::InsertTail (OsclAny * *aPtr*)** [protected]

6.121.2.6 **bool OsclDoubleListBase::IsEmpty ()**

6.121.2.7 **void OsclDoubleListBase::Reset ()**

6.121.2.8 **void OsclDoubleListBase::SetOffset (int32 *anOffset*)**

6.121.3 Field Documentation

6.121.3.1 **OsclDoubleLink OsclDoubleListBase::iHead** [protected]

6.121.3.2 **int32 OsclDoubleListBase::iOffset** [protected]

The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

6.122 OsclDoubleRunner< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Public Methods

- [OsclDoubleRunner \(OsclDoubleListBase &aQue\)](#)
- void [Set \(T &aLink\)](#)
- [operator T * \(\)](#)
- T * [operator++ \(int\)](#)
- T * [operator- \(int\)](#)
- void [SetToHead \(\)](#)
- void [SetToTail \(\)](#)

Protected Attributes

- int32 [iOffset](#)
- OsclDoubleLink * [iHead](#)
- OsclDoubleLink * [iNext](#)

```
template<class T> class OsclDoubleRunner< T >
```

6.122.1 Constructor & Destructor Documentation

6.122.1.1 template<class T> OsclDoubleRunner< T >::OsclDoubleRunner ([OsclDoubleListBase & aQue](#)) [inline]

6.122.2 Member Function Documentation

6.122.2.1 template<class T> OsclDoubleRunner< T >::operator T * () [inline]

6.122.2.2 template<class T> T* OsclDoubleRunner< T >::operator++ (int) [inline]

6.122.2.3 template<class T> T* OsclDoubleRunner< T >::operator- (int)

6.122.2.4 template<class T> void OsclDoubleRunner< T >::Set (T & aLink) [inline]

6.122.2.5 template<class T> void OsclDoubleRunner< T >::SetToHead () [inline]

6.122.2.6 template<class T> void OsclDoubleRunner< T >::SetToTail () [inline]

6.122.3 Field Documentation

6.122.3.1 template<class T> [OsclDoubleLink* OsclDoubleRunner< T >::iHead](#) [protected]

6.122.3.2 template<class T> [OsclDoubleLink* OsclDoubleRunner< T >::iNext](#) [protected]

6.122.3.3 template<class T> int32 [OsclDoubleRunner< T >::iOffset](#) [protected]

The documentation for this class was generated from the following file:

-
- [oscl_double_list.h](#)

6.123 OsclError Class Reference

```
#include <oscl_error.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [PushL \(_OsclHeapBase *aPtr\)](#)
- OSCL_IMPORT_REF void [PushL \(OsclAny *aPtr\)](#)
- OSCL_IMPORT_REF void [PushL \(OsclTrapItem anItem\)](#)
- OSCL_IMPORT_REF void [Pop \(\)](#)
- OSCL_IMPORT_REF void [Pop \(int32 aCount\)](#)
- OSCL_IMPORT_REF void [PopDealloc \(\)](#)
- OSCL_IMPORT_REF void [PopDealloc \(int32 aCount\)](#)
- OSCL_IMPORT_REF void [Leave \(int32 aReason\)](#)
- OSCL_IMPORT_REF void [LeaveIfNull \(OsclAny *a\)](#)
- OSCL_IMPORT_REF void [LeaveIfError \(int32 aReason\)](#)

6.123.1 Detailed Description

User Error class

6.123.2 Member Function Documentation

6.123.2.1 OSCL_IMPORT_REF void OsclError::Leave (int32 *aReason*) [static]

Do a Leave error, with the given reason code. When a leave occurs, all items on the cleanup stack for the current trap level will be destroyed, and execution will jump to the trap handler.

6.123.2.2 OSCL_IMPORT_REF void OsclError::LeaveIfError (int32 *aReason*) [static]

Evaluate the input parameter, and if it is an error code (non-zero), then do a Leave with the provided reason code.

6.123.2.3 OSCL_IMPORT_REF void OsclError::LeaveIfNull (OsclAny * *a*) [static]

Evaluate the input parameter, and if it is null, do a Leave with OsclErrNoMemory reason code.

6.123.2.4 OSCL_IMPORT_REF void OsclError::Pop (int32 *aCount*) [static]

Pop the cleanup stack N times

6.123.2.5 OSCL_IMPORT_REF void OsclError::Pop () [static]

Pop the cleanup stack

6.123.2.6 OSCL_IMPORT_REF void OsclError::PopDealloc (int32 *aCount*) [static]

PopDealloc N times

6.123.2.7 OSCL_IMPORT_REF void OsclError::PopDealloc () [static]

Destroy the item on the top of the cleanup stack and pop it

6.123.2.8 OSCL_IMPORT_REF void OsclError::PushL ([OsclTrapItem](#) *anItem*) [static]

Push an [OsclTrapItem](#) onto the cleanup stack

6.123.2.9 OSCL_IMPORT_REF void OsclError::PushL ([OsclAny](#) * *aPtr*) [static]

Push an OsclAny item onto the cleanup stack.

6.123.2.10 OSCL_IMPORT_REF void OsclError::PushL ([_OsclHeapBase](#) * *aPtr*) [static]

Push an [_OsclHeapBase](#) item onto the cleanup stack.

The documentation for this class was generated from the following file:

- [oscl_error.h](#)

6.124 OsclErrorAllocator Class Reference

This class provides static methods to invoke the user defined memory allocation routines.

```
#include <oscl_error_allocator.h>
```

Public Methods

- **OsclErrorAllocator (Oscl_DefAlloc *allocator)**
constructor method
- **void * operator new (uint32 size, OsclAny *aPtr)**
placement new operator that allocates memory using the user defined methods
- **void operator delete (OsclAny *aPtr, OsclAny *aPtr2)**
delete operator that doesn't do anything, user has to deallocate manually

Static Public Methods

- **OsclAny * allocate (uint32 aSize)**
static method to allocate a block of memory on heap
- **OsclAny deallocate (OsclAny *aPointer)**
static method to deallocate a block of memory on heap

6.124.1 Detailed Description

This class provides static methods to invoke the user defined memory allocation routines.

This class must be instantiated before the static methods are called, else asserts will happen

6.124.2 Constructor & Destructor Documentation

6.124.2.1 OsclErrorAllocator::OsclErrorAllocator (**Oscl_DefAlloc * allocator**) [inline]

constructor method

Parameters:

allocator - a pointer to the concrete object that provides the allocator/deallocator

6.124.3 Member Function Documentation

6.124.3.1 OsclAny* OsclErrorAllocator::allocate (uint32 *aSize*) [inline, static]

static method to allocate a block of memory on heap

Parameters:

aSize - number of bytes to allocate

6.124.3.2 OsclAny OsclErrorAllocator::deallocate (OsclAny * aPointer) [inline, static]

static method to deallocate a block of memory on heap

Parameters:

aPointer - pointer to block of memory to be deallocated

6.124.3.3 void OsclErrorAllocator::operator delete (OsclAny * aPtr, OsclAny * aPtr2) [inline]

delete operator that doesn't do anything, user has to deallocate manually

6.124.3.4 void* OsclErrorAllocator::operator new (uint32 size, OsclAny * aPtr) [inline]

placement new operator that allocates memory using the user defined methods

The documentation for this class was generated from the following file:

- [oscl_error_allocator.h](#)

6.125 OsclErrorTrap Class Reference

```
#include <oscl_error.h>
```

Static Public Methods

- OSCL_IMPORT_REF int32 [Init \(Oscl_DefAlloc *aAlloc=NULL\)](#)
- OSCL_IMPORT_REF int32 [Cleanup \(\)](#)
- OSCL_IMPORT_REF [OsclErrorTrapImp * GetErrorTrapImp \(\)](#)

6.125.1 Member Function Documentation

6.125.1.1 OSCL_IMPORT_REF int32 OsclErrorTrap::Cleanup () [static]

Cleanup and destroy error trap for the calling thread.

Returns:

0 for success, or an error

6.125.1.2 OSCL_IMPORT_REF [OsclErrorTrapImp*](#) OsclErrorTrap::GetErrorTrapImp () [static]

Get the ErrorTrapImp for the current thread. Leaves on error.

6.125.1.3 OSCL_IMPORT_REF int32 OsclErrorTrap::Init ([Oscl_DefAlloc * aAlloc = NULL](#)) [static]

Allocate and initialize error trap for the calling thread.

Parameters:

aAlloc: optional, allocator to use for the internal implementation.

Returns:

0 for success, or an error

The documentation for this class was generated from the following file:

- [oscl_error.h](#)

6.126 OsclErrorTrapImp Class Reference

```
#include <oscl_error_trapcleanup.h>
```

Public Methods

- OSCL_IMPORT_REF void [UnTrap \(\)](#)

Static Public Methods

- OSCL_IMPORT_REF OsclErrorTrapImp * [Trap \(\)](#)
- OSCL_IMPORT_REF OsclErrorTrapImp * [TrapNoTls \(OsclErrorTrapImp *\)](#)

Data Fields

- [OsclJump * iJumpData](#)
- int32 [iLeave](#)
- [OsclTrapStack * iTrapStack](#)

Friends

- class [OsclErrorTrap](#)
- class [OsclError](#)
- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclJump](#)
- class [OsclJumpMark](#)
- class [OsclTrapStack](#)
- class [CPVInterfaceProxy](#)
- class [OsclScheduler](#)

6.126.1 Detailed Description

A per-thread cleanup stack with nested trap support.

6.126.2 Member Function Documentation

6.126.2.1 OSCL_IMPORT_REF OsclErrorTrapImp* OsclErrorTrapImp::Trap () [static]

PV trap cleanup. Public for use in macros only.

6.126.2.2 OSCL_IMPORT_REF OsclErrorTrapImp* OsclErrorTrapImp::TrapNoTls (OsclErrorTrapImp *) [static]

6.126.2.3 OSCL_IMPORT_REF void OsclErrorTrapImp::UnTrap ()

these are used in public macros, but aren't intended as public methods or members.

6.126.3 Friends And Related Function Documentation

6.126.3.1 **friend class CPVInterfaceProxy [friend]**

6.126.3.2 **friend class OsclError [friend]**

6.126.3.3 **friend class OsclErrorTrap [friend]**

6.126.3.4 **friend class OsclExecScheduler [friend]**

6.126.3.5 **friend class OsclExecSchedulerCommonBase [friend]**

6.126.3.6 **friend class OsclJump [friend]**

6.126.3.7 **friend class OsclJumpMark [friend]**

6.126.3.8 **friend class OsclScheduler [friend]**

6.126.3.9 **friend class OsclTrapStack [friend]**

6.126.4 Field Documentation

6.126.4.1 **OsclJump* OsclErrorTrapImp::iJumpData**

6.126.4.2 **int32 OsclErrorTrapImp::iLeave**

6.126.4.3 **OsclTrapStack* OsclErrorTrapImp::iTrapStack**

The documentation for this class was generated from the following file:

- [oscl_error_trapcleanup.h](#)

6.127 OsclException< LeaveCode > Class Template Reference

`oscl_exception.h` contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

```
#include <oscl_exception.h>
```

Public Methods

- `OsclException ()`

Static Public Methods

- `int getLeaveCode ()`

6.127.1 Detailed Description

`template<int LeaveCode> class OsclException< LeaveCode >`

`oscl_exception.h` contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

All PacketVideo exception classes will be derived from the OsclException class. Each derived class will have a static function where the leave code can be obtained. This avoids the issue of having static members in a DLL. The function needs to be static so it can be called without an instance of the class

6.127.2 Constructor & Destructor Documentation

6.127.2.1 `template<int LeaveCode> OsclException< LeaveCode >::OsclException ()`
[inline]

6.127.3 Member Function Documentation

6.127.3.1 `template<int LeaveCode> int OsclException< LeaveCode >::getLeaveCode ()`
[inline, static]

The documentation for this class was generated from the following file:

- `oscl_exception.h`

6.128 OsclExclusiveArrayPtr< T > Class Template Reference

The OsclExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory.

```
#include <oscl_exclusive_ptr.h>
```

Public Methods

- **OsclExclusiveArrayPtr (T *inPtr=0)**
Default constructor Initializes the pointer and takes ownership.
- **OsclExclusiveArrayPtr (OsclExclusiveArrayPtr< T > &_Y)**
Copy constructor.
- **OsclExclusiveArrayPtr< T > & operator= (OsclExclusiveArrayPtr< T > &_Y)**
Assignment operator from an another OsclExclusiveArrayPtr.
- **virtual ~OsclExclusiveArrayPtr ()**
Destructor.
- **T & operator* () const**
The indirection operator () accesses a value indirectly, through a pointer.*
- **T * operator-> () const**
The indirection operator (->) accesses a value indirectly, through a pointer.
- **T * get () const**
get() method returns the pointer, currently owned by the class.
- **T * release ()**
release() method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.
- **bool set (T *ptr)**
set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

Protected Attributes

- **T * _Ptr**

6.128.1 Detailed Description

template<class T> class OsclExclusiveArrayPtr< T >

The OsclExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an `OsclExclusivePtr` object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The `OsclExclusivePtr` is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

6.128.2 Constructor & Destructor Documentation

6.128.2.1 template<class T> OsclExclusiveArrayPtr< T >::OsclExclusiveArrayPtr (T * *inPtr* = 0) [inline, explicit]

Default constructor Initializes the pointer and takes ownership.

6.128.2.2 template<class T> OsclExclusiveArrayPtr< T >::OsclExclusiveArrayPtr (OsclExclusiveArrayPtr< T > & *_Y*) [inline]

Copy constructor.

Initializes the pointer and takes ownership from another `OsclExclusiveArrayPtr`. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

6.128.2.3 template<class T> virtual OsclExclusiveArrayPtr< T >::~OsclExclusiveArrayPtr () [inline, virtual]

Destructor.

The pointer is deleted in case this class still has ownership

6.128.3 Member Function Documentation

6.128.3.1 template<class T> T* OsclExclusiveArrayPtr< T >::get () const [inline]

`get()` method returns the pointer, currently owned by the class.

6.128.3.2 template<class T> T& OsclExclusiveArrayPtr< T >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the `OsclExclusiveArrayPtr` can be used like the regular pointer that it was initialized with.

6.128.3.3 template<class T> T* OsclExclusiveArrayPtr< T >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the `OsclExclusiveArrayPtr` can be used like the regular pointer that it was initialized with.

6.128.3.4 template<class T> OsclExclusiveArrayPtr<T>& OsclExclusiveArrayPtr< T >::operator= (OsclExclusiveArrayPtr< T > & _Y) [inline]

Assignment operator from an another OsclExclusiveArrayPtr.

Parameters:

_Y The value parameter should be another OsclExclusiveArrayPtr

Returns:

Returns a reference to this OsclExclusiveArrayPtr instance with pointer initialized.

Precondition:

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the OsclExclusiveArrayPtr given as the input parameter. The ownership of the pointer is transferred.

6.128.3.5 template<class T> T* OsclExclusiveArrayPtr< T >::release () [inline]

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

6.128.3.6 template<class T> bool OsclExclusiveArrayPtr< T >::set (T *ptr) [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

6.128.4 Field Documentation

6.128.4.1 template<class T> T* OsclExclusiveArrayPtr< T >::_Ptr [protected]

The documentation for this class was generated from the following file:

- [oscl_exclusive_ptr.h](#)

6.129 OsclExclusivePtr< T > Class Template Reference

The OsclExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory.

```
#include <oscl_exclusive_ptr.h>
```

Public Methods

- **OsclExclusivePtr** (T *inPtr=0)
Default constructor Initializes the pointer and takes ownership.
- **OsclExclusivePtr** (OsclExclusivePtr< T > &_Y)
Copy constructor.
- OsclExclusivePtr< T > & **operator=** (OsclExclusivePtr< T > &_Y)
Assignment operator from an another OsclExclusivePtr.
- virtual ~**OsclExclusivePtr** ()
Destructor.
- T & **operator *** () const
The indirection operator () accesses a value indirectly, through a pointer.*
- T * **operator ->** () const
The indirection operator (->) accesses a value indirectly, through a pointer.
- T * **get** () const
***get()** method returns the pointer, currently owned by the class.*
- T * **release** ()
***release()** method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool **set** (T *ptr)
***set()** method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

Protected Attributes

- T * **_Ptr**

6.129.1 Detailed Description

template<class T> class OsclExclusivePtr< T >

The OsclExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an OsclExclusivePtr object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The OsclExclusivePtr is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

6.129.2 Constructor & Destructor Documentation

6.129.2.1 template<class T> OsclExclusivePtr< T >::OsclExclusivePtr (T * *inPtr* = 0) [inline, explicit]

Default constructor Initializes the pointer and takes ownership.

6.129.2.2 template<class T> OsclExclusivePtr< T >::OsclExclusivePtr (OsclExclusivePtr< T > & *_Y*) [inline]

Copy constructor.

Initializes the pointer and takes ownership from another OsclExclusivePtr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

6.129.2.3 template<class T> virtual OsclExclusivePtr< T >::~OsclExclusivePtr () [inline, virtual]

Destructor.

The pointer is deleted in case this class still has ownership

6.129.3 Member Function Documentation

6.129.3.1 template<class T> T* OsclExclusivePtr< T >::get () const [inline]

[get\(\)](#) method returns the pointer, currently owned by the class.

6.129.3.2 template<class T> T& OsclExclusivePtr< T >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusivePtr can be used like the regular pointer that it was initialized with.

6.129.3.3 template<class T> T* OsclExclusivePtr< T >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclExclusivePtr can be used like the regular pointer that it was initialized with.

6.129.3.4 template<class T> OsclExclusivePtr<T>& OsclExclusivePtr< T >::operator= (OsclExclusivePtr< T > & _Y) [inline]

Assignment operator from an another OsclExclusivePtr.

Parameters:

_Y The value parameter should be another OsclExclusivePtr

Returns:

Returns a reference to this OsclExclusivePtr instance with pointer initialized.

Precondition:

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the OsclExclusivePtr given as the input parameter. The ownership of the pointer is transferred.

6.129.3.5 template<class T> T* OsclExclusivePtr< T >::release () [inline]

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

6.129.3.6 template<class T> bool OsclExclusivePtr< T >::set (T *ptr) [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

6.129.4 Field Documentation

6.129.4.1 template<class T> T* OsclExclusivePtr< T >::_Ptr [protected]

The documentation for this class was generated from the following file:

- [oscl_exclusive_ptr.h](#)

6.130 OsclExclusivePtrA< T, Alloc > Class Template Reference

The OsclExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory.

```
#include <oscl_exclusive_ptr.h>
```

Public Methods

- **OsclExclusivePtrA** (T *inPtr=0)
Default constructor Initializes the pointer and takes ownership.
- **OsclExclusivePtrA** (OsclExclusivePtrA< T, Alloc > &_Y)
Copy constructor.
- OsclExclusivePtrA< T, Alloc > & **operator=** (OsclExclusivePtrA< T, Alloc > &_Y)
Assignment operator from an another OsclExclusiveArrayPtr.
- virtual ~**OsclExclusivePtrA** ()
Destructor.
- T & **operator *** () const
The indirection operator () accesses a value indirectly, through a pointer.*
- T * **operator ->** () const
The indirection operator (->) accesses a value indirectly, through a pointer.
- T * **get** () const
***get()** method returns the pointer, currently owned by the class.*
- T * **release** ()
***release()** method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool **set** (T *ptr)
***set()** method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

Protected Attributes

- T * **_Ptr**

6.130.1 Detailed Description

template<class T, class Alloc> class OsclExclusivePtrA< T, Alloc >

The OsclExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an [OsclExclusivePtr](#) object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The [OsclExclusivePtr](#) is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

6.130.2 Constructor & Destructor Documentation

6.130.2.1 template<class T, class Alloc> OsclExclusivePtrA< T, Alloc >::OsclExclusivePtrA (T * *inPtr* = 0) [inline, explicit]

Default constructor Initializes the pointer and takes ownership.

6.130.2.2 template<class T, class Alloc> OsclExclusivePtrA< T, Alloc >::OsclExclusivePtrA (OsclExclusivePtrA< T, Alloc > & *_Y*) [inline]

Copy constructor.

Initializes the pointer and takes ownership from another [OsclExclusiveArrayPtr](#). Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

6.130.2.3 template<class T, class Alloc> virtual OsclExclusivePtrA< T, Alloc >::~OsclExclusivePtrA () [inline, virtual]

Destructor.

The pointer is deleted in case this class still has ownership

6.130.3 Member Function Documentation

6.130.3.1 template<class T, class Alloc> T* OsclExclusivePtrA< T, Alloc >::get () const [inline]

[get\(\)](#) method returns the pointer, currently owned by the class.

6.130.3.2 template<class T, class Alloc> T& OsclExclusivePtrA< T, Alloc >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the [OsclExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

6.130.3.3 template<class T, class Alloc> T* OsclExclusivePtrA< T, Alloc >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the [OsclExclusiveArrayPtr](#) can be used like the regular pointer that it was initialized with.

6.130.3.4 template<class T, class Alloc> OsclExclusivePtrA<T, Alloc>& OsclExclusivePtrA< T, Alloc >::operator= (OsclExclusivePtrA< T, Alloc > & _Y) [inline]

Assignment operator from an another [OsclExclusiveArrayPtr](#).

Parameters:

_Y The value parameter should be another [OsclExclusiveArrayPtr](#)

Returns:

Returns a reference to this [OsclExclusiveArrayPtr](#) instance with pointer initialized.

Precondition:

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the [OsclExclusiveArrayPtr](#) given as the input parameter. The ownership of the pointer is transferred.

6.130.3.5 template<class T, class Alloc> T* OsclExclusivePtrA< T, Alloc >::release () [inline]

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

6.130.3.6 template<class T, class Alloc> bool OsclExclusivePtrA< T, Alloc >::set (T *ptr) [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

6.130.4 Field Documentation

6.130.4.1 template<class T, class Alloc> T* OsclExclusivePtrA< T, Alloc >::_Ptr [protected]

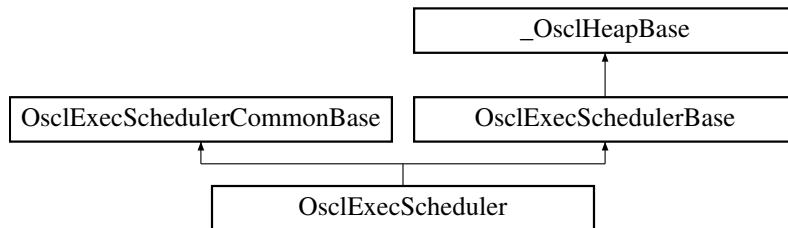
The documentation for this class was generated from the following file:

- [oscl_exclusive_ptr.h](#)

6.131 OsclExecScheduler Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for OsclExecScheduler::



Public Methods

- OSCL_IMPORT_REF void [RunSchedulerNonBlocking](#) (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)
- OSCL_IMPORT_REF void [RegisterForCallback](#) ([OsclSchedulerObserver](#) *aCallback, [OsclAny](#) *aCallbackContext)

Static Public Methods

- OSCL_IMPORT_REF [OsclExecScheduler](#) * [Current](#) ()

Friends

- class [OsclScheduler](#)

6.131.1 Member Function Documentation

6.131.1.1 OSCL_IMPORT_REF OsclExecScheduler* OsclExecScheduler::Current () [static]

Get currently installed scheduler for calling thread, or NULL if no scheduler is installed.

6.131.1.2 OSCL_IMPORT_REF void OsclExecScheduler::RegisterForCallback ([OsclSchedulerObserver](#) * aCallback, [OsclAny](#) * aCallbackContext)

Register for a notification when non-blocking scheduler needs to run again.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

6.131.1.3 OSCL_IMPORT_REF void OsclExecScheduler::RunSchedulerNonBlocking (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)

Run PV scheduler in non-blocking mode. This call returns when the desired number of Run calls have been made, or when there are no more active objects that are ready to run.

Parameters:

aTargetCount: (input param) the maximum number of Run calls to make.

aReady: (output param) tells the number of active objects that are currently ready to run.

aDelayMsec: (output param) If no active objects are ready to run, but one or more active objects are waiting on timers, this parameter will tell the time interval from the current time until the first of the pending timer objects will be ready to run, in milliseconds.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

6.131.2 Friends And Related Function Documentation

6.131.2.1 friend class OsclScheduler [friend]

Reimplemented from [OsclExecSchedulerCommonBase](#).

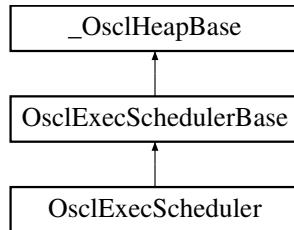
The documentation for this class was generated from the following file:

- [oscl_scheduler.h](#)

6.132 OsclExecSchedulerBase Class Reference

```
#include <oscl_scheduler_types.h>
```

Inheritance diagram for OsclExecSchedulerBase::



Friends

- class [OsclExecScheduler](#)
- class [OsclCoeActiveScheduler](#)
- class [PVActiveBase](#)

6.132.1 Detailed Description

OsclActiveSchedulerBase is the base for [OsclExecScheduler](#). The non-Symbian OsclActiveSchedulerBase class is functionally similar to a subset of Symbian CActiveScheduler class.

6.132.2 Friends And Related Function Documentation

6.132.2.1 friend class OsclCoeActiveScheduler [friend]

6.132.2.2 friend class OsclExecScheduler [friend]

6.132.2.3 friend class PVActiveBase [friend]

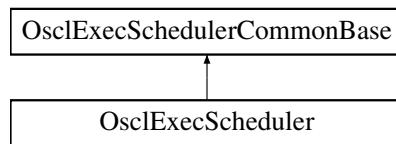
The documentation for this class was generated from the following file:

- [oscl_scheduler_types.h](#)

6.133 OsclExecSchedulerCommonBase Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for OsclExecSchedulerCommonBase::



Public Methods

- OSCL_IMPORT_REF void [StartScheduler](#) (OsclSemaphore *sem=NULL)
- OSCL_IMPORT_REF void [StopScheduler](#) ()
- OSCL_IMPORT_REF void [SuspendScheduler](#) ()
- OSCL_IMPORT_REF void [ResumeScheduler](#) ()
- OSCL_IMPORT_REF void [StartNativeScheduler](#) ()

Static Public Methods

- OSCL_IMPORT_REF [OsclNameString< PVSCHEDNAMELEN > *](#) [GetName](#) ()
- OSCL_IMPORT_REF uint32 [GetId](#) ()

Protected Types

- enum [TOtherExecStats](#) { [EOtherExecStats_WaitTime](#), [EOtherExecStats_QueueTime](#), [EOtherExecStats_NativeOS](#), [EOtherExecStats_ReleaseTime](#), [EOtherExecStats_Last](#) }

Protected Methods

- virtual ~[OsclExecSchedulerCommonBase](#) ()
- void [InstallScheduler](#) ()
- void [UninstallScheduler](#) ()
- void [Error](#) (int32 anError) const
- [OsclExecSchedulerCommonBase](#) ([Oscl_DefAlloc](#) *)
- virtual void [ConstructL](#) (const char *name, int)
- void [BeginScheduling](#) (bool blocking, bool native)
- void [EndScheduling](#) ()
- void [BlockingLoopL](#) ()
- bool [IsStarted](#) ()
- bool [IsInstalled](#) ()
- void [AddToExecTimerQ](#) ([PVActiveBase](#) *active, uint32)
- void [PendComplete](#) ([PVActiveBase](#) *, int32 aReason, [TPVThreadContext](#) aContext)
- void [RequestCanceled](#) ([PVActiveBase](#) *)
- [PVActiveBase](#) * [UpdateTimers](#) (uint32 &aDelay)
- [PVActiveBase](#) * [UpdateTimersMsec](#) (uint32 &aDelay)
- [PVActiveBase](#) * [WaitForReadyAO](#) ()

- void [CallRunExec \(PVActiveBase *\)](#)
- void [ConstructStatQ \(\)](#)
- void [BeginStats \(\)](#)
- void [EndStats \(\)](#)
- void [CleanupStatQ \(\)](#)
- PVActiveBase * [FindPVBase \(PVActiveBase *active, OsclDoubleList< PVActiveBase > &\)](#)
- void [CleanupExecQ \(\)](#)
- void [InitExecQ \(int\)](#)
- void [ResetLogPerf \(\)](#)
- void [IncLogPerf \(uint32\)](#)

Static Protected Methods

- OsclExecSchedulerCommonBase * [GetScheduler \(\)](#)
- OsclExecSchedulerCommonBase * [SetScheduler \(OsclExecSchedulerCommonBase *\)](#)
- void [ShowStats \(PVActiveStats *active\)](#)
- void [ShowSummaryStats \(PVActiveStats *active, PVLogger *, int64, int64 &, float &\)](#)

Protected Attributes

- bool [iBlockingMode](#)
- bool [iNativeMode](#)
- PVSchedulerStopper * [iStopper](#)
- OsclNoYieldMutex [iStopperCrit](#)
- PVThreadContext [iThreadContext](#)
- OsclNameString< PVSCHEDNAMELEN > [iName](#)
- bool [iDoStop](#)
- bool [iDoSuspend](#)
- bool [iSuspended](#)
- OsclSemaphore [iResumeSem](#)
- OsclErrorTrapImp * [iErrorTrapImp](#)
- OsclReadyQ [iReadyQ](#)
- OsclTimerQ [iExecTimerQ](#)
- uint32 [iNumAOAdded](#)
- OsclDoubleList< PVActiveStats > [iPVStatQ](#)
- PVActiveStats * [iOtherExecStats \[EOtherExecStats_Last\]](#)
- uint8 * [iTTotalTicksTemp](#)
- int64 [iGrandTotalTicks](#)
- float [iTTotalPercent](#)
- uint32 [iTime](#)
- int32 [iDelta](#)
- PVActiveStats * [iPVStats](#)
- PVLogger * [iLogger](#)
- PVLogger * [iDebugLogger](#)
- char * [iLogPerfIndentStr](#)
- int32 [iLogPerfIndentStrLen](#)
- uint32 [iLogPerfTotal](#)
- Oscl_DefAlloc * [iAlloc](#)
- OsclMemAllocator [iDefAlloc](#)

Static Protected Attributes

- const uint32 [iTTimeCompareThreshold](#)

Friends

- class [OsclScheduler](#)
- class [PVThreadContext](#)
- class [OsclCoeActiveScheduler](#)
- class [OsclTimerCompare](#)
- class [OsclReadyQ](#)
- class [OsclError](#)
- class [PVActiveStats](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [PVActiveBase](#)
- class [PVSchedulerStopper](#)
- class [OsclExecScheduler](#)

6.133.1 Member Enumeration Documentation

6.133.1.1 enum OsclExecSchedulerCommonBase::TOtherExecStats [protected]

Enumeration values:

- EOtherExecStats_WaitTime**
- EOtherExecStats_QueueTime**
- EOtherExecStats_NativeOS**
- EOtherExecStats_ReleaseTime**
- EOtherExecStats_Last**

6.133.2 Constructor & Destructor Documentation

6.133.2.1 `virtual OsclExecSchedulerCommonBase::~OsclExecSchedulerCommonBase ()`
[protected, virtual]

6.133.2.2 `OsclExecSchedulerCommonBase::OsclExecSchedulerCommonBase (Oscl_DefAlloc *)`
[protected]

6.133.3 Member Function Documentation

6.133.3.1 `void OsclExecSchedulerCommonBase::AddToExecTimerQ (PVActiveBase * active, uint32) [protected]`

6.133.3.2 `void OsclExecSchedulerCommonBase::BeginScheduling (bool blocking, bool native)`
[protected]

6.133.3.3 `void OsclExecSchedulerCommonBase::BeginStats () [protected]`

6.133.3.4 `void OsclExecSchedulerCommonBase::BlockingLoopL () [protected]`

6.133.3.5 `void OsclExecSchedulerCommonBase::CallRunExec (PVActiveBase *) [protected]`

6.133.3.6 `void OsclExecSchedulerCommonBase::CleanupExecQ () [protected]`

6.133.3.7 `void OsclExecSchedulerCommonBase::CleanupStatQ () [protected]`

6.133.3.8 `virtual void OsclExecSchedulerCommonBase::ConstructL (const char * name, int)`
[protected, virtual]

6.133.3.9 `void OsclExecSchedulerCommonBase::ConstructStatQ () [protected]`

6.133.3.10 `void OsclExecSchedulerCommonBase::EndScheduling () [protected]`

6.133.3.11 `void OsclExecSchedulerCommonBase::EndStats () [protected]`

6.133.3.12 `void OsclExecSchedulerCommonBase::Error (int32 anError) const [protected]`

6.133.3.13 `PVActiveBase* OsclExecSchedulerCommonBase::FindPVBase (PVActiveBase * active, OsclDoubleList< PVActiveBase > &) [protected]`

6.133.3.14 `OSCL_IMPORT_REF uint32 OsclExecSchedulerCommonBase::GetId () [static]`

Get numeric ID of current thread.

6.133.3.15 `OSCL_IMPORT_REF OsclNameString< PVSCHEDNAMELEN >* OsclExecSchedulerCommonBase::GetName () [static]`

Get name of scheduler for current thread.

- 6.133.3.16** `OsclExecSchedulerCommonBase* OsclExecSchedulerCommonBase::GetScheduler ()`
 [static, protected]
- 6.133.3.17** `void OsclExecSchedulerCommonBase::IncLogPerf (uint32) [protected]`
- 6.133.3.18** `void OsclExecSchedulerCommonBase::InitExecQ (int) [protected]`
- 6.133.3.19** `void OsclExecSchedulerCommonBase::InstallScheduler () [protected]`
- 6.133.3.20** `bool OsclExecSchedulerCommonBase::IsInstalled () [inline, protected]`
- 6.133.3.21** `bool OsclExecSchedulerCommonBase::IsStarted () [protected]`
- 6.133.3.22** `void OsclExecSchedulerCommonBase::PendComplete (PVActiveBase *, int32 aReason, TPVThreadContext aContext) [protected]`
- 6.133.3.23** `void OsclExecSchedulerCommonBase::RequestCanceled (PVActiveBase *) [protected]`
- 6.133.3.24** `void OsclExecSchedulerCommonBase::ResetLogPerf () [protected]`
- 6.133.3.25** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::ResumeScheduler ()`

Resume scheduling immediately. This API only applies to a blocking loop scheduler.

- 6.133.3.26** `OsclExecSchedulerCommonBase* OsclExecSchedulerCommonBase::SetScheduler (OsclExecSchedulerCommonBase *) [static, protected]`
- 6.133.3.27** `void OsclExecSchedulerCommonBase::ShowStats (PVActiveStats * active) [static, protected]`
- 6.133.3.28** `void OsclExecSchedulerCommonBase::ShowSummaryStats (PVActiveStats * active, PVLogger *, int64, int64 &, float &) [static, protected]`
- 6.133.3.29** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::StartNativeScheduler ()`

Start the OS native scheduling loop. This is an alternative to the PV scheduling loop. To stop the native scheduler, use the StopScheduler API.

- 6.133.3.30** `OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::StartScheduler (OsclSemaphore * sem = NULL)`

Start scheduling. This call blocks until scheduler is stopped or an error occurs.

Parameters:

sem: optional startup semaphore. If provided, the scheduler will signal this semaphore when the startup has progressed to the point that it's safe to call StopScheduler or SuspendScheduler from another thread.

6.133.3.31 OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::StopScheduler ()

Stop scheduling. This API may be called from the scheduling thread or some other thread.

6.133.3.32 OSCL_IMPORT_REF void OsclExecSchedulerCommonBase::SuspendScheduler ()

Suspend scheduling when the current Run is complete. This API only applies to a blocking loop scheduler.

6.133.3.33 void OsclExecSchedulerCommonBase::UninstallScheduler () [protected]**6.133.3.34 PVActiveBase* OsclExecSchedulerCommonBase::UpdateTimers (uint32 & aDelay) [protected]****6.133.3.35 PVActiveBase* OsclExecSchedulerCommonBase::UpdateTimersMsec (uint32 & aDelay) [protected]****6.133.3.36 PVActiveBase* OsclExecSchedulerCommonBase::WaitForReadyAO () [protected]****6.133.4 Friends And Related Function Documentation****6.133.4.1 friend class OsclActiveObject [friend]****6.133.4.2 friend class OsclCoeActiveScheduler [friend]****6.133.4.3 friend class OsclError [friend]****6.133.4.4 friend class OsclExecScheduler [friend]****6.133.4.5 friend class OsclReadyQ [friend]****6.133.4.6 friend class OsclScheduler [friend]**

Reimplemented in [OsclExecScheduler](#).

- 6.133.4.7 friend class OsclTimerCompare [friend]
- 6.133.4.8 friend class OsclTimerObject [friend]
- 6.133.4.9 friend class PVActiveBase [friend]
- 6.133.4.10 friend class PVActiveStats [friend]
- 6.133.4.11 friend class PVSchedulerStopper [friend]
- 6.133.4.12 friend class PVThreadContext [friend]

6.133.5 Field Documentation

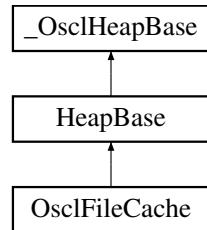
- 6.133.5.1 **Oscl_DefAlloc*** OsclExecSchedulerCommonBase::iAlloc [protected]
- 6.133.5.2 bool OsclExecSchedulerCommonBase::iBlockingMode [protected]
- 6.133.5.3 **PVLogger*** OsclExecSchedulerCommonBase::iDebugLogger [protected]
- 6.133.5.4 **OsclMemAllocator** OsclExecSchedulerCommonBase::iDefAlloc [protected]
- 6.133.5.5 int32 OsclExecSchedulerCommonBase::iDelta [protected]
- 6.133.5.6 bool OsclExecSchedulerCommonBase::iDoStop [protected]
- 6.133.5.7 bool OsclExecSchedulerCommonBase::iDoSuspend [protected]
- 6.133.5.8 **OsclErrorTrapImp*** OsclExecSchedulerCommonBase::iErrorTrapImp [protected]
- 6.133.5.9 **OsclTimerQ** OsclExecSchedulerCommonBase::iExecTimerQ [protected]
- 6.133.5.10 **int64** OsclExecSchedulerCommonBase::iGrandTotalTicks [protected]
- 6.133.5.11 **PVLogger*** OsclExecSchedulerCommonBase::iLogger [protected]
- 6.133.5.12 char* OsclExecSchedulerCommonBase::iLogPerfIndentStr [protected]
- 6.133.5.13 int32 OsclExecSchedulerCommonBase::iLogPerfIndentStrLen [protected]
- 6.133.5.14 uint32 OsclExecSchedulerCommonBase::iLogPerfTotal [protected]
- 6.133.5.15 **OsclNameString<PVSCHEDEXNAMELEN>** OsclExecSchedulerCommonBase::iName [protected]
- 6.133.5.16 bool OsclExecSchedulerCommonBase::iNativeMode [protected]
- 6.133.5.17 uint32 OsclExecSchedulerCommonBase::iNumAOAdded [protected]
- 6.133.5.18 **PVActiveStats*** OsclExecSchedulerCommonBase::iOtherExecStats[EOtherExecStats_-Last] [protected]
- 6.133.5.19 **OsclDoubleList<PVActiveStats>** OsclExecSchedulerCommonBase::iPVStatQ [protected]

- [oscl_scheduler.h](#)

6.134 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 \(void *outputBuffer, uint32 size, uint32 numelements\)](#)
- int32 [Write \(const void *inputBuffer, uint32 size, uint32 numelements\)](#)
- int32 [FileSize \(\)](#)
- int32 [Seek \(int32 offset, Oscl_File::seek_type origin\)](#)
- int32 [Tell \(\)](#)
- int32 [Flush \(\)](#)
- int32 [EndOfFile \(\)](#)

6.134.1 Constructor & Destructor Documentation

6.134.1.1 `OsclFileCache::OsclFileCache (Oscl_File & aContainer)`

6.134.1.2 `OsclFileCache::~OsclFileCache ()`

6.134.2 Member Function Documentation

6.134.2.1 `void OsclFileCache::Close ()`

6.134.2.2 `int32 OsclFileCache::EndOfFile () [inline]`

6.134.2.3 `int32 OsclFileCache::FileSize () [inline]`

6.134.2.4 `int32 OsclFileCache::Flush ()`

6.134.2.5 `int32 OsclFileCache::Open (uint32 mode, uint32 cache_size)`

6.134.2.6 `uint32 OsclFileCache::Read (void * outputBuffer, uint32 size, uint32 numelements)`

6.134.2.7 `int32 OsclFileCache::Seek (int32 offset, Oscl_File::seek_type origin)`

6.134.2.8 `int32 OsclFileCache::Tell () [inline]`

6.134.2.9 `int32 OsclFileCache::Write (const void * inputBuffer, uint32 size, uint32 numelements)`

The documentation for this class was generated from the following file:

- `oscl_file_cache.h`

6.135 OsclFileHandle Class Reference

```
#include <oscl_file_handle.h>
```

Public Methods

- [OsclFileHandle \(TOsclFileHandle aHandle\)](#)
- [OsclFileHandle \(const OsclFileHandle &aHandle\)](#)
- [TOsclFileHandle Handle \(\) const](#)

Friends

- class [Oscl_File](#)

6.135.1 Detailed Description

OsclFileHandle is a container for a handle to a previously-opened file.

6.135.2 Constructor & Destructor Documentation

6.135.2.1 OsclFileHandle::OsclFileHandle (TOsclFileHandle *aHandle*) [inline]

6.135.2.2 OsclFileHandle::OsclFileHandle (const OsclFileHandle & *aHandle*) [inline]

6.135.3 Member Function Documentation

6.135.3.1 TOsclFileHandle OsclFileHandle::Handle () const [inline]

6.135.4 Friends And Related Function Documentation

6.135.4.1 friend class Oscl_File [friend]

The documentation for this class was generated from the following file:

- [oscl_file_handle.h](#)

6.136 OsclFileStats Class Reference

```
#include <oscl_file_stats.h>
```

Public Methods

- [OsclFileStats \(Oscl_File *c\)](#)
- void [Start \(uint32 &aTicks\)](#)
- void [End \(TOsclFileOp aOp, uint32 aStart, uint32 aParam=0, uint32 aParam2=0\)](#)
- void [Log \(TOsclFileOp, PVLogger *, uint32\)](#)
- void [LogAll \(PVLogger *, uint32\)](#)

6.136.1 Constructor & Destructor Documentation

6.136.1.1 OsclFileStats::OsclFileStats ([Oscl_File](#) * *c*)

6.136.2 Member Function Documentation

6.136.2.1 void OsclFileStats::End ([TOsclFileOp](#) *aOp*, [uint32](#) *aStart*, [uint32](#) *aParam = 0*, [uint32](#) *aParam2 = 0*)

6.136.2.2 void OsclFileStats::Log ([TOsclFileOp](#), [PVLogger](#) *, [uint32](#))

6.136.2.3 void OsclFileStats::LogAll ([PVLogger](#) *, [uint32](#))

6.136.2.4 void OsclFileStats::Start ([uint32](#) & *aTicks*)

The documentation for this class was generated from the following file:

- [oscl_file_stats.h](#)

6.137 OsclFileStatsItem Class Reference

```
#include <oscl_file_stats.h>
```

Data Fields

- uint32 [iOpCount](#)
- uint32 [iParam](#)
- uint32 [iParam2](#)
- uint32 [iStartTick](#)
- uint32 [iTTotalTicks](#)

6.137.1 Field Documentation

6.137.1.1 uint32 OsclFileStatsItem::iOpCount

6.137.1.2 uint32 OsclFileStatsItem::iParam

6.137.1.3 uint32 OsclFileStatsItem::iParam2

6.137.1.4 uint32 OsclFileStatsItem::iStartTick

6.137.1.5 uint32 OsclFileStatsItem::iTTotalTicks

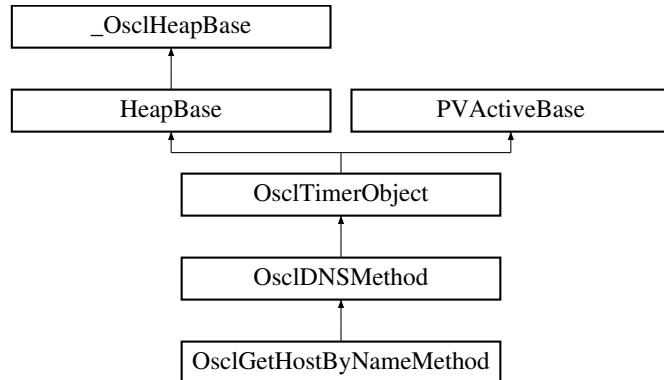
The documentation for this class was generated from the following file:

- [oscl_file_stats.h](#)

6.138 OsclGetHostByNameMethod Class Reference

```
#include <oscl_dns_gethostbyname.h>
```

Inheritance diagram for OsclGetHostByNameMethod::



Public Methods

- [~OsclGetHostByNameMethod \(\)](#)
- [TPVDNSEvent GetHostByName \(char *name, OsclNetworkAddress *addr, int32 aTimeout\)](#)

Static Public Methods

- [OsclGetHostByNameMethod * NewL \(Oscl_DefAlloc &a, OsclDNSI *aDNS, OsclDNSObserver *aObserver, uint32 aId\)](#)

6.138.1 Constructor & Destructor Documentation

6.138.1.1 OsclGetHostByNameMethod::~OsclGetHostByNameMethod ()

6.138.2 Member Function Documentation

6.138.2.1 [TPVDNSEvent OsclGetHostByNameMethod::GetHostByName \(char * name, OsclNetworkAddress * addr, int32 aTimeout\)](#)

6.138.2.2 [OsclGetHostByNameMethod* OsclGetHostByNameMethod::NewL \(Oscl_DefAlloc &a, OsclDNSI *aDNS, OsclDNSObserver *aObserver, uint32 aId\) \[static\]](#)

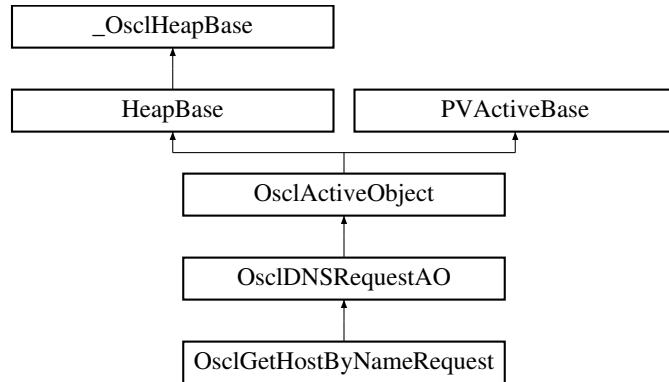
The documentation for this class was generated from the following file:

- [oscl_dns_gethostbyname.h](#)

6.139 OsclGetHostByNameRequest Class Reference

```
#include <oscl_dns_gethostbyname.h>
```

Inheritance diagram for OsclGetHostByNameRequest::



Friends

- class [OsclGetHostByNameMethod](#)

6.139.1 Friends And Related Function Documentation

6.139.1.1 friend class OsclGetHostByNameMethod [friend]

The documentation for this class was generated from the following file:

- [oscl_dns_gethostbyname.h](#)

6.140 OsclInit Class Reference

```
#include <oscl_init.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [Init](#) (int32 &aError, const [OsclSelect](#) *aSelect=NULL)
- OSCL_IMPORT_REF void [Cleanup](#) (int32 &aError, const [OsclSelect](#) *aSelect=NULL)

6.140.1 Detailed Description

Per-thread oscl initialization and cleanup.

6.140.2 Member Function Documentation

6.140.2.1 OSCL_IMPORT_REF void OsclInit::Cleanup (int32 & aError, const [OsclSelect](#) * aSelect = NULL) [static]

This routine cleans up the Oscl modules in the calling thread.

Parameters:

err: (output) error code of any leave that occurs in initialization.

config: (input param) optional set of initialization parameters. If null, then full initialization with default parameters will be performed. For proper cleanup, the parameters should match the ones used during the Init call.

6.140.2.2 OSCL_IMPORT_REF void OsclInit::Init (int32 & aError, const [OsclSelect](#) * aSelect = NULL) [static]

This routine initializes the Oscl modules in the calling thread.

Parameters:

err: (output) error code of any leave that occurs in initialization.

config: (input param) optional set of initialization parameters. If null, then full initialization with default parameters will be performed.

The documentation for this class was generated from the following file:

- [oscl_init.h](#)

6.141 OsclInteger64Transport Struct Reference

```
#include <oscl_int64_utils.h>
```

Data Fields

- uint32 [iHigh](#)
- uint32 [iLow](#)

6.141.1 Detailed Description

OsclInteger64Transport Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

6.141.2 Field Documentation

6.141.2.1 uint32 OsclInteger64Transport::iHigh

6.141.2.2 uint32 OsclInteger64Transport::iLow

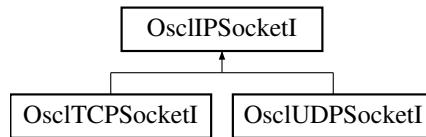
The documentation for this struct was generated from the following file:

- [oscl_int64_utils.h](#)

6.142 OsclIPSocketI Class Reference

```
#include <oscl_ip_socket.h>
```

Inheritance diagram for OsclIPSocketI::



Public Methods

- int32 [Bind \(OsclNetworkAddress &aAddress\)](#)
- int32 [Join \(OsclNetworkAddress &aAddress\)](#)
- int32 [SetRecvBufferSize \(uint32 size\)](#)
- virtual int32 [Close \(\)=0](#)
- virtual uint8 * [GetRecvData \(int32 *aLength\)=0](#)
- virtual uint8 * [GetSendData \(int32 *aLength\)=0](#)
- virtual ~[OsclIPSocketI \(\)](#)
- [OsclSocketServI * SocketServ \(\)](#)
- [Oscl_DefAlloc & Alloc \(\)](#)

Protected Methods

- [OsclIPSocketI \(Oscl_DefAlloc &a\)](#)
- void [ConstructL \(OsclSocketObserver *aObs, OsclSocketI *aSock, OsclSocketServI *aServ, uint32 aId\)](#)

Protected Attributes

- [Oscl_DefAlloc & iAlloc](#)
- [OsclNetworkAddress iAddress](#)
- uint32 [iId](#)
- [OsclSocketObserver * iObserver](#)
- [OsclSocketI * iSocket](#)
- [OsclSocketServI * iSocketServ](#)
- [PVLogger * iLogger](#)

Friends

- class [OsclSocketRequestAO](#)
- class [OsclSocketMethod](#)

6.142.1 Constructor & Destructor Documentation

6.142.1.1 `virtual OsclIPSocketI::~OsclIPSocketI () [inline, virtual]`

6.142.1.2 `OsclIPSocketI::OsclIPSocketI (Oscl_DefAlloc & a) [inline, protected]`

6.142.2 Member Function Documentation

6.142.2.1 `Oscl_DefAlloc& OsclIPSocketI::Alloc () [inline]`

6.142.2.2 `int32 OsclIPSocketI::Bind (OsclNetworkAddress & aAddress)`

6.142.2.3 `virtual int32 OsclIPSocketI::Close () [pure virtual]`

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

6.142.2.4 `void OsclIPSocketI::ConstructL (OsclSocketObserver * aObs, OsclSocketI * aSock, OsclSocketServI * aServ, uint32 aId) [protected]`

6.142.2.5 `virtual uint8* OsclIPSocketI::GetRecvData (int32 * aLength) [pure virtual]`

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

6.142.2.6 `virtual uint8* OsclIPSocketI::GetSendData (int32 * aLength) [pure virtual]`

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

6.142.2.7 int32 OsclIPSocketI::Join ([OsclNetworkAddress](#) & *aAddress*)

6.142.2.8 int32 OsclIPSocketI::SetRecvBufferSize (uint32 *size*)

6.142.2.9 [OsclSocketServI](#)* OsclIPSocketI::SocketServ () [inline]

6.142.3 Friends And Related Function Documentation

6.142.3.1 friend class OsclSocketMethod [friend]

6.142.3.2 friend class OsclSocketRequestAO [friend]

6.142.4 Field Documentation

6.142.4.1 [OsclNetworkAddress](#) OsclIPSocketI::iAddress [protected]

6.142.4.2 [Oscl_DefAlloc](#)& OsclIPSocketI::iAlloc [protected]

6.142.4.3 uint32 OsclIPSocketI::iId [protected]

6.142.4.4 [PVLogger](#)* OsclIPSocketI::iLogger [protected]

6.142.4.5 [OsclSocketObserver](#)* OsclIPSocketI::iObserver [protected]

6.142.4.6 [OsclSocketI](#)* OsclIPSocketI::iSocket [protected]

6.142.4.7 [OsclSocketServI](#)* OsclIPSocketI::iSocketServ [protected]

The documentation for this class was generated from the following file:

- [oscl_ip_socket.h](#)

6.143 OsclJump Class Reference

```
#include <oscl_error_imp_jumps.h>
```

Public Methods

- void [Jump](#) (int a)
- jmp_buf * [Top](#) ()
- [~OsclJump](#) ()

Static Public Methods

- OSCL_IMPORT_REF void [StaticJump](#) (int a)

Friends

- class [OsclErrorTrapImp](#)

6.143.1 Constructor & Destructor Documentation

6.143.1.1 OsclJump::~OsclJump () [inline]

6.143.2 Member Function Documentation

6.143.2.1 void OsclJump::Jump (int a) [inline]

6.143.2.2 OSCL_IMPORT_REF void OsclJump::StaticJump (int a) [static]

6.143.2.3 jmp_buf* OsclJump::Top () [inline]

6.143.3 Friends And Related Function Documentation

6.143.3.1 friend class OsclErrorTrapImp [friend]

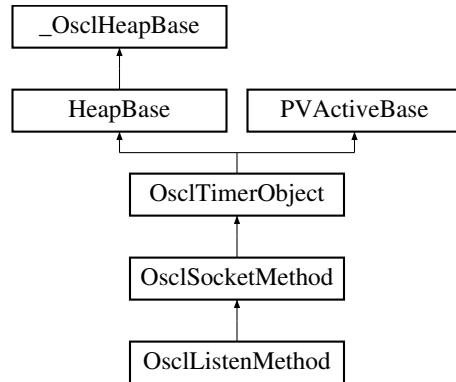
The documentation for this class was generated from the following file:

- [oscl_error_imp_jumps.h](#)

6.144 OsclListenMethod Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OsclListenMethod::



Public Methods

- [~OsclListenMethod \(\)](#)
- [TPVSocketEvent Listen \(uint32 qsize, int32 aTimeout\)](#)
- [OsclListenRequest * ListenRequest \(\)](#)

Static Public Methods

- [OsclListenMethod * NewL \(OsclIPSocketI &c\)](#)

6.144.1 Constructor & Destructor Documentation

6.144.1.1 OsclListenMethod::~OsclListenMethod ()

6.144.2 Member Function Documentation

6.144.2.1 TPVSocketEvent OsclListenMethod::Listen (uint32 qsize, int32 aTimeout)

6.144.2.2 OsclListenRequest* OsclListenMethod::ListenRequest () [inline]

6.144.2.3 OsclListenMethod* OsclListenMethod::NewL (OsclIPSocketI &c) [static]

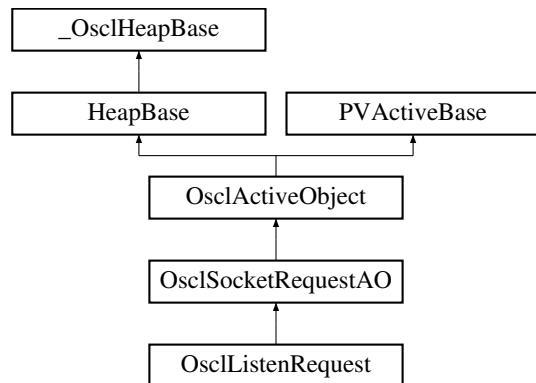
The documentation for this class was generated from the following file:

- [oscl_socket_listen.h](#)

6.145 OsclListenRequest Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OsclListenRequest::



Public Methods

- [OsclListenRequest \(OsclSocketMethod &c\)](#)
- void [Listen \(uint32 qsize\)](#)

6.145.1 Detailed Description

This is the AO that interacts with the socket server

6.145.2 Constructor & Destructor Documentation

6.145.2.1 OsclListenRequest::OsclListenRequest ([OsclSocketMethod & c](#)) [inline]

6.145.3 Member Function Documentation

6.145.3.1 void OsclListenRequest::Listen (uint32 *qsize*)

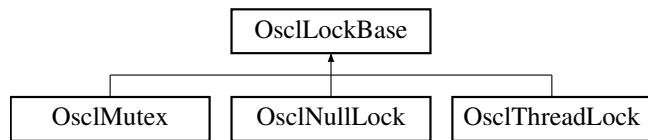
The documentation for this class was generated from the following file:

- [oscl_socket_listen.h](#)

6.146 OsclLockBase Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclLockBase::



Public Methods

- virtual void [Lock \(\)=0](#)
- virtual void [Unlock \(\)=0](#)
- virtual [~OsclLockBase \(\)](#)

6.146.1 Constructor & Destructor Documentation

6.146.1.1 virtual OsclLockBase::~OsclLockBase () [inline, virtual]

6.146.2 Member Function Documentation

6.146.2.1 virtual void OsclLockBase::Lock () [pure virtual]

Implemented in [OsclNullLock](#), [OsclMutex](#), and [OsclThreadLock](#).

6.146.2.2 virtual void OsclLockBase::Unlock () [pure virtual]

Implemented in [OsclNullLock](#), [OsclMutex](#), and [OsclThreadLock](#).

The documentation for this class was generated from the following file:

- [oscl_lock_base.h](#)

6.147 OsclMem Class Reference

```
#include <oscl_mem.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [Init \(\)](#)
- OSCL_IMPORT_REF void [Cleanup \(\)](#)

6.147.1 Member Function Documentation

6.147.1.1 OSCL_IMPORT_REF void OsclMem::Cleanup () [static]

Per-thread cleanup of Oscl Memory @exception: Leaves on error;

6.147.1.2 OSCL_IMPORT_REF void OsclMem::Init () [static]

Per-thread initialization of Oscl Memory

Parameters:

lock: A lock class for use with multi-threaded applications. The lock is needed in use cases where memory may be allocated in one thread and freed in another. In this case, there must be a single lock object, and its pointer must be passed to the [OsclMem::Init](#) call in each thread. If no lock is provided, the memory manager will not be thread-safe. @exception: Leaves on error

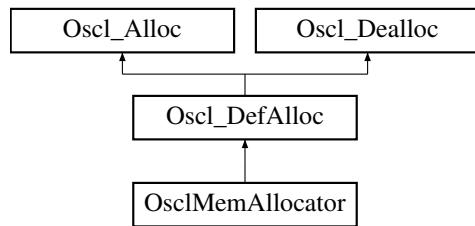
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

6.148 OsclMemAllocator Class Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemAllocator::



Public Methods

- [OsclAny * allocate \(const uint32 n\)](#)
- [OsclAny * allocate_fl \(const uint32 n, const char *file_name, const int line_num\)](#)
- void [deallocate \(OsclAny *p\)](#)

6.148.1 Detailed Description

A simple allocator class. Configurable as to whether this goes through the memory manager or not.

6.148.2 Member Function Documentation

6.148.2.1 [OsclAny* OsclMemAllocator::allocate \(const uint32 n\)](#) [inline, virtual]

This API throws an exception when malloc returns NULL. n must be greater than 0.

Returns:

pointer (or Leave with OsclErrNoMemory)

Implements [Oscl_DefAlloc](#).

6.148.2.2 [OsclAny* OsclMemAllocator::allocate_fl \(const uint32 n, const char *file_name, const int line_num\)](#) [inline, virtual]

Reimplemented from [Oscl_DefAlloc](#).

6.148.2.3 void [OsclMemAllocator::deallocate \(OsclAny *p\)](#) [inline, virtual]

Implements [Oscl_DefAlloc](#).

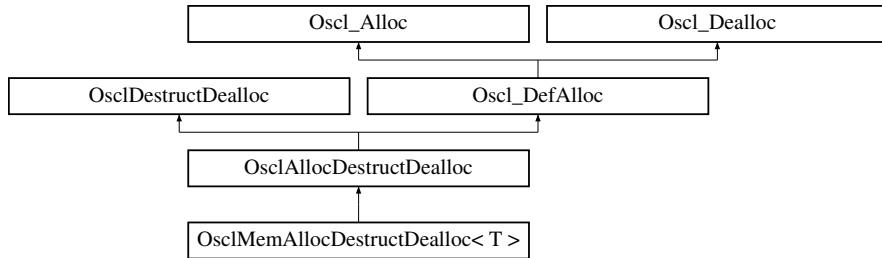
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

6.149 OsclMemAllocDestructDealloc< T > Class Template Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemAllocDestructDealloc< T >::



Public Methods

- [OsclAny * allocate_f1](#) (const uint32 size, const char *file_name, const int line_num)
- [OsclAny * allocate](#) (const uint32 size)
- void [deallocate](#) (OsclAny *p)
- void [destruct_and_dealloc](#) (OsclAny *p)

6.149.1 Detailed Description

`template<class T> class OsclMemAllocDestructDealloc< T >`

An [OsclAllocDestructDealloc](#) class that uses [OsclMemAllocator](#).

6.149.2 Member Function Documentation

6.149.2.1 template<class T> [OsclAny*](#) OsclMemAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]

Implements [Oscl_DefAlloc](#).

6.149.2.2 template<class T> [OsclAny*](#) OsclMemAllocDestructDealloc< T >::allocate_f1 (const uint32 size, const char *file_name, const int line_num) [inline, virtual]

Reimplemented from [Oscl_DefAlloc](#).

6.149.2.3 template<class T> void OsclMemAllocDestructDealloc< T >::deallocate ([OsclAny](#) * p) [inline, virtual]

Implements [Oscl_DefAlloc](#).

**6.149.2.4 template<class T> void OsclMemAllocDestructDealloc< T >::destruct_and_dealloc
(OsclAny * p) [inline, virtual]**

Implements [OsclDestructDealloc](#).

The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

6.150 OsclMemAudit Class Reference

```
#include <oscl_mem_audit.h>
```

Public Methods

- [OsclMemAudit \(\)](#)
- [~OsclMemAudit \(\)](#)
- [void * MM_allocate \(const OsclMemStatsNode *statsNode, uint32 sizeIn, const char *pFileName, uint32 lineNumber, bool allocNodeTracking=false\)](#)
- [bool MM_deallocate \(void *pMemBlockIn\)](#)
- [MM_Stats_t * MM_GetStats \(const char *const tagIn\)](#)
- [uint32 MM_GetStatsInDepth \(const char *tagIn, MM_Stats_CB *array_ptr, uint32 max_nodes\)](#)
- [uint32 MM_GetTreeNodes \(const char *tagIn\)](#)
- [bool MM_AddTag \(const char *tagIn\)](#)
- [const OsclMemStatsNode * MM_GetTagName \(const char *tagIn\)](#)
- [const OsclMemStatsNode * MM_GetExistingTag \(const char *tagIn\)](#)
- [const OsclMemStatsNode * MM_GetRootNode \(\)](#)
- [uint32 MM_GetAllocNodeInfo \(MM_AllocQueryInfo *output_array, uint32 max_array_size, uint32 offset\)](#)
- [MM_AllocQueryInfo * MM_CreateAllocNodeInfo \(uint32 max_array_size\)](#)
- [void MM_ReleaseAllocNodeInfo \(MM_AllocQueryInfo *info\)](#)
- [bool MM_Validate \(const void *ptrIn\)](#)
- [uint32 MM_GetAllocNo \(void\)](#)
- [void MM_GetOverheadStats \(MM_AuditOverheadStats &stats\)](#)
- [uint32 MM_GetNumAllocNodes \(\)](#)
- [uint32 MM_GetMode \(void\)](#)
- [uint8 MM_GetPrefillPattern \(void\)](#)
- [uint32 MM_GetPostfillPattern \(void\)](#)
- [void MM_SetMode \(uint32 inMode\)](#)
- [void MM_SetPrefillPattern \(uint8 pattern\)](#)
- [void MM_SetPostfillPattern \(uint8 pattern\)](#)
- [void MM_SetTagLevel \(uint32 level\)](#)
- [bool MM_SetFailurePoint \(const char *tagIn, uint32 alloc_number\)](#)
- [void MM_UnsetFailurePoint \(const char *tagIn\)](#)
- [int32 MM_GetRefCount \(\)](#)
- [OsclLockBase * GetLock \(\)](#)

Friends

- class [OsclMemGlobalAuditObject](#)

6.150.1 Constructor & Destructor Documentation

6.150.1.1 OsclMemAudit::OsclMemAudit () [inline]

Constructor, create the root node in statistics table

6.150.1.2 OsclMemAudit::~OsclMemAudit () [inline]

A destructor, remove all the nodes in allocation andstatistics table

6.150.2 Member Function Documentation**6.150.2.1 OsclLockBase* OsclMemAudit::GetLock () [inline]**

API to obtain mem lock ptr

6.150.2.2 bool OsclMemAudit::MM_AddTag (const char * tagIn) [inline]

API to add a node and zero out its counters; Note that this tag should be re-used

Parameters:

tagIn input tag

Returns:

true if operation succeeds;

6.150.2.3 void* OsclMemAudit::MM_allocate (const OsclMemStatsNode * statsNode, uint32 sizeIn, const char * pFileName, uint32 lineNumber, bool allocNodeTracking = false) [inline]

The following are APIs to __nothrow_ / const __nothrow_

Returns:

the memory pointer if operation succeeds.

6.150.2.4 MM_AllocQueryInfo* OsclMemAudit::MM_CreateAllocNodeInfo (uint32 max_array_size) [inline]**6.150.2.5 bool OsclMemAudit::MM_deallocate (void * pMemBlockIn) [inline]****Returns:**

true if operation succeeds;

6.150.2.6 uint32 OsclMemAudit::MM_GetAllocNo (void) [inline]

API to get the current allocation number

Returns:

the current allocation number

6.150.2.7 `uint32 OsclMemAudit::MM_GetAllocNodeInfo (MM_AllocQueryInfo * output_array, uint32 max_array_size, uint32 offset) [inline]`

API to query the list of alloc nodes. It copies the information into the provided output array.

Parameters:

output_array the array where the data will be written

max_array_size the max number of output array elements

offset the offset into the alloc node list from which the data should begin.

Returns:

the number of valid nodes in the output array

6.150.2.8 `const OsclMemStatsNode* OsclMemAudit::MM_GetExistingTag (const char * tagIn) [inline]`

API to add a node and zero out its counters; Note that this tag should be re-used

Parameters:

tagIn input tag

Returns:

true if operation succeeds;

6.150.2.9 `uint32 OsclMemAudit::MM_GetMode (void) [inline]`

API to get the operating mode of the mm_audit class.

6.150.2.10 `uint32 OsclMemAudit::MM_GetNumAllocNodes () [inline]`

API to get the number of allocation nodes (records) for allocations that are being tracked individually.

6.150.2.11 `void OsclMemAudit::MM_GetOverheadStats (MM_AuditOverheadStats & stats) [inline]`

API to get the overhead statistics for the memory used by the mm_audit class.

6.150.2.12 `uint32 OsclMemAudit::MM_GetPostfillPattern (void) [inline]`

API to get the postfill pattern. The pattern is used to fill the memory before freeing it.

6.150.2.13 `uint8 OsclMemAudit::MM_GetPrefillPattern (void) [inline]`

API to get the prefill pattern. The pattern is used to fill the memory before returning it to the caller.

6.150.2.14 int32 OsclMemAudit::MM_GetRefCount () [inline]

6.150.2.15 const OsclMemStatsNode* OsclMemAudit::MM_GetRootNode () [inline]

6.150.2.16 MM_Stats_t* OsclMemAudit::MM_GetStats (const char *const tagIn) [inline]

API to get memory statistics through context string(tag)

Returns:

statistics pointer if operation succeeds

6.150.2.17 uint32 OsclMemAudit::MM_GetStatsInDepth (const char * tagIn, MM_Stats_CB * array_ptr, uint32 max_nodes) [inline]

API to get memory statistics in detail through context string(tag) including its subtree

Returns:

statistics pointer array and actual number of nodes if operation succeeds

6.150.2.18 const OsclMemStatsNode* OsclMemAudit::MM_GetTagName (const char * tagIn) [inline]

API to add a node and zero out its counters; Note that this tag should be re-used

Parameters:

tagIn input tag

Returns:

pointer to [OsclMemStatsNode](#) which should be passed to MM_allocate

6.150.2.19 uint32 OsclMemAudit::MM_GetTreeNodes (const char * tagIn) [inline]

API to get the number of tree nodes including the tag node and its subtree

Parameters:

tagIn input tag

Returns:

the number of tree nodes ; 0 means no tag node

6.150.2.20 void OsclMemAudit::MM_ReleaseAllocNodeInfo (MM_AllocQueryInfo * info) [inline]

6.150.2.21 bool OsclMemAudit::MM_SetFailurePoint (const char * tagIn, uint32 alloc_number) [inline]

API to insert allocation failure deterministically according to allocation number associated with tag

Parameters:

tagIn input tag
alloc_number allocation number associated with tag

Returns:

true if operation succeeds;

6.150.2.22 void OsclMemAudit::MM_SetMode (uint32 *inMode*) [inline]

API to set the operating mode of the mm_audit class.

6.150.2.23 void OsclMemAudit::MM_SetPostfillPattern (uint8 *pattern*) [inline]

API to set the postfill pattern.

6.150.2.24 void OsclMemAudit::MM_SetPrefillPattern (uint8 *pattern*) [inline]

API to set the prefill pattern.

6.150.2.25 void OsclMemAudit::MM_SetTagLevel (uint32 *level*) [inline]

API to set the maximum tag level,i.e. tag level for a.b.c.d = 4

Parameters:

level input tag level to be set

6.150.2.26 void OsclMemAudit::MM_UnsetFailurePoint (const char * *tagIn*) [inline]

API to cancel the allocation failure point associated with tag

Parameters:

tagIn input tag

6.150.2.27 bool OsclMemAudit::MM_Validate (const void * *ptrIn*) [inline]

API to check the input pointer is a valid pointer to a chunk of memory

Parameters:

ptrIn input pointer to be validated

Returns:

true if operation succeeds;

6.150.3 Friends And Related Function Documentation

6.150.3.1 friend class OsclMemGlobalAuditObject [friend]

The documentation for this class was generated from the following file:

- [oscl_mem_audit.h](#)

6.151 OSCLMemAutoPtr< T, _Allocator > Class Template Reference

The oscl_auto_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the oscl_auto_ptr expires, its destructor uses delete to free the memory.

```
#include <oscl_mem_auto_ptr.h>
```

Public Methods

- **OSCLMemAutoPtr** (T *inPtr=0)

Default constructor Initializes the pointer and takes ownership.
- **OSCLMemAutoPtr** (const OSCLMemAutoPtr< T > &_Y)

Copy constructor.
- **OSCLMemAutoPtr< T, _Allocator > & operator=** (const OSCLMemAutoPtr< T, _Allocator > &_Y)

Assignment operator from an another oscl_auto_ptr.
- **~OSCLMemAutoPtr** ()

Destructor.
- **T & operator *** () const

The indirection operator () accesses a value indirectly, through a pointer.*
- **T * operator ->** () const

The indirection operator (->) accesses a value indirectly, through a pointer.
- **void takeOwnership** (T *ptr)

The takeOwnership function assigns the value with ownership.
- **void allocate** (**oscl_memsize_t** size)
- **void setWithoutOwnership** (T *ptr)

The takeOwnership function assigns the value with ownership.
- **T * get** () const

get() method returns the pointer, currently owned by the class.
- **T * release** () const

release() method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

Static Public Methods

- **void deallocate** (T *ptr)

Data Fields

- bool [_Ownership](#)

6.151.1 Detailed Description

```
template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> class OSCLMemAuto-  
Ptr< T, _Allocator >
```

The oscl_auto_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the oscl_auto_ptr expires, its destructor uses delete to free the memory.

The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an oscl_auto_ptr object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The oscl_auto_ptr is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

6.151.2 Constructor & Destructor Documentation

```
6.151.2.1 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::OSCLMemAutoPtr (T * inPtr = 0) [inline,  
explicit]
```

Default constructor Initializes the pointer and takes ownership.

```
6.151.2.2 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::OSCLMemAutoPtr (const OSCLMemAutoPtr<  
T > & _Y) [inline]
```

Copy constructor.

Initializes the pointer and takes ownership from another oscl_auto_ptr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

```
6.151.2.3 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>>  
OSCLMemAutoPtr< T, _Allocator >::~OSCLMemAutoPtr () [inline]
```

Destructor.

The pointer is deleted in case this class still has ownership

6.151.3 Member Function Documentation

6.151.3.1 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, _Allocator >::allocate (oscl_memsize_t size) [inline]

6.151.3.2 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, _Allocator >::deallocate (T *ptr) [inline, static]

6.151.3.3 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> T* OSCLMemAutoPtr< T, _Allocator >::get () const [inline]

[get\(\)](#) method returns the pointer, currently owned by the class.

6.151.3.4 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> T& OSCLMemAutoPtr< T, _Allocator >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OSCLMemAutoPtr can be used like the regular pointer that it was initialized with.

6.151.3.5 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> T* OSCLMemAutoPtr< T, _Allocator >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OSCLMemAutoPtr can be used like the regular pointer that it was initialized with.

6.151.3.6 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> OSCLMemAutoPtr< T, _Allocator >& OSCLMemAutoPtr< T, _Allocator >::operator=(const OSCLMemAutoPtr< T, _Allocator > & _Y) [inline]

Assignment operator from an another oscl_auto_ptr.

Parameters:

_Y The value parameter should be another oscl_auto_ptr

Returns:

Returns a reference to this oscl_auto_ptr instance with pointer initialized.

Precondition:

The input class should be non-null and should point to a valid pointer.

This assignment operator initializes the class to the contents of the oscl_auto_ptr given as the input parameter. The ownership of the pointer is transferred.

6.151.3.7 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> T* OSCLMemAutoPtr< T, _Allocator >::release () const [inline]

[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.

6.151.3.8 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, _Allocator >::setWithoutOwnership (T *ptr) [inline]

The takeOwnership function assigns the value with ownership.

6.151.3.9 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> void OSCLMemAutoPtr< T, _Allocator >::takeOwnership (T *ptr) [inline]

The takeOwnership function assigns the value with ownership.

6.151.4 Field Documentation

6.151.4.1 template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> bool OSCLMemAutoPtr< T, _Allocator >::_Ownership

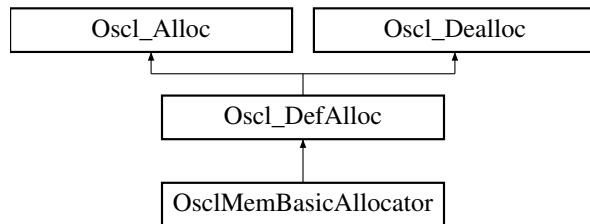
The documentation for this class was generated from the following file:

- [oscl_mem_auto_ptr.h](#)

6.152 OsclMemBasicAllocator Class Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemBasicAllocator::



Public Methods

- [OsclAny * allocate \(const uint32 n\)](#)
- [void deallocate \(OsclAny *p\)](#)

6.152.1 Detailed Description

A simple allocator class that does not use the memory management.

Note: this allocator is for internal use by Oscl only. Higher level code should use [OsclMemAllocator](#).

6.152.2 Member Function Documentation

6.152.2.1 [OsclAny* OsclMemBasicAllocator::allocate \(const uint32 n\) \[inline, virtual\]](#)

This API throws an exception when malloc returns NULL. n must be greater than 0.

Returns:

pointer (or Leave with OsclErrNoMemory)

Implements [Oscl_DefAlloc](#).

6.152.2.2 [void OsclMemBasicAllocator::deallocate \(OsclAny *p\) \[inline, virtual\]](#)

Implements [Oscl_DefAlloc](#).

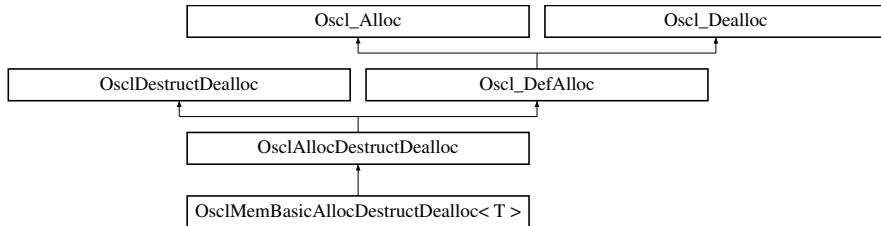
The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

6.153 OsclMemBasicAllocDestructDealloc< T > Class Template Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemBasicAllocDestructDealloc< T >::



Public Methods

- [OsclAny * allocate \(const uint32 size\)](#)
- [void deallocate \(OsclAny *p\)](#)
- [void destruct_and_dealloc \(OsclAny *p\)](#)

6.153.1 Detailed Description

template<class T> class OsclMemBasicAllocDestructDealloc< T >

An [OsclAllocDestructDealloc](#) class that uses [OsclMemBasicAllocator](#).

6.153.2 Member Function Documentation

6.153.2.1 template<class T> [OsclAny*](#) OsclMemBasicAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]

Implements [Oscl_DefAlloc](#).

6.153.2.2 template<class T> void OsclMemBasicAllocDestructDealloc< T >::deallocate ([OsclAny](#) * p) [inline, virtual]

Implements [Oscl_DefAlloc](#).

6.153.2.3 template<class T> void OsclMemBasicAllocDestructDealloc< T >::destruct_and_dealloc ([OsclAny](#) * p) [inline, virtual]

Implements [OsclDestructDealloc](#).

The documentation for this class was generated from the following file:

- [oscl_mem.h](#)

6.154 OsclMemGlobalAuditObject Class Reference

```
#include <oscl_mem.h>
```

Public Types

- `typedef OsclMemAudit audit_type`

Static Public Methods

- `OSCL_IMPORT_REF audit_type * getGlobalMemAuditObject ()`

Friends

- class `OsclMem`

6.154.1 Member Typedef Documentation

6.154.1.1 `typedef OsclMemAudit OsclMemGlobalAuditObject::audit_type`

6.154.2 Member Function Documentation

6.154.2.1 `OSCL_IMPORT_REF audit_type* OsclMemGlobalAuditObject::getGlobalMemAuditObject () [static]`

returns the global audit object. For use in macros only– not a public API.

6.154.3 Friends And Related Function Documentation

6.154.3.1 `friend class OsclMem [friend]`

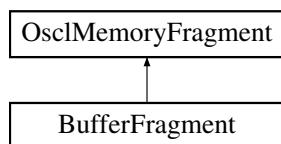
The documentation for this class was generated from the following file:

- `oscl_mem.h`

6.155 OsclMemoryFragment Struct Reference

```
#include <oscl_types.h>
```

Inheritance diagram for OsclMemoryFragment::



Data Fields

- uint32 **len**
- void * **ptr**

6.155.1 Field Documentation

6.155.1.1 uint32 OsclMemoryFragment::len

6.155.1.2 void* OsclMemoryFragment::ptr

The documentation for this struct was generated from the following file:

- [oscl_types.h](#)

6.156 OsclMemPoolAllocator Class Reference

```
#include <oscl_mempool_allocator.h>
```

Public Methods

- [OsclMemPoolAllocator \(Oscl_DefAlloc *gen_alloc=NULL\)](#)
- [virtual ~OsclMemPoolAllocator \(\)](#)
- [OsclAny * CreateMemPool \(const uint32 aNumChunk=2, const uint32 aChunkSize=4\)](#)
- [void DestroyMemPool \(\)](#)
- [uint oscl_mem_aligned_size \(uint size\)](#)

6.156.1 Constructor & Destructor Documentation

6.156.1.1 OsclMemPoolAllocator::OsclMemPoolAllocator ([Oscl_DefAlloc * gen_alloc = NULL](#))

6.156.1.2 virtual OsclMemPoolAllocator::~OsclMemPoolAllocator () [virtual]

6.156.2 Member Function Documentation

6.156.2.1 [OsclAny* OsclMemPoolAllocator::CreateMemPool \(const uint32 aNumChunk = 2, const uint32 aChunkSize = 4\)](#)

6.156.2.2 void OsclMemPoolAllocator::DestroyMemPool ()

6.156.2.3 [uint OsclMemPoolAllocator::oscl_mem_aligned_size \(uint size\)](#)

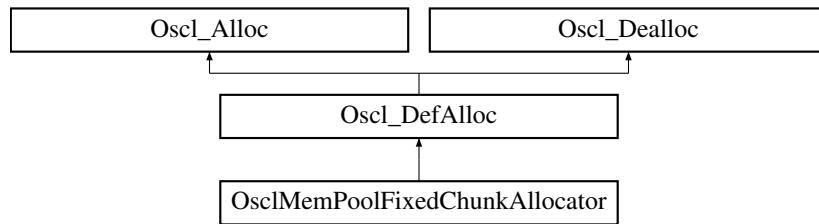
The documentation for this class was generated from the following file:

- [oscl_mempool_allocator.h](#)

6.157 OsclMemPoolFixedChunkAllocator Class Reference

```
#include <oscl_mem_mempool.h>
```

Inheritance diagram for OsclMemPoolFixedChunkAllocator::



Public Methods

- OSCL_IMPORT_REF OsclMemPoolFixedChunkAllocator (const uint32 numchunk=1, const uint32 chunksze=0, Oscl_DefAlloc *gen_alloc=NULL)
- virtual OSCL_IMPORT_REF void enablenullpointerreturn ()
- virtual OSCL_IMPORT_REF ~OsclMemPoolFixedChunkAllocator ()
- virtual OSCL_IMPORT_REF OsclAny * allocate (const uint32 n)
- virtual OSCL_IMPORT_REF void deallocate (OsclAny *p)
- virtual OSCL_IMPORT_REF void notifyfreechunkavailable (OsclMemPoolFixedChunkAllocatorObserver &obs, OsclAny *aContextData=NULL)
- virtual OSCL_IMPORT_REF void CancelFreeChunkAvailableCallback ()
- OSCL_IMPORT_REF void addRef ()
- OSCL_IMPORT_REF void removeRef ()

Protected Methods

- virtual OSCL_IMPORT_REF void createmempool ()
- virtual OSCL_IMPORT_REF void destroymempool ()

Protected Attributes

- uint32 iNumChunk
- uint32 iChunkSize
- uint32 iChunkSizeMemAligned
- Oscl_DefAlloc * iMemPoolAllocator
- OsclAny * iMemPool
- Oscl_Vector< OsclAny *, OsclMemAllocator > iFreeMemChunkList
- bool iCheckNextAvailableFreeChunk
- OsclMemPoolFixedChunkAllocatorObserver * iObserver
- OsclAny * iNextAvailableContextData
- int32 iRefCount
- bool iEnableNullPtrReturn

6.157.1 Constructor & Destructor Documentation

6.157.1.1 OSCL_IMPORT_REF OsclMemPoolFixedChunkAllocator::OsclMemPoolFixedChunkAllocator (const uint32 numchunk = 1, const uint32 chunkszie = 0, Oscl_DefAlloc * gen_alloc = NULL)

This API throws an exception when the memory allocation for pool fails If numchunk and chunkszie parameters are not set, memory pool of 1 chunk will be created in the first call to allocate. The chunk size will be set to the n passed in for [allocate\(\)](#). If numchunk parameter is set to 0, the memory pool will use 1 for numchunk.

Returns:

void

6.157.1.2 virtual OSCL_IMPORT_REF OsclMemPoolFixedChunkAllocator::~OsclMemPoolFixedChunkAllocator () [virtual]

The destructor for the memory pool

6.157.2 Member Function Documentation

6.157.2.1 OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::addRef ()

Increments the reference count for this memory pool allocator

Returns:

void

6.157.2.2 virtual OSCL_IMPORT_REF OsclAny* OsclMemPoolFixedChunkAllocator::allocate (const uint32 n) [virtual]

This API throws an exception when n is greater than the fixed chunk size or there are no free chunk available in the pool, if "enablenullpointerreturn" has not been called. If the memory pool hasn't been created yet, the pool will be created with chunk size equal to n so n must be greater than 0. Exception will be thrown if memory allocation for the memory pool fails.

Returns:

pointer to available chunk from memory pool

Implements [Oscl_DefAlloc](#).

6.157.2.3 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::CancelFreeChunkAvailableCallback () [virtual]

This API will cancel any past callback requests..

Returns:

void

6.157.2.4 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::createmempool() [protected, virtual]

**6.157.2.5 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::deallocate(
OsclAny *p)** [virtual]

This API throws an exception when the pointer p passed in is not part of the memory pool. Exception will be thrown if the memory pool is not set up yet.

Returns:

void

Implements [Oscl_DefAlloc](#).

6.157.2.6 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::destroymempool() [protected, virtual]

6.157.2.7 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::enablenullpointerreturn() [virtual]

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL_LEAVE\(OsclErrNoResources\)](#)" allocate API will return NULL.

Returns:

void

**6.157.2.8 virtual OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::notifyfreechunkavailable(
OsclMemPoolFixedChunkAllocatorObserver &obs, OsclAny *aContextData = NULL)** [virtual]

This API will set the flag to send a callback via specified observer object when the next memory chunk is deallocated by [deallocate\(\)](#) call..

Returns:

void

6.157.2.9 OSCL_IMPORT_REF void OsclMemPoolFixedChunkAllocator::removeRef()

Decrements the reference count for this memory pool allocator When the reference count goes to 0, this instance of the memory pool object is deleted

Returns:

void

6.157.3 Field Documentation

- 6.157.3.1 **bool OsclMemPoolFixedChunkAllocator::iCheckNextAvailableFreeChunk** [protected]
- 6.157.3.2 **uint32 OsclMemPoolFixedChunkAllocator::iChunkSize** [protected]
- 6.157.3.3 **uint32 OsclMemPoolFixedChunkAllocator::iChunkSizeMemAligned** [protected]
- 6.157.3.4 **bool OsclMemPoolFixedChunkAllocator::iEnableNullPtrReturn** [protected]
- 6.157.3.5 **Oscl_Vector<OsclAny*, OsclMemAllocator> OsclMemPoolFixedChunkAllocator::iFreeMemChunkList** [protected]
- 6.157.3.6 **OsclAny* OsclMemPoolFixedChunkAllocator::iMemPool** [protected]
- 6.157.3.7 **Oscl_DefAlloc* OsclMemPoolFixedChunkAllocator::iMemPoolAllocator** [protected]
- 6.157.3.8 **OsclAny* OsclMemPoolFixedChunkAllocator::iNextAvailableContextData** [protected]
- 6.157.3.9 **uint32 OsclMemPoolFixedChunkAllocator::iNumChunk** [protected]
- 6.157.3.10 **OsclMemPoolFixedChunkAllocatorObserver* OsclMemPoolFixedChunkAllocator::iObserver** [protected]
- 6.157.3.11 **int32 OsclMemPoolFixedChunkAllocator::iRefCount** [protected]

The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

6.158 OsclMemPoolFixedChunkAllocatorObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

Public Methods

- virtual void [freechunkavailable \(OsclAny *aContextData\)=0](#)
- virtual [~OsclMemPoolFixedChunkAllocatorObserver \(\)](#)

6.158.1 Constructor & Destructor Documentation

6.158.1.1 virtual [OsclMemPoolFixedChunkAllocatorObserver::~OsclMemPoolFixedChunkAllocatorObserver \(\) \[inline, virtual\]](#)

6.158.2 Member Function Documentation

6.158.2.1 virtual void [OsclMemPoolFixedChunkAllocatorObserver::freechunkavailable \(OsclAny * aContextData\) \[pure virtual\]](#)

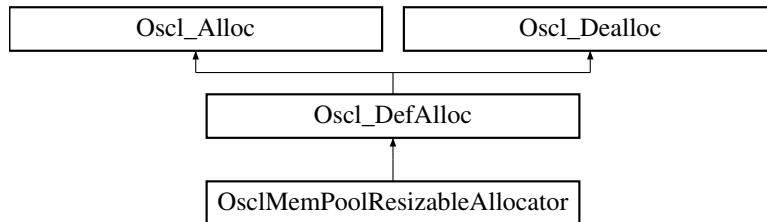
The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

6.159 OsclMemPoolResizableAllocator Class Reference

```
#include <oscl_mem_mempool.h>
```

Inheritance diagram for OsclMemPoolResizableAllocator::



Public Methods

- OSCL_IMPORT_REF [OsclMemPoolResizableAllocator](#) (uint32 aMemPoolBufferSize, uint32 aMemPoolBufferNumLimit=0, uint32 aExpectedNumBlocksPerBuffer=0, [Oscl_DefAlloc](#) *gen_alloc=NULL)
- virtual OSCL_IMPORT_REF void [enablenullpointerreturn](#) ()
- virtual OSCL_IMPORT_REF [OsclAny](#) * [allocate](#) (const uint32 aNumBytes)
- virtual OSCL_IMPORT_REF void [deallocate](#) ([OsclAny](#) *aPtr)
- virtual OSCL_IMPORT_REF bool [trim](#) ([OsclAny](#) *aPtr, uint32 aBytesToFree)
- OSCL_IMPORT_REF uint32 [getBufferSize](#) () const
- virtual OSCL_IMPORT_REF uint32 [getAllocatedSize](#) () const
- virtual OSCL_IMPORT_REF uint32 [getAvailableSize](#) () const
- virtual OSCL_IMPORT_REF uint32 [getLargestContiguousFreeBlockSize](#) () const
- virtual OSCL_IMPORT_REF bool [setMaxSzForNewMemPoolBuffer](#) (uint32 aMaxNewMemPoolBufferSz)
- virtual OSCL_IMPORT_REF void [notifyfreeblockavailable](#) ([OsclMemPoolResizableAllocatorObserver](#) &aObserver, uint32 aRequestedSize=0, [OsclAny](#) *aContextData=NULL)
- virtual OSCL_IMPORT_REF void [CancelFreeChunkAvailableCallback](#) ()
- virtual OSCL_IMPORT_REF void [notifyfreememoryavailable](#) ([OsclMemPoolResizableAllocatorMemoryObserver](#) &aObserver, uint32 aRequestedSize=0, [OsclAny](#) *aContextData=NULL)
- OSCL_IMPORT_REF void [CancelFreeMemoryAvailableCallback](#) ()
- OSCL_IMPORT_REF void [addRef](#) ()
- OSCL_IMPORT_REF void [removeRef](#) ()

Protected Methods

- virtual OSCL_IMPORT_REF ~[OsclMemPoolResizableAllocator](#) ()
- [MemPoolBufferInfo](#) * [addnewmempoolbuffer](#) (uint32 aBufferSize)
- void [destroyallmempoolbuffers](#) ()
- [MemPoolBlockInfo](#) * [findfreeblock](#) (uint32 aBlockSize)
- [OsclAny](#) * [allocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr, uint32 aNumBytes)
- void [deallocateblock](#) ([MemPoolBlockInfo](#) &aBlockPtr)
- bool [validateblock](#) ([OsclAny](#) *aBlockBufPtr)
- uint32 [getMemPoolBufferSize](#) ([MemPoolBufferInfo](#) *aBufferInfo) const
- uint32 [getMemPoolBufferAllocatedSize](#) ([MemPoolBufferInfo](#) *aBufferInfo) const
- uint32 [memoryPoolBufferMgmtOverhead](#) () const

Protected Attributes

- uint32 `iMemPoolBufferSize`
- uint32 `iMemPoolBufferNumLimit`
- uint32 `iExpectedNumBlocksPerBuffer`
- uint32 `iMaxNewMemPoolBufferSz`
- `Oscl_DefAlloc * iMemPoolBufferAllocator`
- `Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator > iMemPoolBufferList`
- uint32 `iBufferInfoAlignedSize`
- uint32 `iBlockInfoAlignedSize`
- bool `iCheckNextAvailable`
- uint32 `iRequestedNextAvailableSize`
- `OsclAny * iNextAvailableContextData`
- `OsclMemPoolResizableAllocatorObserver * iObserver`
- bool `iCheckFreeMemoryAvailable`
- uint32 `iRequestedAvailableFreeMemSize`
- `OsclAny * iFreeMemContextData`
- `OsclMemPoolResizableAllocatorMemoryObserver * iFreeMemPoolObserver`
- int32 `iRefCount`
- bool `iEnableNullPtrReturn`

6.159.1 Constructor & Destructor Documentation

6.159.1.1 OSCL_IMPORT_REF OsclMemPoolResizableAllocator::OsclMemPoolResizableAllocator (uint32 *aMemPoolBufferSize*, uint32 *aMemPoolBufferNumLimit* = 0, uint32 *aExpectedNumBlocksPerBuffer* = 0, `Oscl_DefAlloc * gen_alloc` = NULL)

Create the memory pool allocator with resizing functionality. The size of the memory pool buffer needs to be passed-in. The maximum number of memory pool buffers, expected number of blocks in a memory pool buffer, and outside allocator are optional. This API throws an exception when the memory allocation for the pool buffer fails. If memory pool buffer number limit parameter is not set, the assumption is that there is no limit and memory pool will grow as needed. If the expected number of blocks is not set or not known, the memory pool will use a default value to 10 to allocate extra memory for the block info header.

Returns:

`void`

6.159.1.2 virtual OSCL_IMPORT_REF OsclMemPoolResizableAllocator::~OsclMemPoolResizableAllocator () [protected, virtual]

The destructor for the memory pool. Should not be called directly. Use `removeRef()` instead.

6.159.2 Member Function Documentation

6.159.2.1 MemPoolBufferInfo* OsclMemPoolResizableAllocator::addnewmempoolbuffer (uint32 *aBufferSize*) [protected]

6.159.2.2 OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::addRef ()

Increments the reference count for this memory pool allocator

Returns:

void

**6.159.2.3 virtual OSCL_IMPORT_REF [OsclAny](#)* OsclMemPoolResizableAllocator::allocate
(const uint32 aNumBytes) [virtual]**

Allocates a block from the memory pool that is at least in size requested This API throws an exception if there isn't enough memory (if "enablenullpointerreturn" has not been called) for the requested amount in the pool or if the extra pool buffer cannot be allocated.

Returns:

Pointer to memory buffer from memory pool

Implements [Oscl_DefAlloc](#).

**6.159.2.4 [OsclAny](#)* OsclMemPoolResizableAllocator::allocateblock ([MemPoolBlockInfo](#) &
aBlockPtr, uint32 aNumBytes) [protected]****6.159.2.5 virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::CancelFree-
ChunkAvailableCallback () [virtual]**

This API will cancel any past callback requests..

Returns:

void

**6.159.2.6 OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::CancelFreeMemory-
AvailableCallback ()****6.159.2.7 virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::deallocate
([OsclAny](#) * aPtr) [virtual]**

Deallocates and returns a block back to the memory pool This API throws an exception if the pointer passed in is not part of the memory pool, aligned, or has corrupted block header.

Returns:

void

Implements [Oscl_DefAlloc](#).

**6.159.2.8 void OsclMemPoolResizableAllocator::deallocateblock ([MemPoolBlockInfo](#) &
aBlockPtr) [protected]****6.159.2.9 void OsclMemPoolResizableAllocator::destroyallmempoolbuffers () [protected]****6.159.2.10 virtual OSCL_IMPORT_REF void OsclMemPoolResizable-
Allocator::enablenullpointerreturn () [virtual]**

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "[OSCL_LEAVE\(OsclErrNoResources\)](#)" allocate API will return NULL.

Returns:`void`

6.159.2.11 `MemPoolBlockInfo* OsclMemPoolResizableAllocator::findfreeblock (uint32 aBlockSize) [protected]`

6.159.2.12 `virtual OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getAllocatedSize () [virtual]`

Returns the number of bytes allocated from the buffer<including the overhead bytes that may be allocated by the allocator to keep track of the chunks allocated>

6.159.2.13 `virtual OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getAvailableSize () [virtual]`

Returns the number of bytes available with the buffer

6.159.2.14 `OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getBufferSize ()`

Returns the size of the buffer <including the overhead bytes that may be allocated by the allocator>

6.159.2.15 `virtual OSCL_IMPORT_REF uint32 OsclMemPoolResizableAllocator::getLargestContiguousFreeBlockSize () [virtual]`

Returns the size of the largest available chunk in the memory.

6.159.2.16 `uint32 OsclMemPoolResizableAllocator::getMemPoolBufferAllocatedSize (MemPoolBufferInfo * aBufferInfo) const [protected]`

6.159.2.17 `uint32 OsclMemPoolResizableAllocator::getMemPoolBufferSize (MemPoolBufferInfo * aBufferInfo) const [protected]`

6.159.2.18 `uint32 OsclMemPoolResizableAllocator::memoryPoolBufferMgmtOverhead () [protected]`

6.159.2.19 `virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::notifyfreeblockavailable (OsclMemPoolResizableAllocatorObserver & aObserver, uint32 aRequestedSize = 0, OsclAny * aContextData = NULL) [virtual]`

This API will set the flag to send a callback via specified observer object when the next memory block is deallocated by `deallocate()` call. If the optional requested size parameter is set, the callback is sent when a free memory space of requested size becomes available. The optional context data is returned with the callback and can be used by the user to differentiate between different instances of memory pool objects. This memory pool only allows one notify to be queued. Another call to this function will just overwrite the previous call.

Returns:`void`

6.159.2.20 `virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::notifyfreememoryavailable (OsclMemPoolResizableAllocatorMemoryObserver & aObserver, uint32 aRequestedSize = 0, OsclAny * aContextData = NULL)`
[virtual]

6.159.2.21 `OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::removeRef ()`

Decrements the reference count for this memory pool allocator When the reference count goes to 0, this instance of the memory pool object is deleted

Returns:

`void`

6.159.2.22 `virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::setMaxSzForNewMemPoolBuffer (uint32 aMaxNewMemPoolBufferSz)`
[virtual]

6.159.2.23 `virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::trim (OsclAny * aPtr, uint32 aBytesToFree)` [virtual]

Returns a tail segment of a previously allocated memory block back to the memory pool. The passed-in pointer to the memory buffer is still valid after the call completes but the buffer size is smaller by by specified amount that was freed. This function allows the user to allocate a larger size block initially when the amount needed is unknown and then return the unused portion of the block when the amount becomes known. This API throws an exception if the pointer passed in is not part of the memory pool or the size to return is bigger than the size of the passed-in block. Exception will be thrown if the memory pool is not set up yet.

Returns:

`bool` True if trim operation successful. False if the block wasn't trimmed

6.159.2.24 **bool OsclMemPoolResizableAllocator::validateblock ([OsclAny](#) * *aBlockBufPtr*)** [protected]

6.159.3 Field Documentation

6.159.3.1 **uint32 OsclMemPoolResizableAllocator::iBlockInfoAlignedSize** [protected]

6.159.3.2 **uint32 OsclMemPoolResizableAllocator::iBufferInfoAlignedSize** [protected]

6.159.3.3 **bool OsclMemPoolResizableAllocator::iCheckFreeMemoryAvailable** [protected]

6.159.3.4 **bool OsclMemPoolResizableAllocator::iCheckNextAvailable** [protected]

6.159.3.5 **bool OsclMemPoolResizableAllocator::iEnableNullPtrReturn** [protected]

6.159.3.6 **uint32 OsclMemPoolResizableAllocator::iExpectedNumBlocksPerBuffer** [protected]

6.159.3.7 **[OsclAny](#)* OsclMemPoolResizableAllocator::iFreeMemContextData** [protected]

6.159.3.8 **[OsclMemPoolResizableAllocatorMemoryObserver](#)* OsclMemPoolResizableAllocator::iFreeMemPoolObserver** [protected]

6.159.3.9 **uint32 OsclMemPoolResizableAllocator::iMaxNewMemPoolBufferSz** [protected]

6.159.3.10 **[Oscl_DefAlloc](#)* OsclMemPoolResizableAllocator::iMemPoolBufferAllocator** [protected]

6.159.3.11 **[Oscl_Vector](#)<[MemPoolBufferInfo](#)*, [OsclMemAllocator](#)> OsclMemPoolResizableAllocator::iMemPoolBufferList** [protected]

6.159.3.12 **uint32 OsclMemPoolResizableAllocator::iMemPoolBufferNumLimit** [protected]

6.159.3.13 **uint32 OsclMemPoolResizableAllocator::iMemPoolBufferSize** [protected]

6.159.3.14 **[OsclAny](#)* OsclMemPoolResizableAllocator::iNextAvailableContextData** [protected]

6.159.3.15 **[OsclMemPoolResizableAllocatorObserver](#)* OsclMemPoolResizableAllocator::iObserver** [protected]

6.159.3.16 **int32 OsclMemPoolResizableAllocator::iRefCount** [protected]

6.159.3.17 **uint32 OsclMemPoolResizableAllocator::iRequestedAvailableFreeMemSize** [protected]

6.159.3.18 **uint32 OsclMemPoolResizableAllocator::iRequestedNextAvailableSize** [protected]

The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

6.160 OsclMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference

```
#include <oscl_mem_mempool.h>
```

Data Fields

- uint32 iBlockPreFence
- MemPoolBlockInfo * iNextFreeBlock
- MemPoolBlockInfo * iPrevFreeBlock
- uint32 iBlockSize
- uint8 * iBlockBuffer
- MemPoolBufferInfo * iParentBuffer
- uint32 iBlockPostFence

6.160.1 Field Documentation

6.160.1.1 uint8* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockBuffer

6.160.1.2 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPostFence

6.160.1.3 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPreFence

6.160.1.4 uint32 OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockSize

6.160.1.5 MemPoolBlockInfo* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iNextFree-Block

6.160.1.6 MemPoolBufferInfo* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iParent-Buffer

6.160.1.7 MemPoolBlockInfo* OsclMemPoolResizableAllocator::MemPoolBlockInfo::iPrevFree-Block

The documentation for this struct was generated from the following file:

- `oscl_mem_mempool.h`

6.161 OsclMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference

```
#include <oscl_mem_mempool.h>
```

Data Fields

- uint32 iBufferPreFence
- [OsclAny](#) * iStartAddr
- [OsclAny](#) * iEndAddr
- uint32 iBufferSize
- uint32 iNumOutstanding
- [MemPoolBlockInfo](#) * iNextFreeBlock
- uint32 iAllocatedSz
- uint32 iBufferPostFence

6.161.1 Field Documentation

6.161.1.1 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iAllocatedSz

6.161.1.2 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPostFence

6.161.1.3 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPreFence

6.161.1.4 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferSize

6.161.1.5 [OsclAny](#)* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iEndAddr

6.161.1.6 [MemPoolBlockInfo](#)* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iNextFree-Block

6.161.1.7 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iNumOutstanding

6.161.1.8 [OsclAny](#)* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iStartAddr

The documentation for this struct was generated from the following file:

- [oscl_mem_mempool.h](#)

6.162 OsclMemPoolResizableAllocatorMemoryObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

Public Methods

- virtual void [freememoryavailable \(OsclAny *aContextData\)=0](#)
- virtual [~OsclMemPoolResizableAllocatorMemoryObserver \(\)](#)

6.162.1 Constructor & Destructor Documentation

6.162.1.1 virtual OsclMemPoolResizableAllocatorMemoryObserver::~OsclMemPoolResizableAllocatorMemoryObserver () [inline, virtual]

6.162.2 Member Function Documentation

6.162.2.1 virtual void OsclMemPoolResizableAllocatorMemoryObserver::freememoryavailable (OsclAny * aContextData) [pure virtual]

The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

6.163 OsclMemPoolResizableAllocatorObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

Public Methods

- virtual void [freeblockavailable \(OsclAny *aContextData\)=0](#)
- virtual [~OsclMemPoolResizableAllocatorObserver \(\)](#)

6.163.1 Constructor & Destructor Documentation

6.163.1.1 [virtual OsclMemPoolResizableAllocatorObserver::~OsclMemPoolResizableAllocatorObserver \(\) \[inline, virtual\]](#)

6.163.2 Member Function Documentation

6.163.2.1 [virtual void OsclMemPoolResizableAllocatorObserver::freeblockavailable \(OsclAny **aContextData*\) \[pure virtual\]](#)

The documentation for this class was generated from the following file:

- [oscl_mem_mempool.h](#)

6.164 OsclMemStatsNode Class Reference

```
#include <oscl_mem_audit.h>
```

Public Methods

- [OsclMemStatsNode \(\)](#)
- [void reset \(\)](#)
- [~OsclMemStatsNode \(\)](#)
- [void * operator new \(oscl_memsize_t size\)](#)
- [void * operator new \(oscl_memsize_t size, OsclMemStatsNode *ptr\)](#)
- [void operator delete \(void *ptr\) throw \(\)](#)

Data Fields

- [MM_Stats_t * pMMStats](#)
- [MM_FailInsertParam * pMMFIParam](#)
- [char * tag](#)

6.164.1 Constructor & Destructor Documentation

6.164.1.1 OsclMemStatsNode::OsclMemStatsNode () [inline]

6.164.1.2 OsclMemStatsNode::~OsclMemStatsNode () [inline]

6.164.2 Member Function Documentation

6.164.2.1 void OsclMemStatsNode::operator delete (void *ptr) throw () [inline]

6.164.2.2 void* OsclMemStatsNode::operator new (oscl_memsize_t size, OsclMemStatsNode *ptr) [inline]

6.164.2.3 void* OsclMemStatsNode::operator new (oscl_memsize_t size) [inline]

6.164.2.4 void OsclMemStatsNode::reset () [inline]

6.164.3 Field Documentation

6.164.3.1 MM_FailInsertParam* OsclMemStatsNode::pMMFIParam

6.164.3.2 MM_Stats_t* OsclMemStatsNode::pMMStats

6.164.3.3 char* OsclMemStatsNode::tag

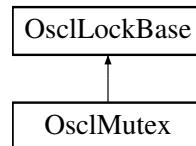
The documentation for this class was generated from the following file:

- [oscl_mem_audit.h](#)

6.165 OsclMutex Class Reference

```
#include <oscl_mutex.h>
```

Inheritance diagram for OsclMutex::



Public Methods

- OSCL_IMPORT_REF OsclMutex ()
- virtual OSCL_IMPORT_REF ~OsclMutex ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Create (void)
- OSCL_IMPORT_REF void Lock ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError TryLock ()
- OSCL_IMPORT_REF void Unlock ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Close (void)

6.165.1 Detailed Description

Class OsclMutex

6.165.2 Constructor & Destructor Documentation

6.165.2.1 OSCL_IMPORT_REF OsclMutex::OsclMutex ()

Class constructor

6.165.2.2 virtual OSCL_IMPORT_REF OsclMutex::~OsclMutex () [virtual]

Class destructor

6.165.3 Member Function Documentation

6.165.3.1 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclMutex::Close (void)

Closes the Mutex

Parameters:

It wont take any parameters

Returns:

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

6.165.3.2 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclMutex::Create (void)

Creates the Mutex

Parameters:

No input arguments

Returns:

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

6.165.3.3 OSCL_IMPORT_REF void OsclMutex::Lock () [virtual]

Locks the Mutex

Parameters:

It wont take any parameters

Returns:

Returns nothing

Implements [OsclLockBase](#).

6.165.3.4 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclMutex::TryLock ()

Try to lock the mutex,if the Mutex is already locked calling thread immediately returns with out blocking

Parameters:

It wont take any parameters

Returns:

Returns SUCCESS_ERROR if the mutex was acquired, MUTEX_LOCKED_ERROR if the mutex cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT_IMPLEMENTED.

6.165.3.5 OSCL_IMPORT_REF void OsclMutex::Unlock () [virtual]

Releases the Mutex

Parameters:

It wont take any parameters

Returns:

Returns nothing

Implements [OsclLockBase](#).

The documentation for this class was generated from the following file:

- [oscl_mutex.h](#)

6.166 OsclNameString< __len > Class Template Reference

```
#include <oscl_namestring.h>
```

Public Methods

- [OsclNameString \(\)](#)
- [OsclNameString \(const char a\[\]\)](#)
- [OsclNameString \(uint8 *a\)](#)
- void [Set \(uint8 *a\)](#)
- void [Set \(const char a\[\]\)](#)
- uint8 * [Str \(\) const](#)
- int32 [MaxLen \(\) const](#)

6.166.1 Detailed Description

```
template<int __len> class OsclNameString< __len >
```

Name string class appropriate for passing short constant ASCII strings around. All strings are automatically truncated and null-terminated.

6.166.2 Constructor & Destructor Documentation

6.166.2.1 template<int __len> OsclNameString< __len >::OsclNameString () [inline]

6.166.2.2 template<int __len> OsclNameString< __len >::OsclNameString (const char a[]) [inline]

6.166.2.3 template<int __len> OsclNameString< __len >::OsclNameString (uint8 * a) [inline]

6.166.3 Member Function Documentation

6.166.3.1 template<int __len> int32 OsclNameString< __len >::MaxLen () const [inline]

6.166.3.2 template<int __len> void OsclNameString< __len >::Set (const char a[]) [inline]

6.166.3.3 template<int __len> void OsclNameString< __len >::Set (uint8 * a) [inline]

Set the string to the input value. The string will be truncated to fit the storage class and automatically null-terminated.

Parameters:

a (input param): null-terminated character string.

6.166.3.4 template<int __len> uint8* OsclNameString< __len >::Str () const [inline]

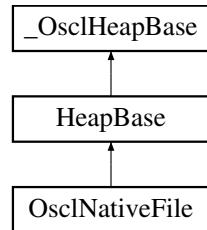
The documentation for this class was generated from the following file:

- [oscl_namestring.h](#)

6.167 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 \(int32 offset, Oscl_File::seek_type origin\)](#)
- [int32 Tell \(\)](#)
- [int32 Flush \(\)](#)
- [int32 EndOfFile \(\)](#)
- [int32 Size \(\)](#)
- [int32 Close \(\)](#)
- [uint32 Mode \(\)](#)
- [int32 GetError \(\)](#)
- [int32 ReadAsync \(OsclAny *buffer, uint32 size, uint32 numelements, OsclAOStatus &status\)](#)
- [uint32 GetReadAsyncNumElements \(\)](#)
- [bool HasAsyncRead \(\)](#)
- [void ReadAsyncCancel \(\)](#)

6.167.1 Constructor & Destructor Documentation

6.167.1.1 `OsclNativeFile::OsclNativeFile ()`

6.167.1.2 `OsclNativeFile::~OsclNativeFile ()`

6.167.2 Member Function Documentation

6.167.2.1 `int32 OsclNativeFile::Close ()`

6.167.2.2 `int32 OsclNativeFile::EndOfFile ()`

6.167.2.3 `int32 OsclNativeFile::Flush ()`

6.167.2.4 `int32 OsclNativeFile::GetError ()`

6.167.2.5 `uint32 OsclNativeFile::GetReadAsyncNumElements ()`

Get the number of elements read in the last call to ReadAsync. @returns: number of elements read.

6.167.2.6 `bool OsclNativeFile::HasAsyncRead ()`

@returns: true if async read is supported natively.

6.167.2.7 `uint32 OsclNativeFile::Mode () [inline]`

6.167.2.8 `int32 OsclNativeFile::Open (const char *filename, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv)`

6.167.2.9 `int32 OsclNativeFile::Open (const oscl_wchar *filename, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv)`

6.167.2.10 `int32 OsclNativeFile::Open (const OsclFileHandle &, uint32 mode, const OsclNativeFileParams ¶ms, Oscl_FileServer &fileserv)`

6.167.2.11 `uint32 OsclNativeFile::Read (OsclAny *buffer, uint32 size, uint32 numelements)`

6.167.2.12 `int32 OsclNativeFile::ReadAsync (OsclAny *buffer, uint32 size, uint32 numelements, OsclAOStatus &status)`

Asynchronous read.

Parameters:

buffer: data buffer, must be at least size*numelements bytes

size: size of elements

numelements: number of elements to read

status: Request status for asynchronous completion @returns: 0 for success.

6.167.2.13 void OsclNativeFile::ReadAsyncCancel ()

Cancel any pending async read.

6.167.2.14 int32 OsclNativeFile::Seek (int32 *offset*, **Oscl_File::seek_type *origin*)****6.167.2.15 int32 OsclNativeFile::Size ()****6.167.2.16 int32 OsclNativeFile::Tell ()****6.167.2.17 uint32 OsclNativeFile::Write (const **OsclAny** * *buffer*, uint32 *size*, uint32 *numelements*)**

The documentation for this class was generated from the following file:

- [oscl_file_native.h](#)

6.168 OsclNativeFileParams Class Reference

```
#include <oscl_file_types.h>
```

Public Methods

- [OsclNativeFileParams \(uint32 mode=0, uint32 bufsize=0, uint32 asyncsize=0\)](#)

Data Fields

- uint32 [iNativeAccessMode](#)
- uint32 [iNativeBufferSize](#)
- uint32 [iAsyncReadBufferSize](#)

6.168.1 Constructor & Destructor Documentation

6.168.1.1 OsclNativeFileParams::OsclNativeFileParams (uint32 mode = 0, uint32 bufsize = 0, uint32 asyncsize = 0) [inline]

6.168.2 Field Documentation

6.168.2.1 uint32 OsclNativeFileParams::iAsyncReadBufferSize

6.168.2.2 uint32 OsclNativeFileParams::iNativeAccessMode

6.168.2.3 uint32 OsclNativeFileParams::iNativeBufferSize

The documentation for this class was generated from the following file:

- [oscl_file_types.h](#)

6.169 OsclNetworkAddress Class Reference

```
#include <oscl_socket_types.h>
```

Public Methods

- [OsclNetworkAddress \(\)](#)
- [OsclNetworkAddress \(const char *addr, int p\)](#)
- [bool operator== \(const OsclNetworkAddress &rhs\) const](#)

Data Fields

- [OsclNameString< PVNETWORKADDRESS_LEN > ipAddr](#)
- [int port](#)

6.169.1 Constructor & Destructor Documentation

6.169.1.1 OsclNetworkAddress::OsclNetworkAddress () [inline]

6.169.1.2 OsclNetworkAddress::OsclNetworkAddress (const char * *addr*, int *p*) [inline]

6.169.2 Member Function Documentation

6.169.2.1 bool OsclNetworkAddress::operator== (const OsclNetworkAddress & *rhs*) const [inline]

6.169.3 Field Documentation

6.169.3.1 OsclNameString<PVNETWORKADDRESS_LEN> OsclNetworkAddress::ipAddr

6.169.3.2 int OsclNetworkAddress::port

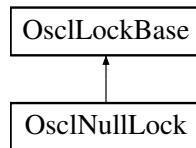
The documentation for this class was generated from the following file:

- [oscl_socket_types.h](#)

6.170 OsclNullLock Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclNullLock::



Public Methods

- virtual void [Lock \(\)](#)
- virtual void [Unlock \(\)](#)
- virtual [~OsclNullLock \(\)](#)

6.170.1 Constructor & Destructor Documentation

6.170.1.1 virtual OsclNullLock::~OsclNullLock () [inline, virtual]

6.170.2 Member Function Documentation

6.170.2.1 virtual void OsclNullLock::Lock () [inline, virtual]

Implements [OsclLockBase](#).

6.170.2.2 virtual void OsclNullLock::Unlock () [inline, virtual]

Implements [OsclLockBase](#).

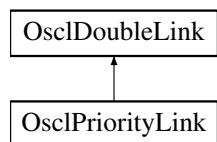
The documentation for this class was generated from the following file:

- [oscl_lock_base.h](#)

6.171 OsclPriorityLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclPriorityLink::



Data Fields

- int32 [iPriority](#)

6.171.1 Field Documentation

6.171.1.1 int32 OsclPriorityLink::iPriority

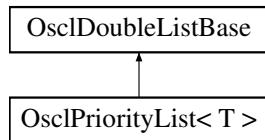
The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

6.172 OsclPriorityList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclPriorityList< T >::



Public Methods

- OSCL_INLINE OsclPriorityList ()
- OSCL_INLINE OsclPriorityList (int32 anOffset)
- OSCL_INLINE void Insert (T &aRef)
- OSCL_INLINE bool IsHead (const T *aPtr) const
- OSCL_INLINE bool IsTail (const T *aPtr) const
- OSCL_INLINE T * Head () const
- OSCL_INLINE T * Tail () const

```
template<class T> class OsclPriorityList< T >
```

6.172.1 Constructor & Destructor Documentation

6.172.1.1 template<class T> OSCL_INLINE OsclPriorityList< T >::OsclPriorityList ()

6.172.1.2 template<class T> OSCL_INLINE OsclPriorityList< T >::OsclPriorityList (int32
anOffset)

6.172.2 Member Function Documentation

6.172.2.1 template<class T> OSCL_INLINE T* OsclPriorityList< T >::Head ()

6.172.2.2 template<class T> OSCL_INLINE void OsclPriorityList< T >::Insert (T & aRef)

6.172.2.3 template<class T> OSCL_INLINE bool OsclPriorityList< T >::IsHead (const T * aPtr)
const

6.172.2.4 template<class T> OSCL_INLINE bool OsclPriorityList< T >::IsTail (const T * aPtr)
const

6.172.2.5 template<class T> OSCL_INLINE T* OsclPriorityList< T >::Tail ()

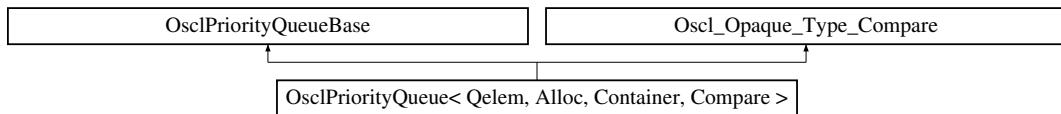
The documentation for this class was generated from the following file:

- [oscl_double_list.h](#)

6.173 OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference

```
#include <oscl_priqueue.h>
```

Inheritance diagram for OsclPriorityQueue< Qelem, Alloc, Container, Compare >::



Public Types

- `typedef Container::value_type value_type`
- `typedef Container container_type`
- `typedef Container::iterator iterator`
- `typedef Container::const_reference const_reference`

Public Methods

- `bool empty () const`
- `uint32 size () const`
- `void reserve (uint32 n)`
- `const_reference top () const`
- `const Container & vec ()`
- `void push (const value_type &input)`
- `void pop ()`
- `int remove (const value_type &input)`
- `OsclPriorityQueue ()`
- `virtual ~OsclPriorityQueue ()`

Protected Methods

- `void push_heap (iterator first, iterator last)`
- `void pop_heap (iterator first, iterator last)`
- `iterator find_heap (const value_type &input, iterator first, iterator last)`
- `int validate ()`
- `void swap (OsclAny *dest, const OsclAny *src)`
- `int compare_LT (OsclAny *a, OsclAny *b) const`
- `int compare_EQ (const OsclAny *a, const OsclAny *b) const`

Protected Attributes

- `Container c`
- `Compare comp`

Friends

- class [oscl_priqueue_test](#)

```
template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> class OsclPriorityQueue< Qelem, Alloc, Container, Compare >
```

6.173.1 Member Typedef Documentation

- 6.173.1.1 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> typedef Container::const_reference OsclPriorityQueue< Qelem, Alloc, Container, Compare >::const_reference
- 6.173.1.2 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> typedef Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::container_type
- 6.173.1.3 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> typedef Container::iterator OsclPriorityQueue< Qelem, Alloc, Container, Compare >::iterator
- 6.173.1.4 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> typedef Container::value_type OsclPriorityQueue< Qelem, Alloc, Container, Compare >::value_type

6.173.2 Constructor & Destructor Documentation

- 6.173.2.1 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> OsclPriorityQueue< Qelem, Alloc, Container, Compare >::OsclPriorityQueue () [inline]
- 6.173.2.2 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> virtual OsclPriorityQueue< Qelem, Alloc, Container, Compare >::~OsclPriorityQueue () [inline, virtual]

6.173.3 Member Function Documentation

- 6.173.3.1 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::compare_EQ (const [OsclAny](#) * a, const [OsclAny](#) * b) const [inline, protected, virtual]

Return a==b.

Implements [Oscl_Opaque_Type_Compare](#).

- 6.173.3.2 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::compare_LT ([OsclAny](#) * a, [OsclAny](#) * b) const [inline, protected, virtual]

Return a<b.

Implements [Oscl_Opaque_Type_Compare](#).

- 6.173.3.3 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> bool OsclPriorityQueue< Qelem, Alloc, Container, Compare >::empty () const [inline]`
- 6.173.3.4 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> iterator OsclPriorityQueue< Qelem, Alloc, Container, Compare >::find_heap (const value_type & input, iterator first, iterator last) [inline, protected]`
- 6.173.3.5 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::pop () [inline]`
- 6.173.3.6 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::pop_heap (iterator first, iterator last) [inline, protected]`
- 6.173.3.7 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::push (const value_type & input) [inline]`
- 6.173.3.8 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::push_heap (iterator first, iterator last) [inline, protected]`
- 6.173.3.9 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::remove (const value_type & input) [inline]`
- 6.173.3.10 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::reserve (uint32 n) [inline]`
- 6.173.3.11 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> uint32 OsclPriorityQueue< Qelem, Alloc, Container, Compare >::size () const [inline]`
- 6.173.3.12 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::swap (OsclAny * dest, const OsclAny * src) [inline, protected, virtual]`

Swap element at "a" with element at "b". Both pointers must be non-NULL.

Implements [Oscl_Opaque_Type_Compare](#).

6.173 OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference

- 6.173.3.13 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **const_reference** OsclPriorityQueue< Qelem, Alloc, Container, Compare >::top () const [inline]
- 6.173.3.14 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::validate () [inline, protected]
- 6.173.3.15 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> **const Container&** OsclPriorityQueue< Qelem, Alloc, Container, Compare >::vec () [inline]

6.173.4 Friends And Related Function Documentation

- 6.173.4.1 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> friend class oscl_priqueue_test [friend]

6.173.5 Field Documentation

- 6.173.5.1 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::c [protected]
- 6.173.5.2 template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Compare OsclPriorityQueue< Qelem, Alloc, Container, Compare >::comp [protected]

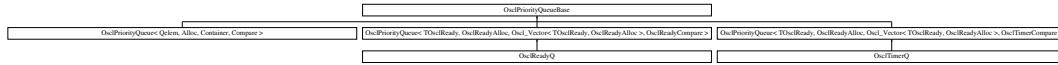
The documentation for this class was generated from the following file:

- [oscl_priqueue.h](#)

6.174 OsclPriorityQueueBase Class Reference

```
#include <oscl_pqueue.h>
```

Inheritance diagram for OsclPriorityQueueBase::



Protected Methods

- virtual ~OsclPriorityQueueBase ()
- OSCL_IMPORT_REF void [push_heap](#) (OsclAny *first, OsclAny *last)
- OSCL_IMPORT_REF void [pop_heap](#) (OsclAny *first, OsclAny *last)
- OSCL_IMPORT_REF OsclAny * [find_heap](#) (const OsclAny *input, OsclAny *first, OsclAny *last)
- OSCL_IMPORT_REF int [remove](#) (const OsclAny *input)
- void [construct](#) (Oscl_Opaque_Type_Compare *ot, Oscl_Vector_Base *vec)

6.174.1 Detailed Description

OsclPriorityQueueBase is a non-templatized base class for [OsclPriorityQueue](#). The purpose of this base class is to avoid large inline routines in the [OsclPriorityQueue](#) implementation. This class is not intended for direct instantiation except by [OsclPriorityQueue](#).

6.174.2 Constructor & Destructor Documentation

6.174.2.1 virtual OsclPriorityQueueBase::~OsclPriorityQueueBase () [inline, protected, virtual]

6.174.3 Member Function Documentation

6.174.3.1 void OsclPriorityQueueBase::construct (Oscl_Opaque_Type_Compare * ot, Oscl_Vector_Base * vec) [inline, protected]

6.174.3.2 OSCL_IMPORT_REF OsclAny* OsclPriorityQueueBase::find_heap (const OsclAny * input, OsclAny * first, OsclAny * last) [protected]

6.174.3.3 OSCL_IMPORT_REF void OsclPriorityQueueBase::pop_heap (OsclAny * first, OsclAny * last) [protected]

6.174.3.4 OSCL_IMPORT_REF void OsclPriorityQueueBase::push_heap (OsclAny * first, OsclAny * last) [protected]

6.174.3.5 OSCL_IMPORT_REF int OsclPriorityQueueBase::remove (const OsclAny * input) [protected]

The documentation for this class was generated from the following file:

- [oscl_pqueue.h](#)

6.175 OsclProcStatus Class Reference

```
#include <oscl_procstatus.h>
```

Public Types

- enum `eOsclProcError` { `SUCCESS_ERROR` = 0, `OTHER_ERROR`, `TOO_MANY_THREADS_ERROR`, `BAD_THREADID_ADDR_ERROR`, `MAX_THRDS_REACHED_ERROR`, `INVALID_THREAD_ID_ERROR`, `NOT_ENOUGH_MEMORY_ERROR`, `OUTOFMEMORY_ERROR`, `NOT_ENOUGH_RESOURCES_ERROR`, `THREAD_1_INACTIVE_ERROR`, `ALREADY_SUSPENDED_ERROR`, `NOT_SUSPENDED_ERROR`, `INVALID_THREAD_ERROR`, `INVALID_PARAM_ERROR`, `NO_PERMISSION_ERROR`, `INVALID_PRIORITY_ERROR`, `PSHARED_NOT_ZERO_ERROR`, `EXCEED_MAX_COUNT_VARIABLE_ERROR`, `THREAD_BLOCK_ERROR`, `EXCEED_MAX_SEM_COUNT_ERROR`, `INVALID_HANDLE_ERROR`, `INVALID_OPERATION_ERROR`, `INVALID_FUNCTION_ERROR`, `INVALID_ACCESS_ERROR`, `INVALID_ARGUMENT_ERROR`, `SYSTEM_RESOURCES_UNAVAILABLE_ERROR`, `INVALID_POINTER_ERROR`, `RELOCK_MUTEX_ERROR`, `THREAD_NOT_OWN_MUTEX_ERROR`, `MUTEX_LOCKED_ERROR`, `WAIT_ABANDONED_ERROR`, `WAIT_TIMEOUT_ERROR`, `SEM_NOT_SIGNALED_ERROR`, `PSHARED_ATTRIBUTE_SETTING_ERROR`, `NOT_IMPLEMENTED` }

6.175.1 Detailed Description

Class OsclProcStatus

6.175.2 Member Enumeration Documentation

6.175.2.1 enum OsclProcStatus::eOsclProcError

List of enums which contain error codes

Enumeration values:

`SUCCESS_ERROR`
`OTHER_ERROR`
`TOO_MANY_THREADS_ERROR`
`BAD_THREADID_ADDR_ERROR`
`MAX_THRDS_REACHED_ERROR`
`INVALID_THREAD_ID_ERROR`
`NOT_ENOUGH_MEMORY_ERROR`
`OUTOFMEMORY_ERROR`
`NOT_ENOUGH_RESOURCES_ERROR`
`THREAD_1_INACTIVE_ERROR`
`ALREADY_SUSPENDED_ERROR`
`NOT_SUSPENDED_ERROR`
`INVALID_THREAD_ERROR`
`INVALID_PARAM_ERROR`
`NO_PERMISSION_ERROR`

INVALID_PRIORITY_ERROR
PSHARED_NOT_ZERO_ERROR
EXCEED_MAX_COUNT_VARIABLE_ERROR
THREAD_BLOCK_ERROR
EXCEED_MAX_SEM_COUNT_ERROR
INVALID_HANDLE_ERROR
INVALID_OPERATION_ERROR
INVALID_FUNCTION_ERROR
INVALID_ACCESS_ERROR
INVALID_ARGUMENT_ERROR
SYSTEM_RESOURCES_UNAVAILABLE_ERROR
INVALID_POINTER_ERROR
RELOCK_MUTEX_ERROR
THREAD_NOT_OWN_MUTEX_ERROR
MUTEX_LOCKED_ERROR
WAIT_ABANDONED_ERROR
WAIT_TIMEOUT_ERROR
SEM_NOT_SIGNALED_ERROR
PSHARED_ATTRIBUTE_SETTING_ERROR
NOT_IMPLEMENTED

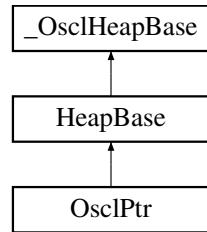
The documentation for this class was generated from the following file:

- [oscl_procstatus.h](#)

6.176 OsclPtr Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclPtr::



Public Methods

- [OsclPtr \(uint8 *ptr, int32 &len, int32 max\)](#)
- [OsclPtr \(const OsclPtr &d\)](#)
- [uint8 * Ptr \(\)](#)
- [void SetLength \(int32 l\)](#)
- [int32 Length \(\)](#)
- [void Zero \(\)](#)
- [void Set \(OsclPtr &v\)](#)
- [void Set \(uint8 *ptr, int32 len, int32 max\)](#)
- [void Append \(OsclPtrC &v\)](#)

6.176.1 Constructor & Destructor Documentation

6.176.1.1 OsclPtr::OsclPtr (uint8 *ptr, int32 &len, int32 max) [inline]

6.176.1.2 OsclPtr::OsclPtr (const OsclPtr &d) [inline]

6.176.2 Member Function Documentation

6.176.2.1 void OsclPtr::Append (OsclPtrC &v) [inline]

6.176.2.2 int32 OsclPtr::Length () [inline]

6.176.2.3 uint8* OsclPtr::Ptr () [inline]

6.176.2.4 void OsclPtr::Set (uint8 *ptr, int32 len, int32 max) [inline]

6.176.2.5 void OsclPtr::Set (OsclPtr &v) [inline]

6.176.2.6 void OsclPtr::SetLength (int32 l) [inline]

6.176.2.7 void OsclPtr::Zero () [inline]

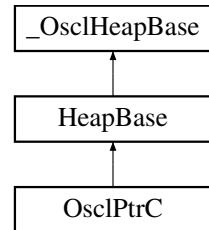
The documentation for this class was generated from the following file:

- [oscl_file_async_read.h](#)

6.177 OsclPtrC Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclPtrC::



Public Methods

- [OsclPtrC](#) (const uint8 *ptr, int32 len, int32 max)
- [OsclPtrC](#) (const OsclPtrC &d)
- const uint8 * [Ptr](#) ()
- void [SetLength](#) (int32 l)
- int32 [Length](#) ()
- void [Zero](#) ()
- void [Set](#) (OsclPtrC *v)
- void [Set](#) (uint8 *ptr, int32 len, int32 max)
- OsclPtrC [Right](#) (int32 size)
- OsclPtrC [Left](#) (int32 size)

6.177.1 Constructor & Destructor Documentation

6.177.1.1 `OsclPtrC::OsclPtrC (const uint8 *ptr, int32 len, int32 max)` [inline]

6.177.1.2 `OsclPtrC::OsclPtrC (const OsclPtrC & d)` [inline]

6.177.2 Member Function Documentation

6.177.2.1 `OsclPtrC OsclPtrC::Left (int32 size)` [inline]

6.177.2.2 `int32 OsclPtrC::Length ()` [inline]

6.177.2.3 `const uint8* OsclPtrC::Ptr ()` [inline]

6.177.2.4 `OsclPtrC OsclPtrC::Right (int32 size)` [inline]

6.177.2.5 `void OsclPtrC::Set (uint8 *ptr, int32 len, int32 max)` [inline]

6.177.2.6 `void OsclPtrC::Set (OsclPtrC *v)` [inline]

6.177.2.7 `void OsclPtrC::SetLength (int32 l)` [inline]

6.177.2.8 `void OsclPtrC::Zero ()` [inline]

The documentation for this class was generated from the following file:

- [oscl_file_async_read.h](#)

6.178 OsclRand Class Reference

```
#include <oscl_rand.h>
```

Public Methods

- OSCL_COND_IMPORT_REF void [Seed](#) (int32 seed)
- OSCL_COND_IMPORT_REF int32 [Rand](#) ()

6.178.1 Member Function Documentation

6.178.1.1 OSCL_COND_IMPORT_REF int32 OsclRand::Rand ()

6.178.1.2 OSCL_COND_IMPORT_REF void OsclRand::Seed (int32 *seed*)

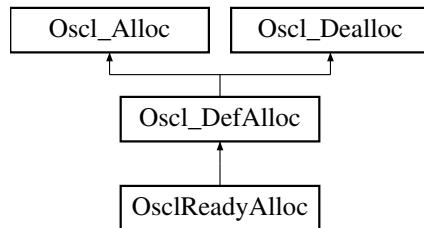
The documentation for this class was generated from the following file:

- [oscl_rand.h](#)

6.179 OsclReadyAlloc Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyAlloc::



Public Methods

- [OsclAny * allocate \(const uint32 size\)](#)
- [OsclAny * allocate_fl \(const uint32 size, const char *file_name, const int line_num\)](#)
- void [deallocate \(OsclAny *p\)](#)

6.179.1 Member Function Documentation

6.179.1.1 [OsclAny* OsclReadyAlloc::allocate \(const uint32 size\) \[virtual\]](#)

Implements [Oscl_DefAlloc](#).

6.179.1.2 [OsclAny* OsclReadyAlloc::allocate_fl \(const uint32 size, const char *file_name, const int line_num\) \[virtual\]](#)

Reimplemented from [Oscl_DefAlloc](#).

6.179.1.3 [void OsclReadyAlloc::deallocate \(OsclAny *p\) \[virtual\]](#)

Implements [Oscl_DefAlloc](#).

The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

6.180 OsclReadyCompare Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Static Public Methods

- int [compare \(TOsclReady &a, TOsclReady &b\)](#)

6.180.1 Member Function Documentation

6.180.1.1 int OsclReadyCompare::compare (TOsclReady &*a*, TOsclReady &*b*) [static]

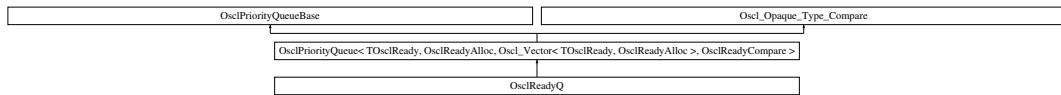
The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

6.181 OsclReadyQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyQ::



Public Methods

- void [Construct](#) (int)
- void [ThreadLogon](#) ()
- void [ThreadLogoff](#) ()
- void [Remove](#) (TOscIReady)
- bool [IsIn](#) (TOscIReady)
- uint32 [Depth](#) ()
- [TOscIReady PopTop](#) ()
- [TOscIReady Top](#) ()
- [TOscIReady WaitAndPopTop](#) ()
- [TOscIReady WaitAndPopTop](#) (uint32)
- int32 [PendComplete](#) (PVActiveBase *pvbase, int32 aReason)
- int32 [WaitForRequestComplete](#) (PVActiveBase *)
- void [RegisterForCallback](#) (OsclSchedulerObserver *aCallback, OsclAny *aCallbackContext)
- void [TimerCallback](#) (uint32 aDelayMicrosec)
- [OsclSchedulerObserver * Callback](#) ()

6.181.1 Member Function Documentation

6.181.1.1 `OsclSchedulerObserver* OsclReadyQ::Callback () [inline]`

6.181.1.2 `void OsclReadyQ::Construct (int)`

6.181.1.3 `uint32 OsclReadyQ::Depth () [inline]`

6.181.1.4 `bool OsclReadyQ::IsIn (TOsclReady)`

6.181.1.5 `int32 OsclReadyQ::PendComplete (PVActiveBase *pvbase, int32 aReason)`

6.181.1.6 `TOsclReady OsclReadyQ::PopTop ()`

6.181.1.7 `void OsclReadyQ::RegisterForCallback (OsclSchedulerObserver *aCallback, OsclAny *aCallbackContext)`

6.181.1.8 `void OsclReadyQ::Remove (TOsclReady)`

6.181.1.9 `void OsclReadyQ::ThreadLogoff ()`

6.181.1.10 `void OsclReadyQ::ThreadLogon ()`

6.181.1.11 `void OsclReadyQ::TimerCallback (uint32 aDelayMicrosec)`

6.181.1.12 `TOsclReady OsclReadyQ::Top ()`

6.181.1.13 `TOsclReady OsclReadyQ::WaitAndPopTop (uint32)`

6.181.1.14 `TOsclReady OsclReadyQ::WaitAndPopTop ()`

6.181.1.15 `int32 OsclReadyQ::WaitForRequestComplete (PVActiveBase *)`

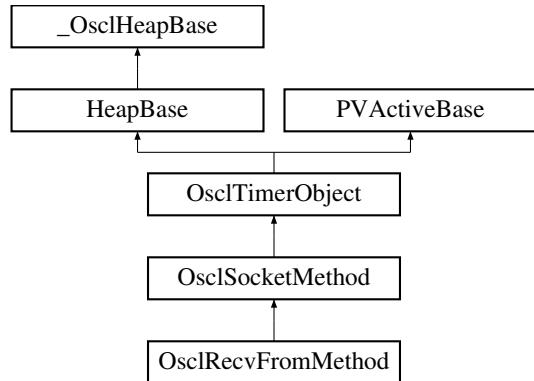
The documentation for this class was generated from the following file:

- `oscl_scheduler_readyq.h`

6.182 OsclRecvFromMethod Class Reference

```
#include <oscl_socket_recv_from.h>
```

Inheritance diagram for OsclRecvFromMethod:::



Public Methods

- [`~OsclRecvFromMethod \(\)`](#)
- [`TPVSocketEvent RecvFrom \(uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeout, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource\)`](#)
- [`uint8 * GetRecvData \(int32 *aLength\)`](#)
- [`OsclRecvFromRequest * RecvFromRequest \(\)`](#)

Static Public Methods

- [`OsclRecvFromMethod * NewL \(OsclIPSocketI &c\)`](#)

6.182.1 Constructor & Destructor Documentation

6.182.1.1 OsclRecvFromMethod::~OsclRecvFromMethod ()

6.182.2 Member Function Documentation

6.182.2.1 `uint8* OsclRecvFromMethod::GetRecvData (int32 * aLength)`

6.182.2.2 `OsclRecvFromMethod* OsclRecvFromMethod::NewL (OsclIPSocketI & c) [static]`

6.182.2.3 `TPVSocketEvent OsclRecvFromMethod::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeout, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource)`

6.182.2.4 `OsclRecvFromRequest* OsclRecvFromMethod::RecvFromRequest () [inline]`

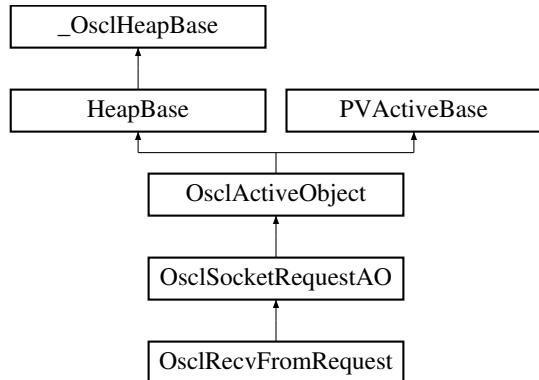
The documentation for this class was generated from the following file:

-
- [oscl_socket_recv_from.h](#)

6.183 OsclRecvFromRequest Class Reference

```
#include <oscl_socket_recv_from.h>
```

Inheritance diagram for OsclRecvFromRequest::



Public Methods

- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvFromRequest (OsclSocketMethod &c)`
- `void RecvFrom (uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource)`
- `void Success ()`

6.183.1 Detailed Description

This is the AO that interacts with the socket server

6.183.2 Constructor & Destructor Documentation

6.183.2.1 OsclRecvFromRequest::OsclRecvFromRequest (`OsclSocketMethod &c`) [inline]

6.183.3 Member Function Documentation

6.183.3.1 `uint8* OsclRecvFromRequest::GetRecvData (int32 * aLength)`

6.183.3.2 `void OsclRecvFromRequest::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource)`

6.183.3.3 `void OsclRecvFromRequest::Success () [virtual]`

Reimplemented from `OsclSocketRequestAO`.

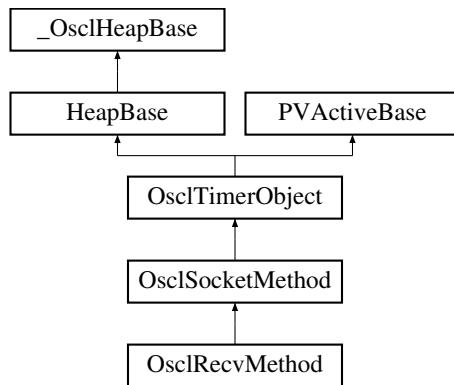
The documentation for this class was generated from the following file:

- [oscl_socket_recv_from.h](#)

6.184 OsclRecvMethod Class Reference

```
#include <oscl_socket_recv.h>
```

Inheritance diagram for OsclRecvMethod::



Public Methods

- [~OsclRecvMethod \(\)](#)
- [TPVSocketEvent Recv \(uint8 *&aPtr, uint32 aMaxLen, int32 aTimeout\)](#)
- [uint8 * GetRecvData \(int32 *aLength\)](#)
- [OsclRecvRequest * RecvRequest \(\)](#)

Static Public Methods

- [OsclRecvMethod * NewL \(OsclIPSocketI &c\)](#)

6.184.1 Constructor & Destructor Documentation

6.184.1.1 OsclRecvMethod::~OsclRecvMethod ()

6.184.2 Member Function Documentation

6.184.2.1 uint8* OsclRecvMethod::GetRecvData (int32 * aLength)

6.184.2.2 OsclRecvMethod* OsclRecvMethod::NewL (OsclIPSocketI & c) [static]

6.184.2.3 TPVSocketEvent OsclRecvMethod::Recv (uint8 *& aPtr, uint32 aMaxLen, int32 aTimeout)

6.184.2.4 OsclRecvRequest* OsclRecvMethod::RecvRequest () [inline]

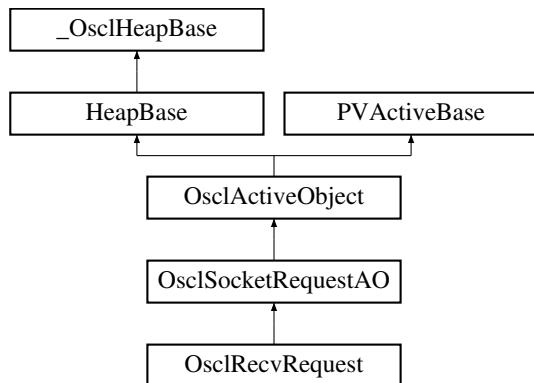
The documentation for this class was generated from the following file:

- [oscl_socket_recv.h](#)

6.185 OsclRecvRequest Class Reference

```
#include <oscl_socket_recv.h>
```

Inheritance diagram for OsclRecvRequest::



Public Methods

- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvRequest (OsclSocketMethod &c)`
- `void Recv (uint8 *&aPtr, uint32 aMaxLen)`
- `void Success ()`

6.185.1 Detailed Description

This is the AO that interacts with the socket server

6.185.2 Constructor & Destructor Documentation

6.185.2.1 OsclRecvRequest::OsclRecvRequest (`OsclSocketMethod &c`) [inline]

6.185.3 Member Function Documentation

6.185.3.1 `uint8* OsclRecvRequest::GetRecvData (int32 * aLength)`

6.185.3.2 `void OsclRecvRequest::Recv (uint8 *& aPtr, uint32 aMaxLen)`

6.185.3.3 `void OsclRecvRequest::Success () [virtual]`

Reimplemented from `OsclSocketRequestAO`.

The documentation for this class was generated from the following file:

- `oscl_socket_recv.h`

6.186 OsclRefCounter Class Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounter:::



Public Methods

- virtual void `addRef ()=0`
- virtual void `removeRef ()=0`
- virtual uint32 `getCount ()=0`
- virtual `~OsclRefCounter ()`

6.186.1 Detailed Description

Interface class for OsclRefCounter implementations

6.186.2 Constructor & Destructor Documentation

6.186.2.1 virtual OsclRefCounter::~OsclRefCounter () [inline, virtual]

6.186.3 Member Function Documentation

6.186.3.1 virtual void OsclRefCounter::addRef () [pure virtual]

Add to the reference count

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_DefAllocWithRefCounter< DefAlloc >`.

6.186.3.2 virtual uint32 OsclRefCounter::getCount () [pure virtual]

Gets the current number of references

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_DefAllocWithRefCounter< DefAlloc >`.

6.186.3.3 virtual void OsclRefCounter::removeRef () [pure virtual]

Delete from reference count

Implemented in `OsclRefCounterDA`, `OsclRefCounterSA< DeallocType >`, `OsclRefCounterMTDA< LockType >`, `OsclRefCounterMTSA< DeallocType, LockType >`, and `Oscl_DefAllocWithRefCounter< DefAlloc >`.

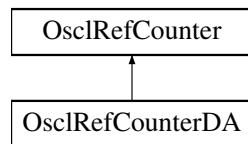
The documentation for this class was generated from the following file:

-
- [oscl_refcounter.h](#)

6.187 OsclRefCounterDA Class Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterDA::



Public Methods

- [OsclRefCounterDA \(OsclAny *p, OsclDestructDealloc *dealloc\)](#)
- virtual [~OsclRefCounterDA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

6.187.1 Detailed Description

Implementation of an [OsclRefCounter](#) that uses a dynamically created deallocator.

6.187.2 Constructor & Destructor Documentation

6.187.2.1 OsclRefCounterDA::OsclRefCounterDA ([OsclAny *p](#), [OsclDestructDealloc *dealloc](#)) [inline]

Constructor Takes a pointer to the buffer to track, and a pointer to the deallocator object which will be used to delete the buffer.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to `delete()`.

Parameters:

p pointer to the buffer to track

dealloc pointer to the deallocator to use when deleting the buffer

6.187.2.2 virtual OsclRefCounterDA::~OsclRefCounterDA () [inline, virtual]

Destructor empty

6.187.3 Member Function Documentation

6.187.3.1 void OsclRefCounterDA::addRef () [inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

6.187.3.2 uint32 OsclRefCounterDA::getCount () [inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

6.187.3.3 void OsclRefCounterDA::removeRef () [inline, virtual]

Remove from the reference count

Implements [OsclRefCounter](#).

The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

6.188 OsclRefCounterMemFrag Class Reference

```
#include <oscl_refcounter_memfrag.h>
```

Public Methods

- [OsclRefCounterMemFrag \(OsclMemoryFragment &m, OsclRefCounter *r, uint32 in_capacity\)](#)
- [OsclRefCounterMemFrag \(const OsclRefCounterMemFrag &x\)](#)
- [OsclRefCounterMemFrag \(\)](#)
- [OsclRefCounterMemFrag & operator= \(const OsclRefCounterMemFrag &x\)](#)
- [~OsclRefCounterMemFrag \(\)](#)
- [OsclRefCounter * getRefCounter \(\)](#)
- [OsclMemoryFragment & getMemFrag \(\)](#)
- [OsclAny * getMemFragPtr \(\)](#)
- [uint32 getMemFragSize \(\)](#)
- [uint32 getCapacity \(\)](#)
- [uint32 getCount \(\)](#)

6.188.1 Detailed Description

Class to contain a memory fragment with it's associated reference counter.

6.188.2 Constructor & Destructor Documentation

6.188.2.1 OsclRefCounterMemFrag::OsclRefCounterMemFrag ([OsclMemoryFragment & m](#), [OsclRefCounter * r](#), [uint32 in_capacity](#)) [inline]

Constructor. A valid memory fragment and reference counter are required as input. The memory fragment structure will be copied locally.

Parameters:

- m* reference to memory fragment
- r* pointer to the reference counter associated with the memory fragment.

6.188.2.2 OsclRefCounterMemFrag::OsclRefCounterMemFrag ([const OsclRefCounterMemFrag & x](#)) [inline]

Copy constructor.

6.188.2.3 OsclRefCounterMemFrag::OsclRefCounterMemFrag () [inline]

Default constructor.

6.188.2.4 OsclRefCounterMemFrag::~OsclRefCounterMemFrag () [inline]

Destructor. Removes this object's reference from the reference counter. The reference counter will not be deleted. The reference counter is designed to self-delete when it's reference count reaches 0.

6.188.3 Member Function Documentation

6.188.3.1 **uint32 OsclRefCounterMemFrag::getCapacity () [inline]**

Returns the capacity of the memory fragment

Returns:

6.188.3.2 **uint32 OsclRefCounterMemFrag::getCount () [inline]**

Returns the reference counter's current count.

6.188.3.3 **OsclMemoryFragment& OsclRefCounterMemFrag::getMemFrag () [inline]**

Returns a reference to the contained memory fragment structure.

6.188.3.4 **OsclAny* OsclRefCounterMemFrag::getMemFragPtr () [inline]**

Returns a pointer to the memory fragment data.

6.188.3.5 **uint32 OsclRefCounterMemFrag::getMemFragSize () [inline]**

Returns the size of the memory fragment data which equals its filled size.

Returns:

6.188.3.6 **OsclRefCounter* OsclRefCounterMemFrag::getRefCounter () [inline]**

Returns a pointer to the contained reference counter object

6.188.3.7 **OsclRefCounterMemFrag& OsclRefCounterMemFrag::operator= (const OsclRefCounterMemFrag & x) [inline]**

Assignment Operator

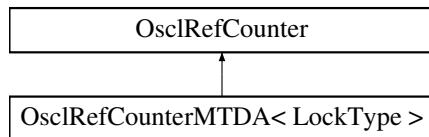
The documentation for this class was generated from the following file:

- [oscl_refcounter_memfrag.h](#)

6.189 OsclRefCounterMTDA< LockType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterMTDA< LockType >::



Public Methods

- [OsclRefCounterMTDA \(OsclAny *p, OsclDestructDealloc *dealloc\)](#)
- virtual [~OsclRefCounterMTDA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

6.189.1 Detailed Description

template<class LockType> class OsclRefCounterMTDA< LockType >

Implementation of [OsclRefCounterDA](#) for multi-threaded use. A templated lock class must be specified.

6.189.2 Constructor & Destructor Documentation

6.189.2.1 template<class LockType> OsclRefCounterMTDA< LockType >::OsclRefCounterMTDA (OsclAny *p, OsclDestructDealloc *dealloc) [inline]

Constructor Takes a pointer to the buffer to track, and a pointer to the deallocator object which will be used to delete the buffer.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to [delete\(\)](#).

Parameters:

p pointer to the buffer to track

dealloc pointer to the deallocator to use when deleting the buffer

6.189.2.2 template<class LockType> virtual OsclRefCounterMTDA< LockType >::~OsclRefCounterMTDA () [inline, virtual]

Destructor empty

6.189.3 Member Function Documentation

**6.189.3.1 template<class LockType> void OsclRefCounterMTDA< LockType >::addRef ()
[inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**6.189.3.2 template<class LockType> uint32 OsclRefCounterMTDA< LockType >::getCount ()
[inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**6.189.3.3 template<class LockType> void OsclRefCounterMTDA< LockType >::removeRef ()
[inline, virtual]**

Remove from the reference count

Implements [OsclRefCounter](#).

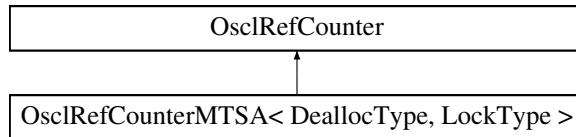
The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

6.190 OsclRefCounterMTSA< DeallocType, LockType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterMTSA< DeallocType, LockType >::



Public Methods

- [OsclRefCounterMTSA \(OsclAny *p\)](#)
- virtual [~OsclRefCounterMTSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

6.190.1 Detailed Description

```
template<class DeallocType, class LockType> class OsclRefCounterMTSA< DeallocType, LockType >
```

Implementation of [OsclRefCounterSA](#) for multi-threaded use. A templated lock class must be specified.

6.190.2 Constructor & Destructor Documentation

6.190.2.1 template<class DeallocType, class LockType> OsclRefCounterMTSA< DeallocType, LockType >::OsclRefCounterMTSA (OsclAny * p) [inline]

Constructor Takes a pointer to the buffer to track.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to `delete()`.

Parameters:

p pointer to the buffer to track

6.190.2.2 template<class DeallocType, class LockType> virtual OsclRefCounterMTSA< DeallocType, LockType >::~OsclRefCounterMTSA () [inline, virtual]

Destructor empty

6.190.3 Member Function Documentation

6.190.3.1 template<class DeallocType, class LockType> void OsclRefCounterMTSA< DeallocType, LockType >::addRef () [inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

6.190.3.2 template<class DeallocType, class LockType> uint32 OsclRefCounterMTSA< DeallocType, LockType >::getCount () [inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

6.190.3.3 template<class DeallocType, class LockType> void OsclRefCounterMTSA< DeallocType, LockType >::removeRef () [inline, virtual]

Remove from the reference count

Implements [OsclRefCounter](#).

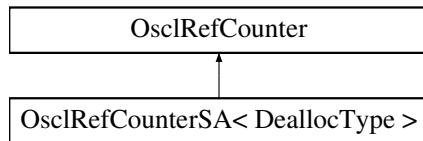
The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

6.191 OsclRefCounterSA< DeallocType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterSA< DeallocType >::



Public Methods

- [OsclRefCounterSA \(OsclAny *p\)](#)
- virtual [~OsclRefCounterSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

6.191.1 Detailed Description

template<class DeallocType> class OsclRefCounterSA< DeallocType >

Implementation of an [OsclRefCounter](#) that uses a statically created deallocator.

6.191.2 Constructor & Destructor Documentation

6.191.2.1 template<class DeallocType> OsclRefCounterSA< DeallocType >::OsclRefCounterSA (OsclAny *p) [inline]

Constructor Takes a pointer to the buffer to track.

When the reference count reaches zero, the buffer will be deleted by the deallocator. Also, the [OsclRefCounter](#) object (this) will self-destruct when the reference count is zero. In some cases the [OsclRefCounter](#) object will be part of the buffer being deleted. For such cases, the object pointer must be equal to the buffer pointer given at construction. If the object is not part of the buffer being deleted, it will self-destruct with a call to [delete\(\)](#).

Parameters:

p pointer to the buffer to track

6.191.2.2 template<class DeallocType> virtual OsclRefCounterSA< DeallocType >::~OsclRefCounterSA () [inline, virtual]

Destructor empty

6.191.3 Member Function Documentation

6.191.3.1 template<class DeallocType> void OsclRefCounterSA< DeallocType >::addRef ()
[inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

6.191.3.2 template<class DeallocType> uint32 OsclRefCounterSA< DeallocType >::getCount ()
[inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

6.191.3.3 template<class DeallocType> void OsclRefCounterSA< DeallocType >::removeRef ()
[inline, virtual]

Remove from the reference count

Implements [OsclRefCounter](#).

The documentation for this class was generated from the following file:

- [oscl_refcounter.h](#)

6.192 OsclRegistryAccessClient Class Reference

```
#include <oscl_registry_access_client.h>
```

Public Methods

- OSCL_IMPORT_REF OsclRegistryAccessClient ()
- OSCL_IMPORT_REF ~OsclRegistryAccessClient ()
- OSCL_IMPORT_REF int32 Connect ()
- OSCL_IMPORT_REF OsclComponentFactory GetFactory (OSCL_String &aComponent)
- OSCL_IMPORT_REF void GetFactories (OSCL_String &aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &aVec)
- OSCL_IMPORT_REF void Close ()

6.192.1 Constructor & Destructor Documentation

6.192.1.1 OSCL_IMPORT_REF OsclRegistryAccessClient::OsclRegistryAccessClient ()

6.192.1.2 OSCL_IMPORT_REF OsclRegistryAccessClient::~OsclRegistryAccessClient ()

6.192.2 Member Function Documentation

6.192.2.1 OSCL_IMPORT_REF void OsclRegistryAccessClient::Close ()

Close and cleanup session.

6.192.2.2 OSCL_IMPORT_REF int32 OsclRegistryAccessClient::Connect ()

Create a session.

Returns:

OsclErrNone on success.

6.192.2.3 OSCL_IMPORT_REF void OsclRegistryAccessClient::GetFactories (OSCL_String & aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > & aVec)

Get all factories for a given registry type.

Parameters:

aRegistry: registry MIME type

aVec: output component factory + mimestring vector.

6.192.2.4 OSCL_IMPORT_REF OsclComponentFactory OsclRegistryAccessClient::GetFactory (OSCL_String & aComponent)

Lookup a factory by registry and component mime type.

Parameters:

aComponent: registry+component MIME type

Returns:

Factory. Factory will be NULL if not found.

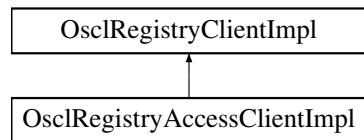
The documentation for this class was generated from the following file:

- [oscl_registry_access_client.h](#)

6.193 OsclRegistryAccessClientImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryAccessClientImpl::



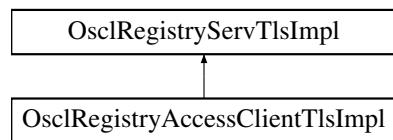
The documentation for this class was generated from the following file:

- [oscl_registry_client_impl.h](#)

6.194 OsclRegistryAccessClientTlsImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryAccessClientTlsImpl::



The documentation for this class was generated from the following file:

- [oscl_registry_client_impl.h](#)

6.195 OsclRegistryAccessElement Class Reference

```
#include <oscl_registry_types.h>
```

Data Fields

- [OsclComponentFactory](#) iFactory
- [OSCL_HeapString< OsclMemAllocator >](#) iMimeType

6.195.1 Detailed Description

A class used to access the registry data

6.195.2 Field Documentation

6.195.2.1 [OsclComponentFactory](#) OsclRegistryAccessElement::iFactory

6.195.2.2 [OSCL_HeapString<OsclMemAllocator>](#) OsclRegistryAccessElement::iMimeType

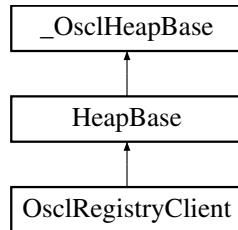
The documentation for this class was generated from the following file:

- [oscl_registry_types.h](#)

6.196 OsclRegistryClient Class Reference

```
#include <oscl_registry_client.h>
```

Inheritance diagram for OsclRegistryClient::



Public Methods

- OSCL_IMPORT_REF [OsclRegistryClient \(\)](#)
- OSCL_IMPORT_REF [~OsclRegistryClient \(\)](#)
- OSCL_IMPORT_REF int32 [Connect \(bool aPerThread=false\)](#)
- OSCL_IMPORT_REF int32 [Register \(OSCL_String &aComponentID, OsclComponentFactory aFactory\)](#)
- OSCL_IMPORT_REF int32 [UnRegister \(OSCL_String &aComponentID\)](#)
- OSCL_IMPORT_REF void [Close \(\)](#)

6.196.1 Constructor & Destructor Documentation

6.196.1.1 OSCL_IMPORT_REF OsclRegistryClient::OsclRegistryClient ()

6.196.1.2 OSCL_IMPORT_REF OsclRegistryClient::~OsclRegistryClient ()

6.196.2 Member Function Documentation

6.196.2.1 OSCL_IMPORT_REF void OsclRegistryClient::Close ()

Close and cleanup. All components registered in this session are automatically unregistered.

6.196.2.2 OSCL_IMPORT_REF int32 OsclRegistryClient::Connect (bool *aPerThread* = false)

Create a session.

Parameters:

aPerThread: Select per-thread registry instead of global registry.

Returns:

OsclErrNone on success.

**6.196.2.3 OSCL_IMPORT_REF int32 OsclRegistryClient::Register ([OSCL_String &](#)
aComponentID, [OsclComponentFactory](#) *aFactory*)**

Register a component factory by registry ID and component ID.

Parameters:

aComponentID: registry + component mime-string.

aFactory: factory function pointer.

aParam: component Create param.

Returns:

OsclErrNone on success.

**6.196.2.4 OSCL_IMPORT_REF int32 OsclRegistryClient::UnRegister ([OSCL_String &](#)
aComponentID)**

Unregister a previously registered component.

Returns:

OsclErrNone on success.

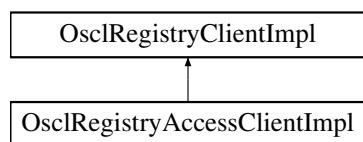
The documentation for this class was generated from the following file:

- [oscl_registry_client.h](#)

6.197 OsclRegistryClientImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryClientImpl:



Protected Methods

- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL_String &, OsclComponentFactory\)](#)
- int32 [UnRegister \(OSCL_String &\)](#)
- [OsclComponentFactory GetFactory \(OSCL_String &\)](#)
- void [GetFactories \(OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &\)](#)

Friends

- class [OsclRegistryClient](#)
- class [OsclRegistryAccessClient](#)

6.197.1 Member Function Documentation

6.197.1.1 **void OsclRegistryClientImpl::Close (void)** [inline, protected]

6.197.1.2 **int32 OsclRegistryClientImpl::Connect ()** [inline, protected]

6.197.1.3 **void OsclRegistryClientImpl::GetFactories (OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &)** [inline, protected]

6.197.1.4 **OsclComponentFactory OsclRegistryClientImpl::GetFactory (OSCL_String &)**
[inline, protected]

6.197.1.5 **int32 OsclRegistryClientImpl::Register (OSCL_String &, OsclComponentFactory)**
[inline, protected]

6.197.1.6 **int32 OsclRegistryClientImpl::UnRegister (OSCL_String &)** [inline,
protected]

6.197.2 Friends And Related Function Documentation

6.197.2.1 **friend class OsclRegistryAccessClient** [friend]

6.197.2.2 **friend class OsclRegistryClient** [friend]

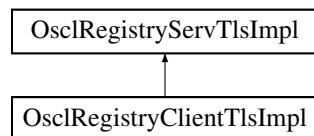
The documentation for this class was generated from the following file:

- [oscl_registry_client_impl.h](#)

6.198 OsclRegistryClientTlsImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryClientTlsImpl::



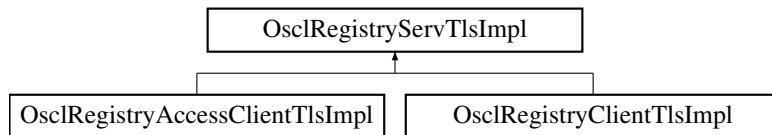
The documentation for this class was generated from the following file:

- [oscl_registry_client_impl.h](#)

6.199 OsclRegistryServTlsImpl Class Reference

```
#include <oscl_registry_serv_impl_tls.h>
```

Inheritance diagram for OsclRegistryServTlsImpl::



Protected Methods

- [OsclRegistryServTlsImpl \(\)](#)
- virtual [~OsclRegistryServTlsImpl \(\)](#)
- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL_String &aComponentID, OsclComponentFactory aFactory\)](#)
- int32 [UnRegister \(OSCL_String &aComponentID\)](#)
- [OsclComponentFactory GetFactory \(OSCL_String &aComponent\)](#)
- void [GetFactories \(OSCL_String &aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &aVec\)](#)

Friends

- class [OsclRegistryClient](#)
- class [OsclRegistryAccessClient](#)

6.199.1 Constructor & Destructor Documentation

6.199.1.1 `OsclRegistryServTlsImpl::OsclRegistryServTlsImpl ()` [protected]

6.199.1.2 `virtual OsclRegistryServTlsImpl::~OsclRegistryServTlsImpl ()` [protected, virtual]

6.199.2 Member Function Documentation

6.199.2.1 `void OsclRegistryServTlsImpl::Close ()` [protected]

6.199.2.2 `int32 OsclRegistryServTlsImpl::Connect ()` [protected]

6.199.2.3 `void OsclRegistryServTlsImpl::GetFactories (OSCL_String & aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > & aVec)` [protected]

6.199.2.4 `OsclComponentFactory OsclRegistryServTlsImpl::GetFactory (OSCL_String & aComponent)` [protected]

6.199.2.5 `int32 OsclRegistryServTlsImpl::Register (OSCL_String & aComponentID, OsclComponentFactory aFactory)` [protected]

6.199.2.6 `int32 OsclRegistryServTlsImpl::UnRegister (OSCL_String & aComponentID)` [protected]

6.199.3 Friends And Related Function Documentation

6.199.3.1 `friend class OsclRegistryAccessClient` [friend]

6.199.3.2 `friend class OsclRegistryClient` [friend]

The documentation for this class was generated from the following file:

- [oscl_registry_serv_impl_tls.h](#)

6.200 OsclScheduler Class Reference

```
#include <oscl_scheduler.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [Init](#) (const char *name, [Oscl_DefAlloc](#) *alloc=NULL, int nreserve=20)
- OSCL_IMPORT_REF void [Cleanup](#) ()

6.200.1 Detailed Description

Per-thread scheduler initialization and cleanup.

6.200.2 Member Function Documentation

6.200.2.1 OSCL_IMPORT_REF void OsclScheduler::Cleanup () [static]

This routine uninstalls and destroys Oscl scheduler for the calling thread.

6.200.2.2 OSCL_IMPORT_REF void OsclScheduler::Init (const char * *name*, [Oscl_DefAlloc](#) * *alloc* = NULL, int *nreserve* = 20) [static]

This routine creates and installs a scheduler in the calling thread.

Parameters:

- name*:** (input param) scheduler name.
- alloc*:** (input param) optional allocator to use for the internal implementation.
- nreserve*:** (input param) optional value for ready queue reserve size.

The documentation for this class was generated from the following file:

- [oscl_scheduler.h](#)

6.201 OsclSchedulerObserver Class Reference

```
#include <oscl_scheduler.h>
```

Public Methods

- virtual void [OsclSchedulerTimerCallback](#) ([OsclAny](#) *aContext, uint32 aDelayMsec)=0
- virtual void [OsclSchedulerReadyCallback](#) ([OsclAny](#) *aContext)=0
- virtual [~OsclSchedulerObserver](#) ()

6.201.1 Detailed Description

OsclSchedulerObserver is an observer class for use when running scheduler in non-blocking mode. The scheduler observer can register for callbacks so it will be notified when it is necessary to run scheduler again. Note: non-blocking mode and scheduler callbacks are not supported on Symbian.

6.201.2 Constructor & Destructor Documentation

6.201.2.1 virtual [OsclSchedulerObserver](#)::[~OsclSchedulerObserver](#) () [inline, virtual]

6.201.3 Member Function Documentation

6.201.3.1 virtual void [OsclSchedulerObserver::OsclSchedulerReadyCallback](#) ([OsclAny](#) * *aContext*) [pure virtual]

[OsclSchedulerReadyCallback](#) is called when the ready queue is updated, meaning an AO is ready to run. Scheduler needs to be run ASAP. Calling context may be any thread, so be careful!

The current observer is cleared before making the callback, so the observer must call [RegisterForCallback](#) again if it wants further notifications.

6.201.3.2 virtual void [OsclSchedulerObserver::OsclSchedulerTimerCallback](#) ([OsclAny](#) * *aContext*, uint32 *aDelayMsec*) [pure virtual]

[OsclSchedulerTimerCallback](#) is called when the front of the timer queue is updated. This means the minimum delay has changed and scheduler needs to be run again after *aDelayMsec*. Calling context is in-thread.

The current observer is cleared before making the callback, so the observer must call [RegisterForCallback](#) again if it wants further notifications.

The documentation for this class was generated from the following file:

- [oscl_scheduler.h](#)

6.202 OsclScopedLock< LockClass > Class Template Reference

The OsclScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope.

```
#include <oscl_lock_base.h>
```

Public Methods

- [OsclScopedLock \(LockClass &inLock\)](#)
Default constructor Initializes the pointer and takes ownership.
- [~OsclScopedLock \(\)](#)
Destructor.

6.202.1 Detailed Description

template<class LockClass> class OsclScopedLock< LockClass >

The OsclScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope.

The purpose of this class is to provide a way to prevent accidental resource leaks in a class or a method, due to "not remembering to unlock" variables which might lead to deadlock conditions.

6.202.2 Constructor & Destructor Documentation

**6.202.2.1 template<class LockClass> OsclScopedLock< LockClass >::OsclScopedLock
(LockClass & inLock) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**6.202.2.2 template<class LockClass> OsclScopedLock< LockClass >::~OsclScopedLock ()
[inline]**

Destructor.

The pointer is deleted in case this class still has ownership

The documentation for this class was generated from the following file:

- [oscl_lock_base.h](#)

6.203 OsclSelect Class Reference

```
#include <oscl_init.h>
```

Public Methods

- [OsclSelect \(\)](#)
- [OsclSelect \(Oscl_DefAlloc *erralloc, Oscl_DefAlloc *schedalloc, const char *name, int32 reserve=10, bool heapcheck=false, FILE *output=NULL\)](#)

Data Fields

- bool [iOsclBase](#)
- bool [iOsclMemory](#)
- bool [iOsclErrorTrap](#)
- bool [iOsclLogger](#)
- bool [iOsclScheduler](#)
- [Oscl_DefAlloc * iErrAlloc](#)
- [Oscl_DefAlloc * iSchedulerAlloc](#)
- const char * [iSchedulerName](#)
- int32 [iSchedulerReserve](#)
- bool [iHeapCheck](#)
- FILE * [iOutputFile](#)

6.203.1 Detailed Description

Oscl Module selection and Init/Cleanup options.

6.203.2 Constructor & Destructor Documentation

6.203.2.1 OsclSelect::OsclSelect () [inline]

6.203.2.2 OsclSelect::OsclSelect ([Oscl_DefAlloc](#) * *erralloc*, [Oscl_DefAlloc](#) * *schedalloc*, const char * *name*, int32 *reserve* = 10, bool *heapcheck* = false, FILE * *output* = NULL) [inline]

6.203.3 Field Documentation

6.203.3.1 [Oscl_DefAlloc](#)* OsclSelect::iErrAlloc

6.203.3.2 bool OsclSelect::iHeapCheck

6.203.3.3 bool OsclSelect::iOsclBase

6.203.3.4 bool OsclSelect::iOsclErrorTrap

6.203.3.5 bool OsclSelect::iOsclLogger

6.203.3.6 bool OsclSelect::iOsclMemory

6.203.3.7 bool OsclSelect::iOsclScheduler

6.203.3.8 FILE* OsclSelect::iOutputFile

6.203.3.9 [Oscl_DefAlloc](#)* OsclSelect::iSchedulerAlloc

6.203.3.10 const char* OsclSelect::iSchedulerName

6.203.3.11 int32 OsclSelect::iSchedulerReserve

The documentation for this class was generated from the following file:

- [oscl_init.h](#)

6.204 OsclSemaphore Class Reference

```
#include <oscl_semaphore.h>
```

Public Methods

- OSCL_IMPORT_REF OsclSemaphore ()
- OSCL_IMPORT_REF ~OsclSemaphore ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Create (uint32 initVal=0)
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Close ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Wait ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Wait (uint32 timeout_msec)
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError TryWait ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Signal ()

6.204.1 Detailed Description

Class Semaphore

6.204.2 Constructor & Destructor Documentation

6.204.2.1 OSCL_IMPORT_REF OsclSemaphore::OsclSemaphore ()

Class constructor

6.204.2.2 OSCL_IMPORT_REF OsclSemaphore::~OsclSemaphore ()

Class destructor

6.204.3 Member Function Documentation

6.204.3.1 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::Close ()

Closes the Semaphore

Parameters:

It wont take any parameters

Returns:

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

6.204.3.2 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::Create (uint32 initVal = 0)

Creates the Semaphore

Parameters:

Intialcount

Returns:

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

6.204.3.3 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::Signal ()

Signals that the thread is finished with the Semaphore

Parameters:

It wont take any parameters

Returns:

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

6.204.3.4 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::TryWait ()

Try to acquire semaphore ,if the semaphore is already acquired by another thread, calling thread immediately returns with out blocking

Parameters:

It wont take any parameters

Returns:

Returns SUCCESS_ERROR if the semaphore was acquired, SEM_LOCKED_ERROR if the semaphore cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT_IMPLEMENTED.

6.204.3.5 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::Wait (uint32 timeout_msec)

Makes the thread to wait on the Semaphore, with a timeout.

Parameters:

timeout in milliseconds.

Returns:

Returns SUCCESS_ERROR if the semaphore was aquired, WAIT_TIMEOUT_ERROR if the timeout expired without acquiring the semaphore, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT_IMPLEMENTED.

6.204.3.6 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclSemaphore::Wait ()

Makes the thread to wait on the Semaphore

Parameters:

It wont take any parameters

Returns:

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

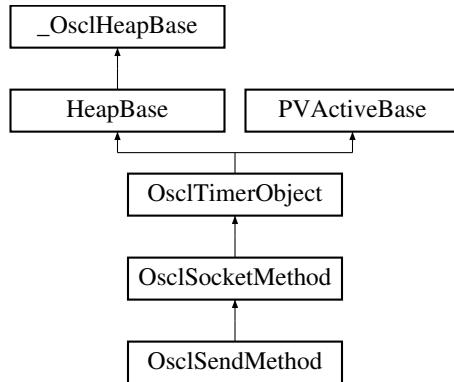
The documentation for this class was generated from the following file:

- [oscl_semaphore.h](#)

6.205 OsclSendMethod Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OsclSendMethod::



Public Methods

- [~OsclSendMethod \(\)](#)
- [TPVSocketEvent Send \(const uint8 *aPtr, uint32 aLen, int32 aTimeout\)](#)
- [uint8 * GetSendData \(int32 *aLength\)](#)
- [OsclSendRequest * SendRequest \(\)](#)

Static Public Methods

- [OsclSendMethod * NewL \(OsclIPSocketI &c\)](#)

6.205.1 Constructor & Destructor Documentation

6.205.1.1 OsclSendMethod::~OsclSendMethod ()

6.205.2 Member Function Documentation

6.205.2.1 uint8* OsclSendMethod::GetSendData (int32 * aLength)

6.205.2.2 OsclSendMethod* OsclSendMethod::NewL (OsclIPSocketI & c) [static]

6.205.2.3 TPVSocketEvent OsclSendMethod::Send (const uint8 *& aPtr, uint32 aLen, int32 aTimeout)

6.205.2.4 OsclSendRequest* OsclSendMethod::SendRequest () [inline]

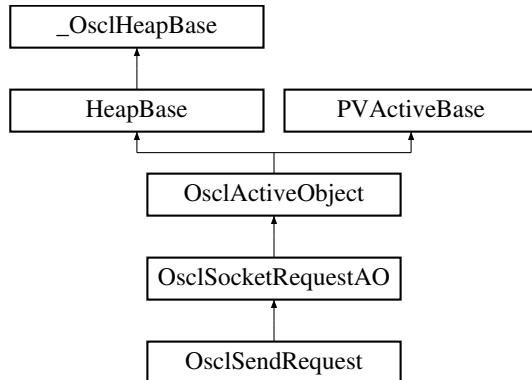
The documentation for this class was generated from the following file:

- [oscl_socket_send.h](#)

6.206 OsclSendRequest Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OsclSendRequest::



Public Methods

- [OsclSendRequest \(OsclSocketMethod &c\)](#)
- void [Send \(const uint8 *&aPtr, uint32 aLen\)](#)
- void [Success \(\)](#)
- uint8 * [GetSendData \(int32 *aLength\)](#)

6.206.1 Constructor & Destructor Documentation

6.206.1.1 OsclSendRequest::OsclSendRequest (OsclSocketMethod & c) [inline]

6.206.2 Member Function Documentation

6.206.2.1 uint8* OsclSendRequest::GetSendData (int32 * aLength)

6.206.2.2 void OsclSendRequest::Send (const uint8 *& aPtr, uint32 aLen)

6.206.2.3 void OsclSendRequest::Success () [virtual]

Reimplemented from [OsclSocketRequestAO](#).

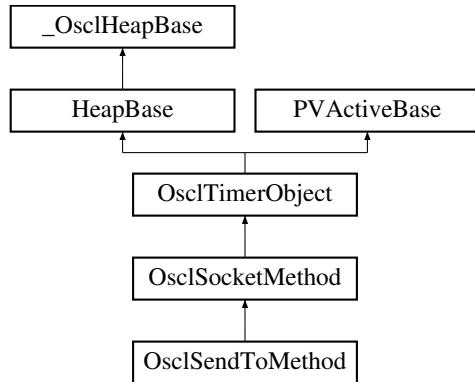
The documentation for this class was generated from the following file:

- [oscl_socket_send.h](#)

6.207 OsclSendToMethod Class Reference

```
#include <oscl_socket_send_to.h>
```

Inheritance diagram for OsclSendToMethod::



Public Methods

- [~OsclSendToMethod \(\)](#)
- [TPVSocketEvent SendTo \(const uint8 *&aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [uint8 * GetSendData \(int32 *aLength\)](#)
- [OsclSendToRequest * SendToRequest \(\)](#)

Static Public Methods

- [OsclSendToMethod * NewL \(OsclIPSocketI &c\)](#)

6.207.1 Constructor & Destructor Documentation

6.207.1.1 OsclSendToMethod::~OsclSendToMethod ()

6.207.2 Member Function Documentation

6.207.2.1 uint8* OsclSendToMethod::GetSendData (int32 * aLength)

6.207.2.2 OsclSendToMethod* OsclSendToMethod::NewL (OsclIPSocketI & c) [static]

6.207.2.3 TPVSocketEvent OsclSendToMethod::SendTo (const uint8 *& aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeout)

6.207.2.4 OsclSendToRequest* OsclSendToMethod::SendToRequest () [inline]

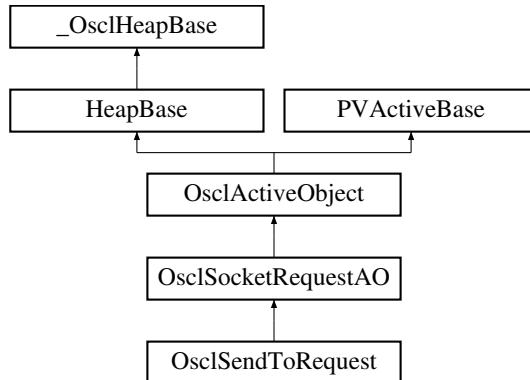
The documentation for this class was generated from the following file:

- [oscl_socket_send_to.h](#)

6.208 OsclSendToRequest Class Reference

```
#include <oscl_socket_send_to.h>
```

Inheritance diagram for OsclSendToRequest::



Public Methods

- [OsclSendToRequest \(OsclSocketMethod &c\)](#)
- void [SendTo \(const uint8 *&aPtr, uint32 aLen, OsclNetworkAddress &aAddress\)](#)
- void [Success \(\)](#)
- uint8 * [GetSendData \(int32 *aLength\)](#)

6.208.1 Detailed Description

This is the AO that interacts with the socket server

6.208.2 Constructor & Destructor Documentation

6.208.2.1 OsclSendToRequest::OsclSendToRequest ([OsclSocketMethod & c](#)) [inline]

6.208.3 Member Function Documentation

6.208.3.1 uint8* OsclSendToRequest::GetSendData (int32 * aLength)

6.208.3.2 void OsclSendToRequest::SendTo (const uint8 *& aPtr, uint32 aLen, [OsclNetworkAddress & aAddress](#))

6.208.3.3 void OsclSendToRequest::Success () [virtual]

Reimplemented from [OsclSocketRequestAO](#).

The documentation for this class was generated from the following file:

- [oscl_socket_send_to.h](#)

6.209 OsclSharedPtr< TheClass > Class Template Reference

A parameterized smart pointer class.

```
#include <oscl_shared_ptr.h>
```

Public Methods

- **OsclSharedPtr ()**
Constructor.
- **OsclSharedPtr (TheClass *inClassPtr, OsclRefCounter *in_refcnt)**
Constructor.
- **OsclSharedPtr (const OsclSharedPtr &inSharedPtr)**
Copy constructor.
- **virtual ~OsclSharedPtr ()**
Destructor.
- **TheClass * operator → ()**
TheClass & operator * ()
The indirection operator returns a reference to an object of the parameterized type.
- **operator TheClass * ()**
Casting operator.
- **TheClass * GetRep ()**
Use this function to get a pointer to the wrapped object.
- **OsclRefCounter * GetRefCounter ()**
Get the refcount pointer. This should primarily be used for conversion operations.
- **int get_count ()**
Get a count of how many references to the object exist.
- **void Bind (const OsclSharedPtr &inHandle)**
Use this function to bind an existing OsclSharedPtr to a already-wrapped object.
- **void Bind (TheClass *ptr, OsclRefCounter *in_refcnt)**
Use this function to bind an existing OsclSharedPtr to a new (unwrapped) object.
- **void Unbind ()**
Use this function of unbind an existing OsclSharedPtr.
- **OsclSharedPtr & operator= (const OsclSharedPtr &inSharedPtr)**
Assignment operator.
- **bool operator== (const OsclSharedPtr &b) const**
Test for equality to see if two PVHandles wrap the same object.

6.209.1 Detailed Description

template<class TheClass> class OsclSharedPtr< TheClass >

A parameterized smart pointer class.

6.209.2 Constructor & Destructor Documentation

6.209.2.1 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr () [inline]

Constructor.

6.209.2.2 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr (TheClass * *inClassPtr*, OsclRefCounter * *in_refcnt*) [inline]

Constructor.

Parameters:

inClassPtr A pointer to an instance of the parameterized type that the new OsclSharedPtr will wrap.

6.209.2.3 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr (const OsclSharedPtr< TheClass > & *inSharedPtr*) [inline]

Copy constructor.

6.209.2.4 template<class TheClass> virtual OsclSharedPtr< TheClass >::~OsclSharedPtr () [inline, virtual]

Destructor.

6.209.3 Member Function Documentation

6.209.3.1 template<class TheClass> int OsclSharedPtr< TheClass >::get_count () [inline]

Get a count of how many references to the object exist.

6.209.3.2 template<class TheClass> OsclRefCounter* OsclSharedPtr< TheClass >::GetRefCounter () [inline]

Get the refcount pointer. This should primarily be used for conversion operations.

6.209.3.3 template<class TheClass> TheClass* OsclSharedPtr< TheClass >::GetRep () [inline]

Use this function to get a pointer to the wrapped object.

**6.209.3.4 template<class TheClass> TheClass& OsclSharedPtr< TheClass >::operator * ()
[inline]**

The indirection operator returns a reference to an object of the parameterized type.

**6.209.3.5 template<class TheClass> OsclSharedPtr< TheClass >::operator TheClass * ()
[inline]**

Casting operator.

**6.209.3.6 template<class TheClass> TheClass* OsclSharedPtr< TheClass >::operator -> ()
[inline]**

The dereferencing operator returns a pointer to the parameterized type and can be used to access member elements of TheClass.

6.209.3.7 template<class TheClass> OsclSharedPtr& OsclSharedPtr< TheClass >::operator= (const OsclSharedPtr< TheClass > & *inSharedPtr*) [inline]

Assignment operator.

6.209.3.8 template<class TheClass> void OsclSharedPtr< TheClass >::Unbind () [inline]

Use this function of unbind an existing OsclSharedPtr.

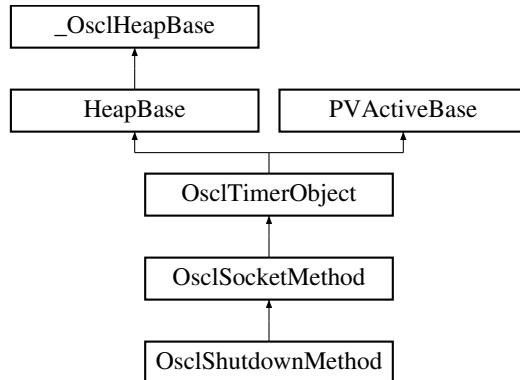
The documentation for this class was generated from the following file:

- [oscl_shared_ptr.h](#)

6.210 OsclShutdownMethod Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OsclShutdownMethod::



Public Methods

- [~OsclShutdownMethod \(\)](#)
- [TPVSocketEvent Shutdown \(TPVSocketShutdown aHow, int32 aTimeout\)](#)
- [OsclShutdownRequest * ShutdownRequest \(\)](#)

Static Public Methods

- [OsclShutdownMethod * NewL \(OsclIPSocketI &c\)](#)

6.210.1 Constructor & Destructor Documentation

6.210.1.1 OsclShutdownMethod::~OsclShutdownMethod ()

6.210.2 Member Function Documentation

6.210.2.1 OsclShutdownMethod* OsclShutdownMethod::NewL (OsclIPSocketI &c) [static]

6.210.2.2 TPVSocketEvent OsclShutdownMethod::Shutdown (TPVSocketShutdown aHow, int32 aTimeout)

6.210.2.3 OsclShutdownRequest* OsclShutdownMethod::ShutdownRequest () [inline]

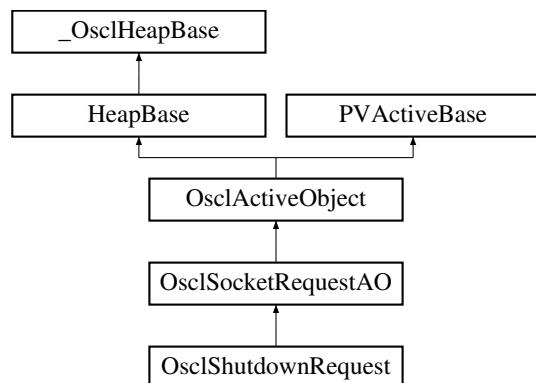
The documentation for this class was generated from the following file:

- [oscl_socket_shutdown.h](#)

6.211 OsclShutdownRequest Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OsclShutdownRequest::



Public Methods

- [OsclShutdownRequest \(OsclSocketMethod &c\)](#)
- [void Shutdown \(TPVSocketShutdown aHow\)](#)

6.211.1 Detailed Description

This is the AO that interacts with the socket server

6.211.2 Constructor & Destructor Documentation

6.211.2.1 [OsclShutdownRequest::OsclShutdownRequest \(OsclSocketMethod & c\) \[inline\]](#)

6.211.3 Member Function Documentation

6.211.3.1 [void OsclShutdownRequest::Shutdown \(TPVSocketShutdown aHow\)](#)

The documentation for this class was generated from the following file:

- [oscl_socket_shutdown.h](#)

6.212 OsclSingleton< T, ID, Registry > Class Template Reference

```
#include <oscl_singleton.h>
```

Public Methods

- [OsclSingleton \(\)](#)
- [~OsclSingleton \(\)](#)
- [T & operator * \(\) const](#)

The indirection operator () accesses a value indirectly, through a pointer.*

- [T * operator → \(\) const](#)

The indirection operator (->) accesses a value indirectly, through a pointer.

- [bool set \(\)](#)

set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

Protected Attributes

- [T * _Ptr](#)

```
template<class T, uint32 ID, class Registry = OsclSingletonRegistry> class OsclSingleton< T, ID, Registry >
```

6.212.1 Constructor & Destructor Documentation

6.212.1.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> OsclSingleton< T, ID, Registry >::OsclSingleton () [inline]

6.212.1.2 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> OsclSingleton< T, ID, Registry >::~OsclSingleton () [inline]

6.212.2 Member Function Documentation

6.212.2.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> T& OsclSingleton< T, ID, Registry >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclSingleton can be used like the regular pointer that it was initialized with.

6.212.2.2 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> T* OsclSingleton< T, ID, Registry >::operator → () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclSingleton can be used like the regular pointer that it was initialized with.

**6.212.2.3 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> bool
OsclSingleton< T, ID, Registry >::set () [inline]**

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

6.212.3 Field Documentation

**6.212.3.1 template<class T, uint32 ID, class Registry = OsclSingletonRegistry> T*
OsclSingleton< T, ID, Registry >::_Ptr [protected]**

The documentation for this class was generated from the following file:

- [oscl_singleton.h](#)

6.213 OsclSingletonRegistry Class Reference

```
#include <oscl_singleton.h>
```

Static Public Methods

- OSCL_IMPORT_REF [OsclAny](#) * getInstance (uint32 ID, int32 &error)
- OSCL_IMPORT_REF void registerInstance ([OsclAny](#) *ptr, uint32 ID, int32 &error)
- OSCL_IMPORT_REF [OsclAny](#) * lockAndGetInstance (uint32 ID, int32 &error)
- OSCL_IMPORT_REF void registerInstanceAndUnlock ([OsclAny](#) *ptr, uint32 ID, int32 &error)

Friends

- class [OsclBase](#)

6.213.1 Member Function Documentation

6.213.1.1 OSCL_IMPORT_REF [OsclAny](#)* OsclSingletonRegistry::getInstance (uint32 *ID*, int32 & *error*) [static]

6.213.1.2 OSCL_IMPORT_REF [OsclAny](#)* OsclSingletonRegistry::lockAndGetInstance (uint32 *ID*, int32 & *error*) [static]

6.213.1.3 OSCL_IMPORT_REF void OsclSingletonRegistry::registerInstance ([OsclAny](#) * *ptr*, uint32 *ID*, int32 & *error*) [static]

6.213.1.4 OSCL_IMPORT_REF void OsclSingletonRegistry::registerInstanceAndUnlock ([OsclAny](#) * *ptr*, uint32 *ID*, int32 & *error*) [static]

6.213.2 Friends And Related Function Documentation

6.213.2.1 friend class OsclBase [friend]

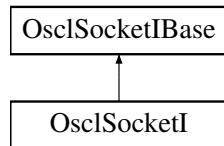
The documentation for this class was generated from the following file:

- [oscl_singleton.h](#)

6.214 OsclSocketI Class Reference

```
#include <oscl_socket_imp_pv.h>
```

Inheritance diagram for OsclSocketI::



Public Methods

- `~OsclSocketI ()`
- `int32 Open (OsclSocketServI &aServer, uint32 addrFamily, uint32 sockType, uint32 protocol)`
- `int32 Open (OsclSocketServI &aServer)`
- `int32 Bind (OsclNetworkAddress &anAddr)`
- `int32 Join (OsclNetworkAddress &anAddr)`
- `int32 Close ()`
- `int32 Listen (uint32 qSize)`
- `int32 SetRecvBufferSize (uint32 size)`
- `void Connect (ConnectParam &, OsclSocketRequestAO &)`
- `void Accept (AcceptParam &, OsclSocketRequestAO &)`
- `void Shutdown (ShutdownParam &, OsclSocketRequestAO &)`
- `void Send (SendParam &, OsclSocketRequestAO &)`
- `void SendSuccess (SendParam &)`
- `void SendTo (SendToParam &, OsclSocketRequestAO &)`
- `void SendToSuccess (SendToParam &)`
- `void Recv (RecvParam &, OsclSocketRequestAO &)`
- `void RecvSuccess (RecvParam &)`
- `void RecvFrom (RecvFromParam &, OsclSocketRequestAO &)`
- `void RecvFromSuccess (RecvFromParam &)`
- `TOsclSocket Socket ()`
- `void ProcessConnect (OsclSocketServRequestQElem *)`
- `void ProcessShutdown (OsclSocketServRequestQElem *)`
- `void ProcessAccept (OsclSocketServRequestQElem *)`
- `void ProcessSendTo (OsclSocketServRequestQElem *)`
- `void ProcessRecvFrom (OsclSocketServRequestQElem *)`
- `void ProcessSend (OsclSocketServRequestQElem *)`
- `void ProcessRecv (OsclSocketServRequestQElem *)`
- `PVLogger * Logger ()`

Static Public Methods

- `OsclSocketI * NewL (Oscl_DefAlloc &a)`
- `bool MakeAddr (OsclNetworkAddress &in, TOsclSockAddr &addr)`
- `void MakeAddr (TOsclSockAddr &in, OsclNetworkAddress &addr)`

Friends

- class [OsclAcceptRequest](#)
- class [OsclConnectRequest](#)
- class [OsclRecvRequest](#)
- class [OsclRecvFromRequest](#)
- class [OsclSendRequest](#)
- class [OsclSendToRequest](#)
- class [OsclShutdownRequest](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

6.214.1 Detailed Description

Socket implementation class

6.214.2 Constructor & Destructor Documentation

6.214.2.1 [OsclSocketI::~OsclSocketI \(\)](#)

6.214.3 Member Function Documentation

6.214.3.1 [void OsclSocketI::Accept \(\[AcceptParam\]\(#\) &, \[OsclSocketRequestAO\]\(#\) &\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

6.214.3.2 [int32 OsclSocketI::Bind \(\[OsclNetworkAddress\]\(#\) & *anAddr*\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

6.214.3.3 [int32 OsclSocketI::Close \(\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

6.214.3.4 [void OsclSocketI::Connect \(\[ConnectParam\]\(#\) &, \[OsclSocketRequestAO\]\(#\) &\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

6.214.3.5 [int32 OsclSocketI::Join \(\[OsclNetworkAddress\]\(#\) & *anAddr*\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

6.214.3.6 [int32 OsclSocketI::Listen \(uint32 *qSize*\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

6.214.3.7 **PVLogger*** OsclSocketI::Logger () [inline]

6.214.3.8 **void** OsclSocketI::MakeAddr (**TOsclSockAddr** & *in*, **OsclNetworkAddress** & *addr*)
[static]

6.214.3.9 **bool** OsclSocketI::MakeAddr (**OsclNetworkAddress** & *in*, **TOsclSockAddr** & *addr*)
[static]

6.214.3.10 **OsclSocketI*** OsclSocketI::NewL (**Oscl_DefAlloc** & *a*) [static]

6.214.3.11 **int32** OsclSocketI::Open (**OsclSocketServI** & *aServer*) [virtual]

Implements [OsclSocketIBase](#).

6.214.3.12 **int32** OsclSocketI::Open (**OsclSocketServI** & *aServer*, **uint32** *addrFamily*, **uint32**
sockType, **uint32** *protocol*) [virtual]

Implements [OsclSocketIBase](#).

6.214.3.13 **void** OsclSocketI::ProcessAccept (**OsclSocketServRequestQElem** *)

6.214.3.14 **void** OsclSocketI::ProcessConnect (**OsclSocketServRequestQElem** *)

6.214.3.15 **void** OsclSocketI::ProcessRecv (**OsclSocketServRequestQElem** *)

6.214.3.16 **void** OsclSocketI::ProcessRecvFrom (**OsclSocketServRequestQElem** *)

6.214.3.17 **void** OsclSocketI::ProcessSend (**OsclSocketServRequestQElem** *)

6.214.3.18 **void** OsclSocketI::ProcessSendTo (**OsclSocketServRequestQElem** *)

6.214.3.19 **void** OsclSocketI::ProcessShutdown (**OsclSocketServRequestQElem** *)

6.214.3.20 **void** OsclSocketI::Recv (**RecvParam** &, **OsclSocketRequestAO** &) [virtual]

Implements [OsclSocketIBase](#).

6.214.3.21 **void** OsclSocketI::RecvFrom (**RecvFromParam** &, **OsclSocketRequestAO** &)
[virtual]

Implements [OsclSocketIBase](#).

6.214.3.22 **void** OsclSocketI::RecvFromSuccess (**RecvFromParam** &) [virtual]

Implements [OsclSocketIBase](#).

6.214.3.23 **void** OsclSocketI::RecvSuccess (**RecvParam** &) [virtual]

Implements [OsclSocketIBase](#).

6.214.3.24 void OsclSocketI::Send ([SendParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

6.214.3.25 void OsclSocketI::SendSuccess ([SendParam](#) &) [virtual]

Implements [OsclSocketIBase](#).

6.214.3.26 void OsclSocketI::SendTo ([SendToParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

6.214.3.27 void OsclSocketI::SendToSuccess ([SendToParam](#) &) [virtual]

Implements [OsclSocketIBase](#).

6.214.3.28 int32 OsclSocketI::SetRecvBufferSize (uint32 *size*)

6.214.3.29 void OsclSocketI::Shutdown ([ShutdownParam](#) &, [OsclSocketRequestAO](#) &) [virtual]

Implements [OsclSocketIBase](#).

6.214.3.30 [TOsclSocket](#) OsclSocketI::Socket () [inline]

6.214.4 Friends And Related Function Documentation

6.214.4.1 friend class OsclAcceptRequest [friend]

6.214.4.2 friend class OsclConnectRequest [friend]

6.214.4.3 friend class OsclRecvFromRequest [friend]

6.214.4.4 friend class OsclRecvRequest [friend]

6.214.4.5 friend class OsclSendRequest [friend]

6.214.4.6 friend class OsclSendToRequest [friend]

6.214.4.7 friend class OsclShutdownRequest [friend]

6.214.4.8 friend class OsclTCPSocket [friend]

Reimplemented from [OsclSocketIBase](#).

6.214.4.9 friend class OsclUDPSocket [friend]

Reimplemented from [OsclSocketIBase](#).

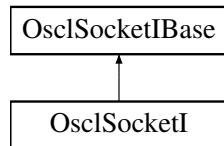
The documentation for this class was generated from the following file:

- [oscl_socket_imp_pv.h](#)

6.215 OsclSocketIBase Class Reference

```
#include <oscl_socket_imp_base.h>
```

Inheritance diagram for OsclSocketIBase::



Public Methods

- virtual ~OsclSocketIBase ()
- virtual int32 Open (OsclSocketServI &aServer, uint32 addrFamily, uint32 sockType, uint32 proto-col)=0
- virtual int32 Open (OsclSocketServI &aServer)=0
- virtual int32 Bind (OsclNetworkAddress &anAddr)=0
- virtual int32 Join (OsclNetworkAddress &anAddr)=0
- virtual int32 Close ()=0
- virtual int32 Listen (uint32 qSize)=0
- virtual void Connect (ConnectParam &, OsclSocketRequestAO &)=0
- virtual void Accept (AcceptParam &, OsclSocketRequestAO &)=0
- virtual void Shutdown (ShutdownParam &, OsclSocketRequestAO &)=0
- virtual void Send (SendParam &, OsclSocketRequestAO &)=0
- virtual void SendSuccess (SendParam &)=0
- virtual void SendTo (SendToParam &, OsclSocketRequestAO &)=0
- virtual void SendToSuccess (SendToParam &)=0
- virtual void Recv (RecvParam &, OsclSocketRequestAO &)=0
- virtual void RecvSuccess (RecvParam &)=0
- virtual void RecvFrom (RecvFromParam &, OsclSocketRequestAO &)=0
- virtual void RecvFromSuccess (RecvFromParam &)=0
- virtual void BindAsync (BindParam &, OsclSocketRequestAO &)
- virtual void ListenAsync (ListenParam &, OsclSocketRequestAO &)
- void CancelFxn (TPVSocketFxn)

Static Public Methods

- bool HasAsyncBind ()
- bool HasAsyncListen ()

Protected Methods

- OsclSocketIBase (Oscl_DefAlloc &a)
- virtual void CancelConnect ()=0
- virtual void CancelAccept ()=0
- virtual void CancelShutdown ()=0
- virtual void CancelSend ()=0

- virtual void [CancelSendTo \(\)=0](#)
- virtual void [CancelRecv \(\)=0](#)
- virtual void [CancelRecvFrom \(\)=0](#)
- virtual void [CancelBind \(\)](#)
- virtual void [CancelListen \(\)](#)
- virtual bool [IsOpen \(\)=0](#)

Static Protected Methods

- int [GetShutdown \(TPVSocketShutdown aOsclVal\)](#)

Protected Attributes

- [Oscl_DefAlloc & iAlloc](#)
- [OsclSocketServI * iSocketServ](#)

Friends

- class [OsclSocketRequest](#)
- class [OsclSocketMethod](#)
- class [OsclSocketRequestAO](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

6.215.1 Detailed Description

Socket implementation base class

6.215.2 Constructor & Destructor Documentation

6.215.2.1 virtual OsclSocketIBase::~OsclSocketIBase () [virtual]

6.215.2.2 OsclSocketIBase::OsclSocketIBase ([Oscl_DefAlloc & a](#)) [protected]

6.215.3 Member Function Documentation

6.215.3.1 virtual void OsclSocketIBase::Accept ([AcceptParam &](#), [OsclSocketRequestAO &](#)) [pure virtual]

Implemented in [OsclSocketI](#).

6.215.3.2 virtual int32 OsclSocketIBase::Bind ([OsclNetworkAddress & anAddr](#)) [pure virtual]

Implemented in [OsclSocketI](#).

- 6.215.3.3 **virtual void OsclSocketIBase::BindAsync (BindParam &, OsclSocketRequestAO &)**
[inline, virtual]
- 6.215.3.4 **virtual void OsclSocketIBase::CancelAccept ()** [protected, pure virtual]
- 6.215.3.5 **virtual void OsclSocketIBase::CancelBind ()** [inline, protected, virtual]
- 6.215.3.6 **virtual void OsclSocketIBase::CancelConnect ()** [protected, pure virtual]
- 6.215.3.7 **void OsclSocketIBase::CancelFxn (TPVSocketFxn)**
- 6.215.3.8 **virtual void OsclSocketIBase::CancelListen ()** [inline, protected, virtual]
- 6.215.3.9 **virtual void OsclSocketIBase::CancelRecv ()** [protected, pure virtual]
- 6.215.3.10 **virtual void OsclSocketIBase::CancelRecvFrom ()** [protected, pure virtual]
- 6.215.3.11 **virtual void OsclSocketIBase::CancelSend ()** [protected, pure virtual]
- 6.215.3.12 **virtual void OsclSocketIBase::CancelSendTo ()** [protected, pure virtual]
- 6.215.3.13 **virtual void OsclSocketIBase::CancelShutdown ()** [protected, pure virtual]
- 6.215.3.14 **virtual int32 OsclSocketIBase::Close ()** [pure virtual]

Implemented in [OsclSocketI](#).

- 6.215.3.15 **virtual void OsclSocketIBase::Connect (ConnectParam &, OsclSocketRequestAO &)**
[pure virtual]

Implemented in [OsclSocketI](#).

- 6.215.3.16 **int OsclSocketIBase::GetShutdown (TPVSocketShutdown aOsclVal)** [static, protected]
- 6.215.3.17 **bool OsclSocketIBase::HasAsyncBind ()** [static]
- 6.215.3.18 **bool OsclSocketIBase::HasAsyncListen ()** [static]
- 6.215.3.19 **virtual bool OsclSocketIBase::IsOpen ()** [protected, pure virtual]
- 6.215.3.20 **virtual int32 OsclSocketIBase::Join (OsclNetworkAddress & anAddr)** [pure virtual]

Implemented in [OsclSocketI](#).

- 6.215.3.21 **virtual int32 OsclSocketIBase::Listen (uint32 qSize)** [pure virtual]

Implemented in [OsclSocketI](#).

6.215.3.22 `virtual void OsclSocketIBase::ListenAsync (ListenParam &, OsclSocketRequestAO &)`
[`inline`, `virtual`]

6.215.3.23 `virtual int32 OsclSocketIBase::Open (OsclSocketServI & aServer)` [pure
`virtual`]

Implemented in [OsclSocketI](#).

6.215.3.24 `virtual int32 OsclSocketIBase::Open (OsclSocketServI & aServer, uint32 addrFamily,`
`uint32 sockType, uint32 protocol)` [pure `virtual`]

Implemented in [OsclSocketI](#).

6.215.3.25 `virtual void OsclSocketIBase::Recv (RecvParam &, OsclSocketRequestAO &)` [pure
`virtual`]

Implemented in [OsclSocketI](#).

6.215.3.26 `virtual void OsclSocketIBase::RecvFrom (RecvFromParam &, OsclSocketRequestAO &)` [pure
`virtual`]

Implemented in [OsclSocketI](#).

6.215.3.27 `virtual void OsclSocketIBase::RecvFromSuccess (RecvFromParam &)` [pure
`virtual`]

Implemented in [OsclSocketI](#).

6.215.3.28 `virtual void OsclSocketIBase::RecvSuccess (RecvParam &)` [pure `virtual`]

Implemented in [OsclSocketI](#).

6.215.3.29 `virtual void OsclSocketIBase::Send (SendParam &, OsclSocketRequestAO &)` [pure
`virtual`]

Implemented in [OsclSocketI](#).

6.215.3.30 `virtual void OsclSocketIBase::SendSuccess (SendParam &)` [pure `virtual`]

Implemented in [OsclSocketI](#).

6.215.3.31 `virtual void OsclSocketIBase::SendTo (SendToParam &, OsclSocketRequestAO &)`
[pure `virtual`]

Implemented in [OsclSocketI](#).

6.215.3.32 virtual void OsclSocketIBase::SendToSuccess ([SendToParam](#) &) [pure virtual]

Implemented in [OsclSocketI](#).

6.215.3.33 virtual void OsclSocketIBase::Shutdown ([ShutdownParam](#) &, [OsclSocketRequestAO](#) &) [pure virtual]

Implemented in [OsclSocketI](#).

6.215.4 Friends And Related Function Documentation

6.215.4.1 friend class OsclSocketMethod [friend]

6.215.4.2 friend class OsclSocketRequest [friend]

6.215.4.3 friend class OsclSocketRequestAO [friend]

6.215.4.4 friend class OsclTCPSocket [friend]

Reimplemented in [OsclSocketI](#).

6.215.4.5 friend class OsclUDPSocket [friend]

Reimplemented in [OsclSocketI](#).

6.215.5 Field Documentation

6.215.5.1 [Oscl_DefAlloc](#)& OsclSocketIBase::iAlloc [protected]

6.215.5.2 [OsclSocketServI](#)* OsclSocketIBase::iSocketServ [protected]

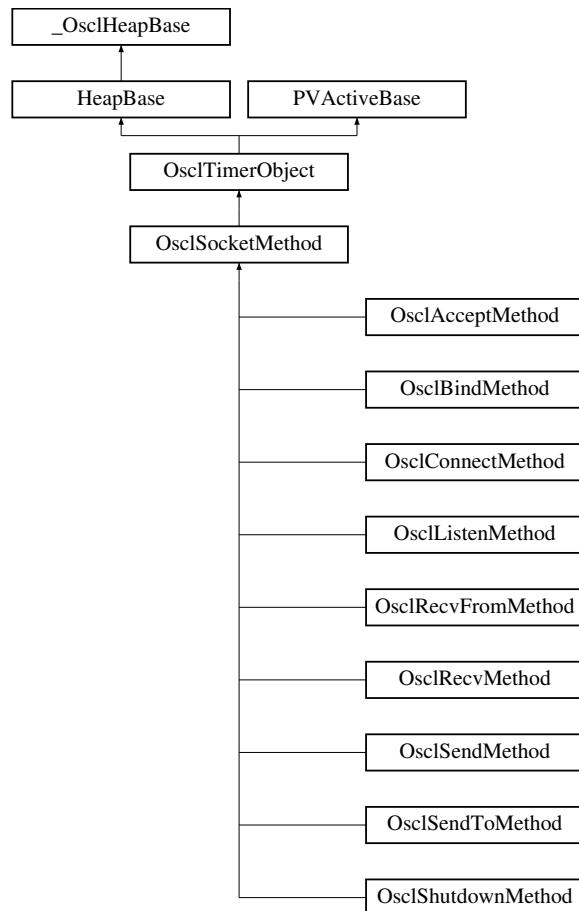
The documentation for this class was generated from the following file:

- [oscl_socket_imp_base.h](#)

6.216 OsclSocketMethod Class Reference

```
#include <oscl_socket_method.h>
```

Inheritance diagram for OsclSocketMethod::



Public Methods

- [OsclSocketMethod \(OsclIPSocketI &aContainer, const char *name, TPVSocketFxn fxn\)](#)
- virtual [~OsclSocketMethod \(\)](#)
- void [Abort \(\)](#)
- void [AbortAll \(\)](#)
- void [CancelMethod \(\)](#)
- [Oscl_DefAlloc & Alloc \(\)](#)

Data Fields

- [OsclIPSocketI & iContainer](#)
- [TPVSocketFxn iSocketFxn](#)

Protected Methods

- void [ConstructL \(OsclSocketRequestAO *aAO\)](#)
- bool [StartMethod \(int32 aTimeoutMsec\)](#)
- void [MethodDone \(\)](#)
- void [Run \(\)](#)

Protected Attributes

- [OsclSocketRequestAO * iSocketRequestAO](#)

6.216.1 Detailed Description

OsclSocketMethod is the base class for all socket methods. Two AOs are required for each socket operation— one to provide a timeout, and one to detect request completion. The OsclSocketMethod class implements the timeout and contains the request completion AO.

6.216.2 Constructor & Destructor Documentation

6.216.2.1 OsclSocketMethod::OsclSocketMethod ([OsclIPSocketI & aContainer](#), [const char * name](#), [TPVSocketFxn ffn](#)) [inline]

6.216.2.2 virtual OsclSocketMethod::~OsclSocketMethod () [inline, virtual]

6.216.3 Member Function Documentation

6.216.3.1 void OsclSocketMethod::Abort () [inline]

6.216.3.2 void OsclSocketMethod::AbortAll () [inline]

6.216.3.3 Oscl_DefAlloc& OsclSocketMethod::Alloc () [inline]

6.216.3.4 void OsclSocketMethod::CancelMethod () [inline]

6.216.3.5 void OsclSocketMethod::ConstructL (OsclSocketRequestAO * aAO) [inline, protected]

6.216.3.6 void OsclSocketMethod::MethodDone () [inline, protected]

6.216.3.7 void OsclSocketMethod::Run () [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request

2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls [ExecError\(\)](#) to handle the leave.

Note that once the active scheduler's [Start\(\)](#) function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

6.216.3.8 bool OsclSocketMethod::StartMethod (int32 *aTimeoutMsec*) [protected]

6.216.4 Field Documentation

6.216.4.1 [OsclIPSocketI&](#) OsclSocketMethod::iContainer

6.216.4.2 [TPVSocketFxn](#) OsclSocketMethod::iSocketFxn

6.216.4.3 [OsclSocketRequestAO*](#) OsclSocketMethod::iSocketRequestAO [protected]

The documentation for this class was generated from the following file:

- [oscl_socket_method.h](#)

6.217 OsclSocketObserver Class Reference

```
#include <oscl_socket_types.h>
```

Public Methods

- virtual OSCL_IMPORT_REF void [HandleSocketEvent](#) (int32 *aId*, [TPVSocketFxn](#) *aFxn*, [TPVSocketEvent](#) *aEvent*, int32 *aError*)=0
- virtual ~[OsclSocketObserver](#) ()

6.217.1 Detailed Description

Socket event observer. The client implements this to get asynchronous command completion.

6.217.2 Constructor & Destructor Documentation

6.217.2.1 virtual OsclSocketObserver::~OsclSocketObserver () [inline, virtual]

6.217.3 Member Function Documentation

6.217.3.1 virtual OSCL_IMPORT_REF void OsclSocketObserver::HandleSocketEvent (int32 *aId*, [TPVSocketFxn](#) *aFxn*, [TPVSocketEvent](#) *aEvent*, int32 *aError*) [pure virtual]

Socket Event callback.

Parameters:

***aId*:** The ID that was supplied when the socket was created.

***aFxn*:** Type of socket function call.

***aEvent*:** Function completion event. Will be EPVSocketSuccess, EPVSocketTimeout, or EPVSocketFailure.

***aError*:** When the event is EPVSocketFailure, this may contain a platform-specific error code, or zero if none is available.

The documentation for this class was generated from the following file:

- [oscl_socket_types.h](#)

6.218 OsclSocketRequest Class Reference

```
#include <oscl_socket_request.h>
```

Public Methods

- [OsclSocketRequest \(\)](#)
- [TPVSocketFxn Fxn \(\)](#)
- void [CancelRequest \(\)](#)
- void [Activate \(SocketRequestParam *iParam, OsclSocketRequestAO &a\)](#)
- void [Complete \(OsclSocketServRequestQElem *, int32 aStatus, int32 aSockErr=0\)](#)

Data Fields

- [OsclSocketRequestAO * iSocketRequestAO](#)
- [SocketRequestParam * iParam](#)
- [OsclSocketI * iSocketI](#)

6.218.1 Detailed Description

This class defines the request interface to the PV socket server.

6.218.2 Constructor & Destructor Documentation

6.218.2.1 [OsclSocketRequest::OsclSocketRequest \(\) \[inline\]](#)

6.218.3 Member Function Documentation

6.218.3.1 [void OsclSocketRequest::Activate \(SocketRequestParam * iParam, OsclSocketRequestAO & a\)](#)

6.218.3.2 [void OsclSocketRequest::CancelRequest \(\)](#)

6.218.3.3 [void OsclSocketRequest::Complete \(OsclSocketServRequestQElem *, int32 aStatus, int32 aSockErr = 0\)](#)

6.218.3.4 [TPVSocketFxn OsclSocketRequest::Fxn \(\) \[inline\]](#)

6.218.4 Field Documentation

6.218.4.1 [SocketRequestParam* OsclSocketRequest::iParam](#)

6.218.4.2 [OsclSocketI* OsclSocketRequest::iSocketI](#)

6.218.4.3 [OsclSocketRequestAO* OsclSocketRequest::iSocketRequestAO](#)

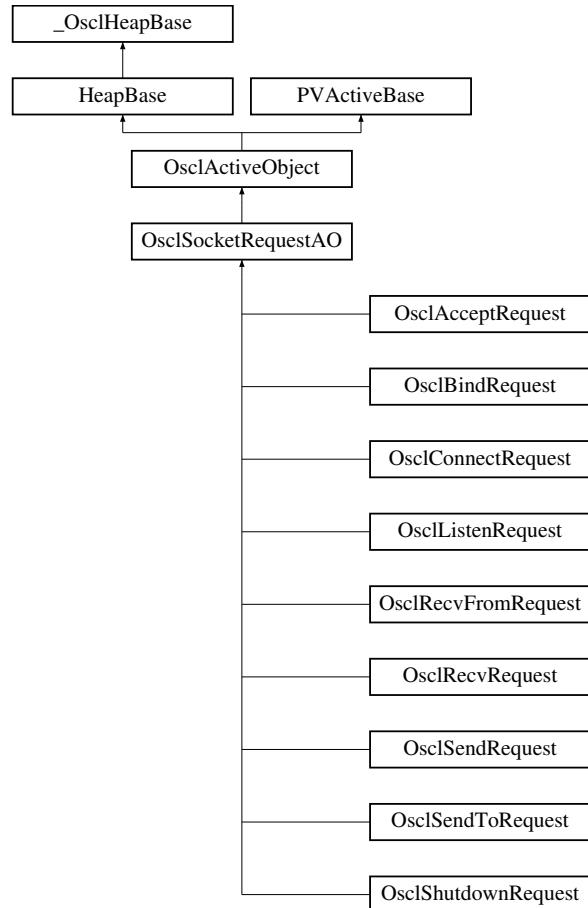
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

6.219 OsclSocketRequestAO Class Reference

```
#include <oscl_socket_method.h>
```

Inheritance diagram for OsclSocketRequestAO::



Public Methods

- void [ConstructL \(\)](#)

Protected Methods

- [OsclSocketRequestAO \(OsclSocketMethod &aContainer, const char *name\)](#)
- virtual [~OsclSocketRequestAO \(\)](#)
- [OsclAny * NewRequest \(const uint32 size\)](#)
- void [CleanupParam \(bool deallocate=false\)](#)
- void [Abort \(\)](#)
- void [RequestDone \(\)](#)
- int [GetSocketError \(\)](#)
- void [DoCancel \(\)](#)
- void [Run \(\)](#)

- virtual void [Success \(\)](#)
- [OsclSocketI * SocketI \(\)](#)
- [OsclSocketObserver * SocketObserver \(\)](#)
- uint32 [Id \(\)](#)
- [Oscl_DefAlloc & Alloc \(\)](#)

Protected Attributes

- [OsclSocketMethod & iContainer](#)
- int32 [iSocketError](#)
- [SocketRequestParam * iParam](#)
- uint32 [iParamSize](#)

Friends

- class [OsclSocketI](#)
- class [OsclSocketMethod](#)
- class [OsclSocketRequest](#)

6.219.1 Detailed Description

This is the base class for all the AOs that interact with the socket server. This object is contained within an [OsclSocketMethod](#) object

6.219.2 Constructor & Destructor Documentation

6.219.2.1 OsclSocketRequestAO::OsclSocketRequestAO ([OsclSocketMethod & aContainer](#), const char * *name*) [inline, protected]

6.219.2.2 virtual OsclSocketRequestAO::~OsclSocketRequestAO () [inline, protected, virtual]

6.219.3 Member Function Documentation

6.219.3.1 void OsclSocketRequestAO::Abort () [inline, protected]

6.219.3.2 [Oscl_DefAlloc& OsclSocketRequestAO::Alloc \(\) \[inline, protected\]](#)

6.219.3.3 void OsclSocketRequestAO::CleanupParam (bool *deallocate* = false) [protected]

6.219.3.4 void OsclSocketRequestAO::ConstructL () [inline]

6.219.3.5 void OsclSocketRequestAO::DoCancel () [inline, protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Reimplemented from [OsclActiveObject](#).

6.219.3.6 int OsclSocketRequestAO::GetSocketError () [inline, protected]

6.219.3.7 uint32 OsclSocketRequestAO::Id () [inline, protected]

6.219.3.8 OsclAny* OsclSocketRequestAO::NewRequest (const uint32 *size*) [protected]

6.219.3.9 void OsclSocketRequestAO::RequestDone () [inline, protected]

6.219.3.10 void OsclSocketRequestAO::Run () [protected, virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's Run() function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

Run() runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's Run() or RunError() functions.

Implements [PVActiveBase](#).

6.219.3.11 OsclSocketI* OsclSocketRequestAO::SocketI () [inline, protected]

6.219.3.12 OsclSocketObserver* OsclSocketRequestAO::SocketObserver () [inline, protected]

6.219.3.13 virtual void OsclSocketRequestAO::Success () [inline, protected, virtual]

Reimplemented in [OsclRecvRequest](#), [OsclRecvFromRequest](#), [OsclSendRequest](#), and [OsclSendToRequest](#).

6.219.4 Friends And Related Function Documentation

6.219.4.1 **friend class OsclSocketI** [friend]

6.219.4.2 **friend class OsclSocketMethod** [friend]

6.219.4.3 **friend class OsclSocketRequest** [friend]

6.219.5 Field Documentation

6.219.5.1 **OsclSocketMethod& OsclSocketRequestAO::iContainer** [protected]

6.219.5.2 **SocketRequestParam* OsclSocketRequestAO::iParam** [protected]

6.219.5.3 **uint32 OsclSocketRequestAO::iParamSize** [protected]

6.219.5.4 **int32 OsclSocketRequestAO::iSocketError** [protected]

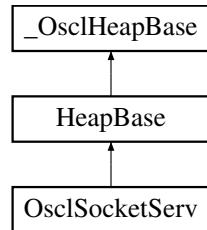
The documentation for this class was generated from the following file:

- [oscl_socket_method.h](#)

6.220 OsclSocketServ Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclSocketServ::



Public Methods

- OSCL_IMPORT_REF ~OsclSocketServ ()
- OSCL_IMPORT_REF int32 Connect (uint32 aMessageSlots=8)
- OSCL_IMPORT_REF void Close (bool aCleanup=true)

Static Public Methods

- OSCL_IMPORT_REF OsclSocketServ * NewL (Oscl_DefAlloc &alloc)

Friends

- class OsclTCPSocket
- class OsclUDPSocket
- class OsclDNS

6.220.1 Constructor & Destructor Documentation

6.220.1.1 OSCL_IMPORT_REF OsclSocketServ::~OsclSocketServ ()

Destructor. The server object must be deleted using the same allocator used in the NewL call.

6.220.2 Member Function Documentation

6.220.2.1 OSCL_IMPORT_REF void OsclSocketServ::Close (bool aCleanup = true)

Close socket server. This is a synchronous method.

Parameters:

aCleanup: cleanup the socket system? the socket cleanup should only be done when all parts of the application are done using sockets.

6.220.2.2 OSCL_IMPORT_REF int32 OsclSocketServ::Connect (uint32 *aMessageSlots* = 8)

Connect to socket server. This is a synchronous method.

Parameters:

Number of message slots.

Returns:

Returns OsclErrNone for success, or a platform-specific code.

**6.220.2.3 OSCL_IMPORT_REF OsclSocketServ* OsclSocketServ::NewL (Oscl_DefAlloc & *alloc*)
[static]**

Create a socket server. May leave if failure.

Parameters:

alloc: Memory allocator.

Returns:

Returns pointer to socket server

6.220.3 Friends And Related Function Documentation**6.220.3.1 friend class OsclDNS [friend]****6.220.3.2 friend class OsclTCPSocket [friend]****6.220.3.3 friend class OsclUDPSocket [friend]**

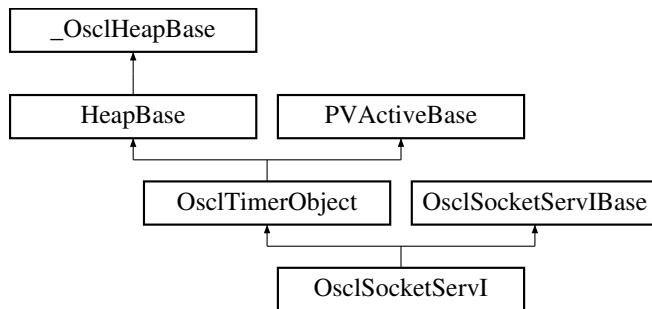
The documentation for this class was generated from the following file:

- [oscl_socket.h](#)

6.221 OsclSocketServI Class Reference

```
#include <oscl_socket_serv_imp_pv.h>
```

Inheritance diagram for OsclSocketServI::



Public Methods

- int32 [Connect](#) (uint32 aMessageSlots)
- void [Close](#) (bool)
- bool [IsServerThread](#) ()

Static Public Methods

- OsclSocketServI * [NewL](#) (Oscl_DefAlloc &a)

Friends

- class [OsclSocketServRequestList](#)
- class [LoopbackSocket](#)
- class [OsclTCPSocketI](#)
- class [OsclUDPSocketI](#)
- class [OsclSocketI](#)
- class [OsclDNSI](#)
- class [OsclSocketRequest](#)
- class [OsclSocketServ](#)

6.221.1 Detailed Description

PV socket server implementation

6.221.2 Member Function Documentation

6.221.2.1 void OsclSocketServI::Close (bool) [virtual]

Implements [OsclSocketServIBase](#).

6.221.2.2 int32 OsclSocketServI::Connect (uint32 *aMessageSlots*) [virtual]

Implements [OsclSocketServIBase](#).

6.221.2.3 bool OsclSocketServI::IsServerThread ()**6.221.2.4 OsclSocketServI* OsclSocketServI::NewL (Oscl_DefAlloc & *a*) [static]****6.221.3 Friends And Related Function Documentation****6.221.3.1 friend class LoopbackSocket [friend]****6.221.3.2 friend class OsclDNSI [friend]****6.221.3.3 friend class OsclSocketI [friend]****6.221.3.4 friend class OsclSocketRequest [friend]****6.221.3.5 friend class OsclSocketServ [friend]****6.221.3.6 friend class OsclSocketServRequestList [friend]****6.221.3.7 friend class OsclTCPSocketI [friend]****6.221.3.8 friend class OsclUDPSocketI [friend]**

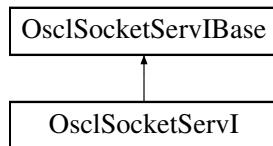
The documentation for this class was generated from the following file:

- [oscl_socket_serv_imp_pv.h](#)

6.222 OsclSocketServIBase Class Reference

```
#include <oscl_socket_serv_imp_base.h>
```

Inheritance diagram for OsclSocketServIBase::



Public Methods

- virtual ~[OsclSocketServIBase](#) ()
- virtual int32 [Connect](#) (uint32 aMessageSlots)=0
- virtual void [Close](#) (bool)=0

Data Fields

- [PVLogger * iLogger](#)

Protected Types

- enum [TSocketServState](#) { [ESocketServ_Idle](#), [ESocketServ_Connected](#), [ESocketServ_Error](#) }

Protected Methods

- [OsclSocketServIBase](#) ([Oscl_DefAlloc](#) &a)
- [TSocketServState State](#) () const
- bool [IsServConnected](#) () const

Protected Attributes

- [Oscl_DefAlloc](#) & [iAlloc](#)
- [TSocketServState](#) [iServState](#)
- int [iServError](#)

6.222.1 Detailed Description

Base class common to all implementations

6.222.2 Member Enumeration Documentation

6.222.2.1 enum OsclSocketServIBase::TSocketServState [protected]

Enumeration values:

[ESocketServ_Idle](#)

ESocketServ_Connected

ESocketServ_Error

6.222.3 Constructor & Destructor Documentation

6.222.3.1 virtual OsclSocketServIBase::~OsclSocketServIBase () [inline, virtual]

6.222.3.2 OsclSocketServIBase::OsclSocketServIBase ([Oscl_DefAlloc](#) & *a*) [inline, protected]

6.222.4 Member Function Documentation

6.222.4.1 virtual void OsclSocketServIBase::Close (bool) [pure virtual]

Implemented in [OsclSocketServI](#).

6.222.4.2 virtual int32 OsclSocketServIBase::Connect (uint32 *aMessageSlots*) [pure virtual]

Implemented in [OsclSocketServI](#).

6.222.4.3 bool OsclSocketServIBase::IsServConnected () const [inline, protected]

6.222.4.4 [TSocketServState](#) OsclSocketServIBase::State () const [inline, protected]

6.222.5 Field Documentation

6.222.5.1 [Oscl_DefAlloc](#)& OsclSocketServIBase::iAlloc [protected]

6.222.5.2 [PVLogger](#)* OsclSocketServIBase::iLogger

6.222.5.3 int OsclSocketServIBase::iServerError [protected]

6.222.5.4 [TSocketServState](#) OsclSocketServIBase::iServState [protected]

The documentation for this class was generated from the following file:

- [oscl_socket_serv_imp_base.h](#)

6.223 OsclSocketServRequestList Class Reference

```
#include <oscl_socket_serv_imp_reqlist.h>
```

Public Methods

- [OsclSocketServRequestList \(\)](#)
- [void Add \(OsclSocketRequest *\)](#)
- [void StartCancel \(OsclSocketRequest *\)](#)
- [void Open \(OsclSocketServI *s\)](#)
- [void Close \(\)](#)
- [void Wakeup \(\)](#)
- [void WaitOnRequests \(\)](#)
- [void Remove \(OsclSocketServRequestQElem *aElem\)](#)

Friends

- class [OsclSocketServI](#)

6.223.1 Detailed Description

PV socket server request queue

6.223.2 Constructor & Destructor Documentation

6.223.2.1 OsclSocketServRequestList::OsclSocketServRequestList ()

6.223.3 Member Function Documentation

6.223.3.1 void OsclSocketServRequestList::Add ([OsclSocketRequest *](#))

6.223.3.2 void OsclSocketServRequestList::Close ()

6.223.3.3 void OsclSocketServRequestList::Open ([OsclSocketServI * s](#))

6.223.3.4 void OsclSocketServRequestList::Remove ([OsclSocketServRequestQElem * aElem](#)) [inline]

6.223.3.5 void OsclSocketServRequestList::StartCancel ([OsclSocketRequest *](#))

6.223.3.6 void OsclSocketServRequestList::WaitOnRequests ()

6.223.3.7 void OsclSocketServRequestList::Wakeup ()

6.223.4 Friends And Related Function Documentation

6.223.4.1 friend class OsclSocketServI [friend]

The documentation for this class was generated from the following file:

-
- [oscl_socket_serv_imp_reqlist.h](#)

6.224 OsclSocketServRequestQElem Class Reference

```
#include <oscl_socket_serv_imp_reqlist.h>
```

Public Methods

- [OsclSocketServRequestQElem \(OsclSocketRequest *r\)](#)

Data Fields

- [OsclSocketRequest * iSocketRequest](#)
- [uint8 iSelect](#)
- [bool iCancel](#)

6.224.1 Constructor & Destructor Documentation

6.224.1.1 [OsclSocketServRequestQElem::OsclSocketServRequestQElem \(OsclSocketRequest * r\)](#)
[inline]

6.224.2 Field Documentation

6.224.2.1 [bool OsclSocketServRequestQElem::iCancel](#)

6.224.2.2 [uint8 OsclSocketServRequestQElem::iSelect](#)

6.224.2.3 [OsclSocketRequest* OsclSocketServRequestQElem::iSocketRequest](#)

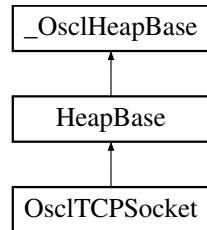
The documentation for this class was generated from the following file:

- [oscl_socket_serv_imp_reqlist.h](#)

6.225 OsclTCPSocket Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclTCPSocket::



Public Methods

- OSCL_IMPORT_REF ~OsclTCPSocket ()
- OSCL_IMPORT_REF int32 Close ()
- OSCL_IMPORT_REF int32 Bind (OsclNetworkAddress &aAddress)
- OSCL_IMPORT_REF TPVSocketEvent BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL_IMPORT_REF void CancelBind ()
- OSCL_IMPORT_REF int32 Listen (int32 aQueueSize)
- OSCL_IMPORT_REF TPVSocketEvent ListenAsync (int32 aQueueSize, int32 aTimeoutMsec=(-1))
- OSCL_IMPORT_REF void CancelListen ()
- OSCL_IMPORT_REF OsclTCPSocket * GetAcceptedSocketL (uint32 aId)
- OSCL_IMPORT_REF uint8 * GetRecvData (int32 *aLength)
- OSCL_IMPORT_REF uint8 * GetSendData (int32 *aLength)
- OSCL_IMPORT_REF TPVSocketEvent Connect (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelConnect ()
- OSCL_IMPORT_REF TPVSocketEvent Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelShutdown ()
- OSCL_IMPORT_REF TPVSocketEvent Accept (int32 aTimeout=-1)
- OSCL_IMPORT_REF void CancelAccept ()
- OSCL_IMPORT_REF TPVSocketEvent Send (const uint8 *aPtr, uint32 aLen, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelSend ()
- OSCL_IMPORT_REF TPVSocketEvent Recv (uint8 *aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelRecv ()

Static Public Methods

- OSCL_IMPORT_REF OsclTCPSocket * NewL (Oscl_DefAlloc &alloc, OsclSocketServ &aServ, OsclSocketObserver *aObserver, uint32 aId)

6.225.1 Detailed Description

The TCP Socket class

6.225.2 Constructor & Destructor Documentation

6.225.2.1 OSCL_IMPORT_REF OsclTCPSocket::~OsclTCPSocket ()

Destructor. The object must be deleted using the same allocator used in the NewL call.

6.225.3 Member Function Documentation

6.225.3.1 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::Accept (int32 *aTimeout* = -1)

Accept incoming connections. This is an asynchronous method.

Parameters:

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

6.225.3.2 OSCL_IMPORT_REF int32 OsclTCPSocket::Bind ([OsclNetworkAddress](#) & *aAddress*)

Bind a TCP socket to an address. This is a synchronous method.

Parameters:

aAddress: Bind address.

Returns:

Returns OsclErrNone for success, or a platform-specific error code.

6.225.3.3 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::BindAsync ([OsclNetworkAddress](#) & *aAddress*, int32 *aTimeoutMsec* = (-1))

Bind a TCP socket to an address. This is an asynchronous method.

Parameters:

aAddress: Bind address.

aTimeoutMsec: Optional timeout. Use a negative value for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

6.225.3.4 OSCL_IMPORT_REF void OsclTCPSocket::CancelAccept ()

Cancel Accept

This method will cancel any pending Accept operation on the current socket, causing the Accept to complete with error EPVSocketCancel. If there is no pending Accept operation, this method will have no effect.

6.225.3.5 OSCL_IMPORT_REF void OsclTCPSocket::CancelBind ()

Cancel Bind

This method will cancel any pending BindAsync operation on the current socket, causing the BindAsync to complete with error EPVSocketCancel. If there is no pending BindAsync operation, this method will have no effect.

6.225.3.6 OSCL_IMPORT_REF void OsclTCPSocket::CancelConnect ()

Cancel Connect

This method will cancel any pending Connect operation on the current socket, causing the Connect to complete with error EPVSocketCancel. If there is no pending Connect operation, this method will have no effect.

6.225.3.7 OSCL_IMPORT_REF void OsclTCPSocket::CancelListen ()

Cancel Async Listen

This method will cancel any pending ListenAsync operation on the current socket, causing the Listen to complete with error EPVSocketCancel. If there is no pending Listen operation, this method will have no effect.

6.225.3.8 OSCL_IMPORT_REF void OsclTCPSocket::CancelRecv ()

Cancel Recv

This method will cancel any pending Recv operation on the current socket, causing the Recv to complete with error EPVSocketCancel. If there is no pending Recv operation, this method will have no effect.

6.225.3.9 OSCL_IMPORT_REF void OsclTCPSocket::CancelSend ()

Cancel Send

This method will cancel any pending Send operation on the current socket, causing the Send to complete with error EPVSocketCancel. If there is no pending Send operation, this method will have no effect.

6.225.3.10 OSCL_IMPORT_REF void OsclTCPSocket::CancelShutdown ()

Cancel Shutdown

This method will cancel any pending Shutdown operation on the current socket, causing the Shutdown to complete with error EPVSocketCancel. If there is no pending Shutdown operation, this method will have no effect.

6.225.3.11 OSCL_IMPORT_REF int32 OsclTCPSocket::Close ()

Close a TCP socket. This is a synchronous method.

Once it is closed a socket cannot be re-opened. Sockets are automatically closed when they are deleted. This method may be used to see any error code returned from the platform's socket close call.

Returns:

Returns OsclErrNone for success, or a platform-specific error code.

**6.225.3.12 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::Connect
(OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1)**

Connect to an address. This is an asynchronous method.

Parameters:

aAddress: a network address.

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

6.225.3.13 OSCL_IMPORT_REF OsclTCPSocket* OsclTCPSocket::GetAcceptedSocketL (uint32 aId)

Retrieve the accept socket after a successful Accept operation. This is a synchronous method.

Parameters:

aId: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

Returns:

Returns pointer to socket, or NULL if error. Note: The caller is responsible for deleting any accepted socket that it retrieves.

6.225.3.14 OSCL_IMPORT_REF uint8* OsclTCPSocket::GetRecvData (int32 * aLength)

Retrieve the received data after a successful Recv operation. This is a synchronous method.

Parameters:

aLength: (output) number of bytes of data received.

Returns:

Returns pointer to received data, or NULL if none.

6.225.3.15 OSCL_IMPORT_REF uint8* OsclTCPSocket::GetSendData (int32 * aLength)

Retrieve the sent data after a successful Send operation. This is a synchronous method.

Parameters:

aLength: (output) number of bytes of data sent.

Returns:

Returns pointer to sent data, or NULL if none.

6.225.3.16 OSCL_IMPORT_REF int32 OsclTCPSocket::Listen (int32 aQueueSize)

Listen. This is a synchronous method.

Parameters:

aQueueSize: Queue size.

Returns:

Returns OsclErrNone for success, or a platform-specific error code.

6.225.3.17 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::ListenAsync (int32 aQueueSize, int32 aTimeoutMsec = (-1))

ListenAsync This is an asynchronous method.

Parameters:

aQueueSize: Queue size.

aTimeoutMsec: Optional timeout. Use a negative value for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

6.225.3.18 OSCL_IMPORT_REF OsclTCPSocket* OsclTCPSocket::NewL (Oscl_DefAlloc & alloc, OsclSocketServ & aServ, OsclSocketObserver * aObserver, uint32 aId) [static]

Create a TCP Socket. May leave if failure.

Parameters:

alloc: Memory allocator.

aServ: Socket server. Must be connected.

aObserver: Socket observer.

aId: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

Returns:

Returns pointer to socket.

6.225.3.19 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::Recv (uint8 * aPtr, uint32 aMaxLen, int32 aTimeoutMsec = -1)

Receive Data. This is an asynchronous method.

Parameters:

aPtr: Buffer for received data.

aMaxLen: Length of buffer.

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

6.225.3.20 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::Send (const uint8 * aPtr, uint32 aLen, int32 aTimeoutMsec = -1)

Send Data. This is an asynchronous method.

Parameters:

aPtr: Data to send.

aLen: Length of data to send.

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

6.225.3.21 OSCL_IMPORT_REF TPVSocketEvent OsclTCPSocket::Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec = -1)

Shutdown a socket. This is an asynchronous method.

Parameters:

aHow: type of shutdown

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

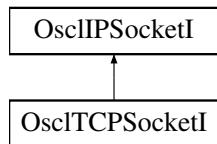
The documentation for this class was generated from the following file:

- [oscl_socket.h](#)

6.226 OsclTCPSocketI Class Reference

```
#include <oscl_tcp_socket.h>
```

Inheritance diagram for OsclTCPSocketI::



Public Methods

- virtual ~OsclTCPSocketI ()
- int32 [Close \(\)](#)
- int32 [Listen \(int aQueueSize\)](#)
- OsclTCPSocketI * [GetAcceptedSocketL \(uint32 aId\)](#)
- uint8 * [GetRecvData \(int32 *aLength\)](#)
- uint8 * [GetSendData \(int32 *aLength\)](#)
- [TPVSocketEvent BindAsync \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelBind \(\)](#)
- [TPVSocketEvent ListenAsync \(uint32 qsize, int32 aTimeoutMsec=-1\)](#)
- void [CancelListen \(\)](#)
- [TPVSocketEvent Connect \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelConnect \(\)](#)
- [TPVSocketEvent Shutdown \(TPVSocketShutdown aHow, int32 aTimeoutMsec=-1\)](#)
- void [CancelShutdown \(\)](#)
- [TPVSocketEvent Accept \(int32 aTimeout=-1\)](#)
- void [CancelAccept \(\)](#)
- [TPVSocketEvent Send \(const uint8 *&aPtr, uint32 aLen, int32 aTimeoutMsec=-1\)](#)
- void [CancelSend \(\)](#)
- [TPVSocketEvent Recv \(uint8 *&aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1\)](#)
- void [CancelRecv \(\)](#)

Static Public Methods

- OsclTCPSocketI * [NewL \(Oscl_DefAlloc &a, OsclSocketServI *aServ, OsclSocketObserver *aObserver, uint32 aId\)](#)

6.226.1 Detailed Description

Internal implementation class for [OsclTCPSocket](#)

6.226.2 Constructor & Destructor Documentation

6.226.2.1 **virtual OsclTCPSocketI::~OsclTCPSocketI () [virtual]**

6.226.3 Member Function Documentation

6.226.3.1 **TPVSocketEvent OsclTCPSocketI::Accept (int32 *aTimeout* = -1) [inline]**

6.226.3.2 **TPVSocketEvent OsclTCPSocketI::BindAsync (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1) [inline]**

6.226.3.3 **void OsclTCPSocketI::CancelAccept () [inline]**

6.226.3.4 **void OsclTCPSocketI::CancelBind () [inline]**

6.226.3.5 **void OsclTCPSocketI::CancelConnect () [inline]**

6.226.3.6 **void OsclTCPSocketI::CancelListen () [inline]**

6.226.3.7 **void OsclTCPSocketI::CancelRecv () [inline]**

6.226.3.8 **void OsclTCPSocketI::CancelSend () [inline]**

6.226.3.9 **void OsclTCPSocketI::CancelShutdown () [inline]**

6.226.3.10 **int32 OsclTCPSocketI::Close () [virtual]**

Implements [OsclIPSocketI](#).

6.226.3.11 **TPVSocketEvent OsclTCPSocketI::Connect (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1) [inline]**

6.226.3.12 **OsclTCPSocketI* OsclTCPSocketI::GetAcceptedSocketL (uint32 *aId*)**

6.226.3.13 **uint8 * OsclTCPSocketI::GetRecvData (int32 * *aLength*) [inline, virtual]**

Implements [OsclIPSocketI](#).

6.226.3.14 **uint8 * OsclTCPSocketI::GetSendData (int32 * *aLength*) [inline, virtual]**

Implements [OsclIPSocketI](#).

- 6.226.3.15 **int32** OsclTCPSocketI::Listen (*int aQueueSize*) [inline]
- 6.226.3.16 **TPVSocketEvent** OsclTCPSocketI::ListenAsync (*uint32 qsize, int32 aTimeoutMsec = -1*) [inline]
- 6.226.3.17 OsclTCPSocketI* OsclTCPSocketI::NewL (**Oscl_DefAlloc** & *a*, **OsclSocketServI** * *aServ*, **OsclSocketObserver** * *aObserver*, *uint32 aId*) [static]
- 6.226.3.18 **TPVSocketEvent** OsclTCPSocketI::Recv (*uint8 *& aPtr, uint32 aMaxLen, int32 aTimeoutMsec = -1*) [inline]
- 6.226.3.19 **TPVSocketEvent** OsclTCPSocketI::Send (*const uint8 *& aPtr, uint32 aLen, int32 aTimeoutMsec = -1*) [inline]
- 6.226.3.20 **TPVSocketEvent** OsclTCPSocketI::Shutdown (**TPVSocketShutdown** *aHow, int32 aTimeoutMsec = -1*) [inline]

The documentation for this class was generated from the following file:

- [oscl_tcp_socket.h](#)

6.227 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)
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError GetPriority (OsclThreadPriority &refThreadPriority)
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError SetPriority (OsclThreadPriority ePriority)
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Suspend ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Resume ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError Terminate (OsclAny *exitcode)

Static Public Methods

- OSCL_IMPORT_REF void Exit (OsclAny *exitcode)
- OSCL_IMPORT_REF void EnableKill ()
- OSCL_IMPORT_REF OsclProcStatus::eOsclProcError GetId (TOsclThreadId &refThreadId)
- OSCL_IMPORT_REF bool CompareId (TOsclThreadId &t1, TOsclThreadId &t2)
- OSCL_IMPORT_REF void SleepMillisec (const int32 msec)

6.227.1 Detailed Description

Thread Class. A subset of Thread APIs. It implements platform independent APIs for thread creation, exiting, suspend, resume, priority and termination. With the use of proper defines it implements the basic thread features. It provides an opaque layer through which user doesn't need to worry about OS specific data.

6.227.2 Constructor & Destructor Documentation

6.227.2.1 OSCL_IMPORT_REF OsclThread::OsclThread ()

Class constructor

6.227.2.2 OSCL_IMPORT_REF OsclThread::~OsclThread ()

Class destructor

6.227.3 Member Function Documentation

6.227.3.1 OSCL_IMPORT_REF bool OsclThread::CompareId (TOsclThreadId & t1, TOsclThreadId & t2) [static]

Static routine to compare whether two thread ID's are equal.

Parameters:

t1, t2: thread ID passed by the application

Returns:

true if equal.

**6.227.3.2 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::Create
(TOsclThreadFuncPtr *func*, int32 *stack_size*, TOsclThreadFuncArg *argument*,
OsclThread_State *state* = Start_on_creation)**

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.

Returns:

eOsclProcError

6.227.3.3 OSCL_IMPORT_REF void OsclThread::EnableKill () [static]

EnableKill is a static function which can be called by the thread routine in order to enable thread termination without waiting for cancellation points. EnableKill only applies to pthread implementations. For other implementations this function will do nothing.

Returns:

None

6.227.3.4 OSCL_IMPORT_REF void OsclThread::Exit (OsclAny * *exitcode*) [static]

Exit is a static function which is used to end the current thread. When called it just ends the execution of the current thread.

Parameters:

exitcode = Exitcode of the thread. This can be used by other threads to know the exit status of this thread.

Returns:

None

**6.227.3.5 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::GetId
(TOsclThreadId & *refThreadId*) [static]**

Static routine to retrieve ID of calling thread.

Parameters:

Thread ID passed by the application

Returns:

Error code

**6.227.3.6 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::GetPriority
(OsclThreadPriority & refThreadPriority)**

GetThreadPriority gets the priority of the thread. It takes reference of the input argument and assigns priority to it from one of the already defined priorities.

Parameters:

int16& refThreadPriority : Output Priority value

Returns:

Error code

6.227.3.7 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::Resume ()

ResumeThread resumes the suspended thread and brings it into execution.

Parameters:

None

Returns:

Error code Note: this function may not be supported on all platforms, and may return NOT_ - IMPLEMENTED.

**6.227.3.8 OSCL_IMPORT_REF OsclProcStatus::eOsclProcError OsclThread::SetPriority
(OsclThreadPriority ePriority)**

SetThreadPriority sets the priority of the thread. It takes priority as the input argument and assigns it to the thread referred.

Parameters:

ePriorityLevel : Input Priority value

Returns:

Error code Note: this function may not be supported on all platforms, and may return NOT_ - IMPLEMENTED.

6.227.3.9 OSCL_IMPORT_REF void OsclThread::SleepMillisec (const int32 msec) [static]

Suspend current thread execution for specified time.

Parameters:

msec, t2: sleep time in milliseconds.

6.227.3.10 OSCL_IMPORT_REF [OsclProcStatus::eOsclProcError](#) OsclThread::Suspend ()

This API suspends the thread being referred. The thread can later be brought into execution by calling OSCL_ResumeThread() on it.

Parameters:

None

Returns:

Error code Note: this function may not be supported on all platforms, and may return NOT_IMPLEMENTED.

6.227.3.11 OSCL_IMPORT_REF [OsclProcStatus::eOsclProcError](#) OsclThread::Terminate ([OsclAny](#) * *exitcode*)

Terminate a thread other than the calling thread.

Note: for pthread implementations, the Terminate call will block until the thread has terminated. By default, threads will not terminate until a cancellation point is reached. The EnableKill method may be used to override this default behavior and allow immediate termination.

Parameters:

exitcode = Exitcode of the thread.

Returns:

Error code

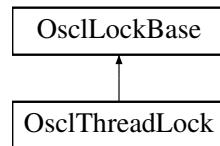
The documentation for this class was generated from the following file:

- [oscl_thread.h](#)

6.228 OsclThreadLock Class Reference

```
#include <oscl_mutex.h>
```

Inheritance diagram for OsclThreadLock::



Public Methods

- OSCL_IMPORT_REF OsclThreadLock ()
- virtual OSCL_IMPORT_REF ~OsclThreadLock ()
- OSCL_IMPORT_REF void Lock ()
- OSCL_IMPORT_REF void Unlock ()

6.228.1 Detailed Description

An implementation of [OsclLockBase](#) using a mutex

6.228.2 Constructor & Destructor Documentation

6.228.2.1 OSCL_IMPORT_REF OsclThreadLock::OsclThreadLock ()

6.228.2.2 virtual OSCL_IMPORT_REF OsclThreadLock::~OsclThreadLock () [virtual]

6.228.3 Member Function Documentation

6.228.3.1 OSCL_IMPORT_REF void OsclThreadLock::Lock () [virtual]

Implements [OsclLockBase](#).

6.228.3.2 OSCL_IMPORT_REF void OsclThreadLock::Unlock () [virtual]

Implements [OsclLockBase](#).

The documentation for this class was generated from the following file:

- [oscl_mutex.h](#)

6.229 OsclTickCount Class Reference

```
#include <oscl_tickcount.h>
```

Static Public Methods

- uint32 [TickCount \(\)](#)
- uint32 [TickCountFrequency \(\)](#)
- uint32 [TickCountPeriod \(\)](#)
- uint32 [TicksToMsec \(uint32 ticks\)](#)
- uint32 [MsecToTicks \(uint32 msec\)](#)

6.229.1 Detailed Description

OsclTickCount class is used to retrieve the system tick count and the tick counter's frequency. The maximum tick count value is equivalent to the maximum uint32 value.

6.229.2 Member Function Documentation

6.229.2.1 uint32 OsclTickCount::MsecToTicks (uint32 *msec*) [static]

This function converts milliseconds to ticks

Returns:

ticks

6.229.2.2 uint32 OsclTickCount::TickCount () [static]

This function returns the current system tick count

Returns:

returns the tick count

6.229.2.3 uint32 OsclTickCount::TickCountFrequency () [static]

This function returns the tick frequency in ticks per second

Returns:

ticks per second

6.229.2.4 uint32 OsclTickCount::TickCountPeriod () [static]

This function returns the tick period in microseconds per tick

Returns:

microseconds per tick

6.229.2.5 uint32 OsclTickCount::TicksToMsec (uint32 *ticks*) [static]

This function converts ticks to milliseconds

Returns:

milliseconds

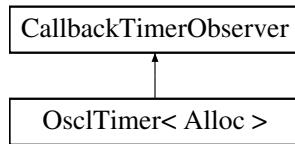
The documentation for this class was generated from the following file:

- [oscl_tickcount.h](#)

6.230 OsclTimer< Alloc > Class Template Reference

```
#include <oscl_timer.h>
```

Inheritance diagram for OsclTimer< Alloc >::



Public Types

- `typedef CallbackTimer< Alloc > callback_timer_type`

Public Methods

- `OsclTimer (const char *name, uint32 frequency=1, int32 priority=OsclActiveObject::EPriorityNominal)`
- `virtual ~OsclTimer ()`
- `void SetObserver (OsclTimerObserver *obs)`
- `void SetFrequency (uint32 frequency)`
- `void SetExactFrequency (uint32 frequency)`
- `void Request (int32 timerID, int32 timeoutInfo, int32 cycles, OsclTimerObserver *obs=0, bool recurring=0)`
- `void Cancel (int32 timerID, int32 timeoutInfo=-1)`
- `void Clear ()`

Protected Methods

- `void TimerBaseElapsed ()`

Friends

- `class CallbackTimer< Alloc >`

template<class Alloc> class OsclTimer< Alloc >

6.230.1 Member Typedef Documentation

6.230.1.1 template<class Alloc> typedef CallbackTimer<Alloc> OsclTimer< Alloc >::callback_timer_type

6.230.2 Constructor & Destructor Documentation

6.230.2.1 template<class Alloc> OsclTimer< Alloc >::OsclTimer (const char * *name*, uint32 *frequency* = 1, int32 *priority* = OsclActiveObject::EPriorityNominal)

Constructor

Parameters:

frequency The frequency of the timer in cycles/second. A value of 1 means the timer will cycle in 1 second intervals.

6.230.2.2 template<class Alloc> OsclTimer< Alloc >::~OsclTimer () [virtual]

6.230.3 Member Function Documentation

6.230.3.1 template<class Alloc> void OsclTimer< Alloc >::Cancel (int32 *timerID*, int32 *timeoutInfo* = -1)

Cancel a timer

Parameters:

timerID used to identify the timer to cancel.

timeoutInfo if not set to -1, this value will be used as additional matching criteria to cancel a timer.

6.230.3.2 template<class Alloc> void OsclTimer< Alloc >::Clear ()

Cancel all pending timers.

6.230.3.3 template<class Alloc> void OsclTimer< Alloc >::Request (int32 *timerID*, int32 *timeoutInfo*, int32 *cycles*, OsclTimerObserver * *obs* = 0, bool *recurring* = 0)

Request a timer

Parameters:

timerID used to identify the timer for cancellation. This value will be returned as part of the timeout event.

timeoutInfo for user info. Returned to the observer on a timeout event

cycles the number of cycles to wait before a timeout event. If the timer frequency is 1 and the cycles are set to 2, then the timeout event will occur in 2 seconds.

obs a local observer object to be called on a timeout event. This observer overrides the global observer if set.

6.230.3.4 template<class Alloc> void OsclTimer< Alloc >::SetExactFrequency (uint32 *frequency*)

Set the exact frequency of the timer in microsecond.

Parameters:

frequency A value of 1 means the timer will cycle in one microsecond intervals, 1000 means millisecond intervals, etc.

6.230.3.5 template<class Alloc> void OsclTimer< Alloc >::SetFrequency (uint32 *frequency*)

Set the frequency of the timer in cycles/second.

Parameters:

frequency A value of 1 means the timer will cycle in one second intervals, 1000 means millisecond intervals, etc.

6.230.3.6 template<class Alloc> void OsclTimer< Alloc >::SetObserver ([OsclTimerObserver](#) * *obs*) [inline]

Set the global observer. Each timer can request a local observer, which if set overrides the global observer.

Parameters:

obs observer object.

6.230.3.7 template<class Alloc> void OsclTimer< Alloc >::TimerBaseElapsed () [protected, virtual]

Implements [CallbackTimerObserver](#).

6.230.4 Friends And Related Function Documentation

6.230.4.1 template<class Alloc> friend class [CallbackTimer](#)< Alloc > [friend]

The documentation for this class was generated from the following file:

- [oscl_timer.h](#)

6.231 OsclTimerCompare Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Static Public Methods

- int [compare \(TOsclReady &a, TOsclReady &b\)](#)

6.231.1 Member Function Documentation

6.231.1.1 int OsclTimerCompare::compare (TOsclReady &*a*, TOsclReady &*b*) [static]

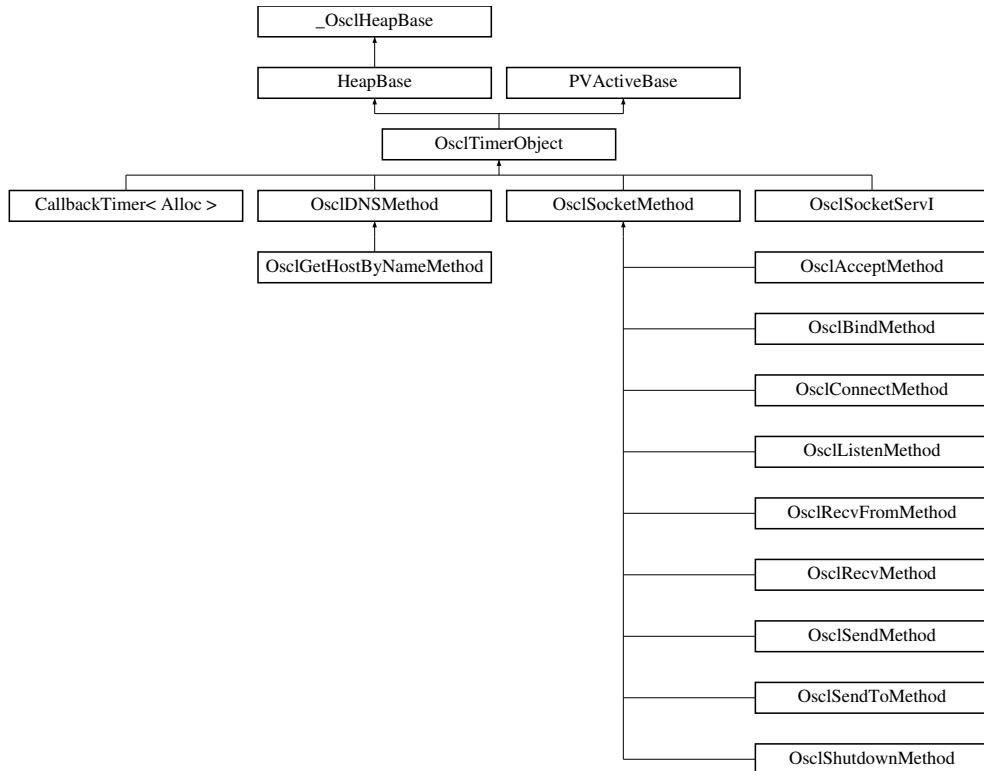
The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

6.232 OsclTimerObject Class Reference

```
#include <oscl_scheduler_ao.h>
```

Inheritance diagram for OsclTimerObject::



Public Methods

- OSCL_IMPORT_REF [OsclTimerObject](#) (int32 aPriority, const char name[])
- virtual OSCL_IMPORT_REF [~OsclTimerObject](#) ()
- OSCL_IMPORT_REF void [AddToScheduler](#) ()
- OSCL_IMPORT_REF void [RemoveFromScheduler](#) ()
- OSCL_IMPORT_REF void [After](#) (int32 aDelayMicrosec)
- OSCL_IMPORT_REF void [RunIfNotReady](#) (uint32 aDelayMicrosec=0)
- OSCL_IMPORT_REF void [SetBusy](#) ()
- OSCL_IMPORT_REF bool [IsBusy](#) () const
- OSCL_IMPORT_REF void [Cancel](#) ()
- OSCL_IMPORT_REF int32 [Priority](#) () const
- OSCL_IMPORT_REF int32 [Status](#) () const
- OSCL_IMPORT_REF void [SetStatus](#) (int32)
- OSCL_IMPORT_REF [OsclAOStatus](#) & [StatusRef](#) ()

Protected Methods

- virtual OSCL_IMPORT_REF void [DoCancel](#) ()
- virtual OSCL_IMPORT_REF int32 [RunError](#) (int32 aError)

6.232.1 Detailed Description

User base class for execution objects. OsclTimerObject defines an exec object with a timer.

6.232.2 Constructor & Destructor Documentation

6.232.2.1 OSCL_IMPORT_REF OsclTimerObject::OsclTimerObject (int32 *aPriority*, const char *name*[])

Constructor.

Parameters:

aPriority (input param): scheduling priority

name (input param): optional name for this AO.

6.232.2.2 virtual OSCL_IMPORT_REF OsclTimerObject::~OsclTimerObject () [virtual]

Destructor.

6.232.3 Member Function Documentation

6.232.3.1 OSCL_IMPORT_REF void OsclTimerObject::AddToScheduler ()

Add this AO to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

6.232.3.2 OSCL_IMPORT_REF void OsclTimerObject::After (int32 *aDelayMicrosec*)

'After' sets the request ready, with request status OSCL_REQUEST_STATUS_PENDING, and starts a timer. When the timer expires, the request will complete with status OSCL_REQUEST_ERR_NONE. Must be called from the same thread in which the active object is scheduled. Will leave if the request is already readied, the object is not added to any scheduler, or the calling thread does not match the scheduling thread.

Parameters:

anInterval: timeout interval in microseconds.

6.232.3.3 OSCL_IMPORT_REF void OsclTimerObject::Cancel ()

Cancel any active request. If the request is pending, this will call the DoCancel routine, wait for the request to cancel, then set the request idle. The AO will not run. If the request is not active, it does nothing. Request must be canceled from the same thread in which it is scheduled.

Reimplemented from [PVActiveBase](#).

6.232.3.4 virtual OSCL_IMPORT_REF void OsclTimerObject::DoCancel () [protected, virtual]

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will cancel the timer. If any additional action is needed, the derived class may override this. If the derived class does override this, it should explicitly call [OsclTimerObject::DoCancel](#) in its own DoCancel routine.

Implements [PVActiveBase](#).

6.232.3.5 OSCL_IMPORT_REF bool OsclTimerObject::IsBusy ()

Return true if this AO is active, false otherwise.

6.232.3.6 OSCL_IMPORT_REF int32 OsclTimerObject::Priority ()

Return scheduling priority of this exec object.

6.232.3.7 OSCL_IMPORT_REF void OsclTimerObject::RemoveFromScheduler ()

Remove this AO from its scheduler. Will leave if the calling thread context does not match the scheduling thread. Cancels any pending request before removing.

Reimplemented from [PVActiveBase](#).

6.232.3.8 virtual OSCL_IMPORT_REF int32 OsclTimerObject::RunError (int32 *aError*) [protected, virtual]

Run Leave handler. This gets called by scheduler when the Run routine leaves. The default implementation simply returns the leave code. If the derived class wants to handle errors from Run, it may override this. The ExecError should return OsclErrNone if it handles the error, otherwise it should return the input error code.

Parameters:

aError: the leave code generated by the Run.

Implements [PVActiveBase](#).

6.232.3.9 OSCL_IMPORT_REF void OsclTimerObject::RunIfNotReady (uint32 *aDelayMicrosec* = 0)

Complete the request after a time interval. RunIfNotReady is identical to [After\(\)](#) except that it first checks the request status, and if it is already readied, it does nothing.

Parameters:

aDelayMicrosec (input param): delay in microseconds.

6.232.3.10 OSCL_IMPORT_REF void OsclTimerObject::SetBusy ()

Set request ready for this AO. Will leave if the request is already readied, or the exec object is not added to any scheduler, or the calling thread context does not match the scheduler thread.

6.232.3.11 OSCL_IMPORT_REF void OsclTimerObject::SetStatus (int32)

6.232.3.12 OSCL_IMPORT_REF int32 OsclTimerObject::Status ()

Request status access

6.232.3.13 OSCL_IMPORT_REF OsclAOStatus& OsclTimerObject::StatusRef ()

The documentation for this class was generated from the following file:

- [oscl_scheduler_ao.h](#)

6.233 OsclTimerObserver Class Reference

```
#include <oscl_timer.h>
```

Public Methods

- virtual void [TimeoutOccurred](#) (int32 timerID, int32 timeoutInfo)=0
- virtual [~OsclTimerObserver](#) ()

6.233.1 Detailed Description

The observer class to receive timeout callbacks

6.233.2 Constructor & Destructor Documentation

6.233.2.1 **virtual OsclTimerObserver::~OsclTimerObserver ()** [inline, virtual]

6.233.3 Member Function Documentation

6.233.3.1 **virtual void OsclTimerObserver::TimeoutOccurred (int32 *timerID*, int32 *timeoutInfo*)** [pure virtual]

This function will be called when the timer associated with this observer is executed

Parameters:

timerID The ID given at timer request.

timeoutInfo Any extra info given at timer request.

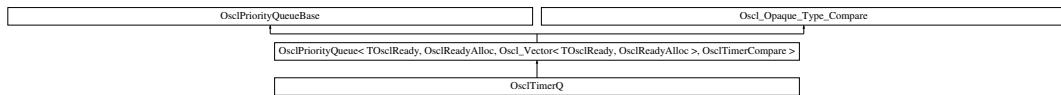
The documentation for this class was generated from the following file:

- [oscl_timer.h](#)

6.234 OsclTimerQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclTimerQ::



Public Methods

- void [Construct](#) (int)
- void [Add](#) ([TOsclReady](#))
- void [Remove](#) ([TOsclReady](#))
- [TOsclReady PopTop](#) ()
- [TOsclReady Top](#) ()
- void [Pop](#) ([TOsclReady](#))
- bool [IsIn](#) ([TOsclReady](#))

6.234.1 Member Function Documentation

6.234.1.1 void OsclTimerQ::Add ([TOsclReady](#))

6.234.1.2 void OsclTimerQ::Construct (int)

6.234.1.3 bool OsclTimerQ::IsIn ([TOsclReady](#))

6.234.1.4 void OsclTimerQ::Pop ([TOsclReady](#))

6.234.1.5 [TOsclReady](#) OsclTimerQ::PopTop ()

6.234.1.6 void OsclTimerQ::Remove ([TOsclReady](#))

6.234.1.7 [TOsclReady](#) OsclTimerQ::Top ()

The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

6.235 OsclTLS< T, ID, Registry > Class Template Reference

```
#include <oscl_tls.h>
```

Public Methods

- `OsclTLS()`
- `~OsclTLS()`
- `T & operator *() const`
The indirection operator () accesses a value indirectly, through a pointer.*
- `T * operator ->() const`
The indirection operator (->) accesses a value indirectly, through a pointer.
- `bool set()`
set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

Protected Attributes

- `T * _Ptr`

`template<class T, uint32 ID, class Registry = OsclTLSRegistry> class OsclTLS< T, ID, Registry >`

6.235.1 Constructor & Destructor Documentation

6.235.1.1 `template<class T, uint32 ID, class Registry = OsclTLSRegistry> OsclTLS< T, ID, Registry >::OsclTLS () [inline]`

6.235.1.2 `template<class T, uint32 ID, class Registry = OsclTLSRegistry> OsclTLS< T, ID, Registry >::~OsclTLS () [inline]`

6.235.2 Member Function Documentation

6.235.2.1 `template<class T, uint32 ID, class Registry = OsclTLSRegistry> T& OsclTLS< T, ID, Registry >::operator *() const [inline]`

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the OsclTLS can be used like the regular pointer that it was initialized with.

6.235.2.2 `template<class T, uint32 ID, class Registry = OsclTLSRegistry> T* OsclTLS< T, ID, Registry >::operator ->() const [inline]`

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the OsclTLS can be used like the regular pointer that it was initialized with.

6.235.2.3 template<class T, uint32 ID, class Registry = OsclTLSRegistry> bool OsclTLS< T, ID, Registry >::set () [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

6.235.3 Field Documentation**6.235.3.1 template<class T, uint32 ID, class Registry = OsclTLSRegistry> T* OsclTLS< T, ID, Registry >::_Ptr [protected]**

The documentation for this class was generated from the following file:

- [oscl_tls.h](#)

6.236 OsclTLSEEx< T, ID, Registry > Class Template Reference

```
#include <oscl_error.h>
```

Public Methods

- `OsclTLSEEx ()`
- `~OsclTLSEEx ()`
- `T & operator * () const`

The indirection operator () accesses a value indirectly, through a pointer.*

- `T * operator -> () const`

The indirection operator (->) accesses a value indirectly, through a pointer.

- `bool set ()`

set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

Protected Attributes

- `T * _Ptr`

```
template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> class OsclTLSEEx< T, ID, Registry >
```

6.236.1 Constructor & Destructor Documentation

6.236.1.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> OsclTLSEEx< T, ID, Registry >::OsclTLSEEx () [inline]

6.236.1.2 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> OsclTLSEEx< T, ID, Registry >::~OsclTLSEEx () [inline]

6.236.2 Member Function Documentation

6.236.2.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T& OsclTLSEEx< T, ID, Registry >::operator * () const [inline]

The indirection operator (*) accesses a value indirectly, through a pointer.

This operator ensures that the `OsclTLS` can be used like the regular pointer that it was initialized with.

6.236.2.2 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T* OsclTLSEEx< T, ID, Registry >::operator -> () const [inline]

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the `OsclTLS` can be used like the regular pointer that it was initialized with.

6.236.2.3 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> bool OsclTLSEEx< T, ID, Registry >::set () [inline]

[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.

6.236.3 Field Documentation

6.236.3.1 template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> T* OsclTLSEEx< T, ID, Registry >::_Ptr [protected]

The documentation for this class was generated from the following file:

- [oscl_error.h](#)

6.237 OsclTLSRegistry Class Reference

```
#include <oscl_tls.h>
```

Static Public Methods

- OSCL_IMPORT_REF [OsclAny](#) * getInstance (uint32 ID, int32 &error)
- OSCL_IMPORT_REF void [registerInstance](#) ([OsclAny](#) *ptr, uint32 ID, int32 &error)

Friends

- class [OsclBase](#)

6.237.1 Member Function Documentation

6.237.1.1 OSCL_IMPORT_REF [OsclAny](#)* OsclTLSRegistry::getInstance (uint32 *ID*, int32 & *error*) [static]

6.237.1.2 OSCL_IMPORT_REF void OsclTLSRegistry::registerInstance ([OsclAny](#) * *ptr*, uint32 *ID*, int32 & *error*) [static]

6.237.2 Friends And Related Function Documentation

6.237.2.1 friend class [OsclBase](#) [friend]

The documentation for this class was generated from the following file:

- [oscl_tls.h](#)

6.238 OsclTLSRegistryEx Class Reference

```
#include <oscl_error.h>
```

Static Public Methods

- [OsclAny * getInstance \(uint32 ID\)](#)
- [void registerInstance \(OsclAny *ptr, uint32 ID\)](#)

6.238.1 Member Function Documentation

6.238.1.1 [OsclAny* OsclTLSRegistryEx::getInstance \(uint32 ID\)](#) [inline, static]

6.238.1.2 [void OsclTLSRegistryEx::registerInstance \(OsclAny *ptr, uint32 ID\)](#) [inline, static]

The documentation for this class was generated from the following file:

- [oscl_error.h](#)

6.239 OsclTrapItem Class Reference

```
#include <oscl_heapbase.h>
```

Public Methods

- OSCL_INLINE [OsclTrapItem \(OsclTrapOperation anOperation\)](#)
- OSCL_INLINE [OsclTrapItem \(OsclTrapOperation anOperation, OsclAny *aPtr\)](#)

Friends

- class [OsclTrapStackItem](#)
- class [OsclTrapStack](#)

6.239.1 Constructor & Destructor Documentation

6.239.1.1 OSCL_INLINE OsclTrapItem::OsclTrapItem ([OsclTrapOperation anOperation](#))

6.239.1.2 OSCL_INLINE OsclTrapItem::OsclTrapItem ([OsclTrapOperation anOperation, OsclAny * aPtr](#))

6.239.2 Friends And Related Function Documentation

6.239.2.1 friend class OsclTrapStack [friend]

6.239.2.2 friend class OsclTrapStackItem [friend]

The documentation for this class was generated from the following file:

- [oscl_heapbase.h](#)

6.240 OsclTrapStack Class Reference

```
#include <oscl_error_trapcleanup.h>
```

Friends

- class [OsclError](#)
- class [OsclErrorTrap](#)
- class [OsclErrorTrapImp](#)

6.240.1 Detailed Description

A common type for cleanup stack and trap mark stack. for internal use only.

6.240.2 Friends And Related Function Documentation

6.240.2.1 friend class OsclError [friend]

6.240.2.2 friend class OsclErrorTrap [friend]

6.240.2.3 friend class OsclErrorTrapImp [friend]

The documentation for this class was generated from the following file:

- [oscl_error_trapcleanup.h](#)

6.241 OsclTrapStackItem Class Reference

```
#include <oscl_error_trapcleanup.h>
```

Public Methods

- [OsclTrapStackItem \(\)](#)
- [OsclTrapStackItem \(_OsclHeapBase *aCBase\)](#)
- [OsclTrapStackItem \(OsclAny *aTAny\)](#)
- [OsclTrapStackItem \(OsclTrapItem aItem\)](#)

Data Fields

- [_OsclHeapBase * iCBase](#)
- [OsclAny * iTAny](#)
- [OsclTrapOperation iTrapOperation](#)
- [OsclTrapStackItem * iNext](#)

6.241.1 Detailed Description

Internal cleanup stack item type.

6.241.2 Constructor & Destructor Documentation

6.241.2.1 OsclTrapStackItem::OsclTrapStackItem () [inline]

6.241.2.2 OsclTrapStackItem::OsclTrapStackItem (_OsclHeapBase * aCBase) [inline]

6.241.2.3 OsclTrapStackItem::OsclTrapStackItem (OsclAny * aTAny) [inline]

6.241.2.4 OsclTrapStackItem::OsclTrapStackItem (OsclTrapItem aItem) [inline]

6.241.3 Field Documentation

6.241.3.1 _OsclHeapBase* OsclTrapStackItem::iCBase

6.241.3.2 OsclTrapStackItem* OsclTrapStackItem::iNext

6.241.3.3 OsclAny* OsclTrapStackItem::iTAny

6.241.3.4 OsclTrapOperation OsclTrapStackItem::iTrapOperation

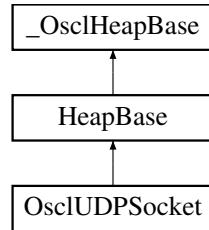
The documentation for this class was generated from the following file:

- [oscl_error_trapcleanup.h](#)

6.242 OsclUDPSocket Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclUDPSocket::



Public Methods

- OSCL_IMPORT_REF ~OsclUDPSocket ()
- OSCL_IMPORT_REF int32 Close ()
- OSCL_IMPORT_REF int32 Bind (OsclNetworkAddress &aAddress)
- OSCL_IMPORT_REF int32 Join (OsclNetworkAddress &aAddress)
- OSCL_IMPORT_REF TPVSocketEvent BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL_IMPORT_REF void CancelBind ()
- OSCL_IMPORT_REF uint8 * GetRecvData (int32 *aLength)
- OSCL_IMPORT_REF uint8 * GetSendData (int32 *aLength)
- OSCL_IMPORT_REF TPVSocketEvent SendTo (const uint8 *aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL_IMPORT_REF void CancelSendTo ()
- OSCL_IMPORT_REF TPVSocketEvent RecvFrom (uint8 *aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiRecvLimit=0, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen=NULL, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource=NULL)
- OSCL_IMPORT_REF void CancelRecvFrom ()
- OSCL_IMPORT_REF int32 SetRecvBufferSize (uint32 size)

Static Public Methods

- OSCL_IMPORT_REF OsclUDPSocket * NewL (Oscl_DefAlloc &alloc, OsclSocketServ &aServ, OsclSocketObserver *aObserver, uint32 aId)

6.242.1 Detailed Description

The UDP Socket class

6.242.2 Constructor & Destructor Documentation

6.242.2.1 OSCL_IMPORT_REF OsclUDPSocket::~OsclUDPSocket ()

Destructor. The object must be deleted using the same allocator used in the NewL call.

6.242.3 Member Function Documentation

6.242.3.1 OSCL_IMPORT_REF int32 OsclUDPSocket::Bind ([OsclNetworkAddress](#) & *aAddress*)

Bind a UDP socket to an address. This is a synchronous method.

Parameters:

aAddress: Bind address.

Returns:

Returns OsclErrNone for success, or a platform-specific error code.

6.242.3.2 OSCL_IMPORT_REF [TPVSocketEvent](#) OsclUDPSocket::BindAsync ([OsclNetworkAddress](#) & *aAddress*, int32 *aTimeoutMsec* = (-1))

Bind a UDP socket to an address. This is an asynchronous method.

Parameters:

aAddress: Bind address.

aTimeoutMsec: Optional timeout. Use a negative value for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

6.242.3.3 OSCL_IMPORT_REF void OsclUDPSocket::CancelBind ()

Cancel Bind

This method will cancel any pending BindAsync operation on the current socket, causing the BindAsync to complete with error EPVSocketCancel. If there is no pending BindAsync operation, this method will have no effect.

6.242.3.4 OSCL_IMPORT_REF void OsclUDPSocket::CancelRecvFrom ()

Cancel RecvFrom

This method will cancel any pending RecvFrom operation on the current socket, causing the RecvFrom to complete with error EPVSocketCancel. If there is no pending RecvFrom operation, this method will have no effect.

6.242.3.5 OSCL_IMPORT_REF void OsclUDPSocket::CancelSendTo ()

Cancel SendTo

This method will cancel any pending SendTo operation on the current socket, causing the SendTo to complete with error EPVSocketCancel. If there is no pending SendTo operation, this method will have no effect.

6.242.3.6 OSCL_IMPORT_REF int32 OsclUDPSocket::Close ()

Close a UDP socket. This is a synchronous method.

Once it is closed a socket cannot be re-opened. Sockets are automatically closed when they are deleted. This method may be used to see any error code returned from the platform's socket close call.

Returns:

Returns OsclErrNone for success, or a platform-specific error code.

6.242.3.7 OSCL_IMPORT_REF uint8* OsclUDPSocket::GetRecvData (int32 * aLength)

Retrieve the received data after a successful RecvFrom operation. This is a synchronous method.

Parameters:

aLength: (output) number of bytes of data received.

Returns:

Returns pointer to received data, or NULL if none.

6.242.3.8 OSCL_IMPORT_REF uint8* OsclUDPSocket::GetSendData (int32 * aLength)

Retrieve the sent data after a successful SendTo operation. This is a synchronous method.

Parameters:

aLength: (output) number of bytes of data sent.

Returns:

Returns pointer to sent data, or NULL if none.

6.242.3.9 OSCL_IMPORT_REF int32 OsclUDPSocket::Join ([OsclNetworkAddress](#) & aAddress)

Bind a UDP socket to an address and Join the multicast group. This is a synchronous method.

Parameters:

aAddress: Bind address.

Returns:

Returns OsclErrNone for success, or a platform-specific error code. May throw an OsclErrNotSupported Exception

6.242.3.10 OSCL_IMPORT_REF OsclUDPSocket* OsclUDPSocket::NewL ([Oscl_DefAlloc](#) & alloc, [OsclSocketServ](#) & aServ, [OsclSocketObserver](#) * aObserver, uint32 aId) [static]

Create a UDP Socket. May leave if failure.

Parameters:

alloc: Memory allocator.

aServ: Socket server. Must be connected.

aObserver: Socket observer.

aId: Socket ID. The caller must assign an ID to each socket. The ID is used to identify the socket in observer callbacks.

Returns:

Returns pointer to socket.

6.242.3.11 OSCL_IMPORT_REF TPVSocketEvent OsclUDPSocket::RecvFrom (uint8 * aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiRecvLimit = 0, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen = NULL, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource = NULL)

Receive Data. This is an asynchronous method.

Parameters:

aPtr: Buffer to receive incoming data

aMaxLen: Length of buffer.

aAddress: (output) Source address.

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

aMultiRecvLimit (optional input): Configures multiple packet receive mode. As long as there are packets queued at the socket and at least aMultiRecvLimit bytes are available in the buffer, recvfrom operations will continue. A value of zero disabled multiple packet mode. The individual packet lengths can be retrieved in the aPacketLen parameter; and the individual packet source addresses can be retrieved in the aPacketSource parameter.

aPacketLen: (optional output) a vector of packet lengths, in case multiple packets were received.

aPacketSource: (optional output) a vector of source addresses, in case multiple packets were received.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

6.242.3.12 OSCL_IMPORT_REF TPVSocketEvent OsclUDPSocket::SendTo (const uint8 * aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1)

Send Data. This is an asynchronous method.

Parameters:

aPtr: Data to send.

aLen: Length of data to send.

aAddress: Destination address.

aTimeoutMsec: Timeout in milliseconds, or (-1) for infinite wait.

Returns:

Will return EPVSocketPending if successful. When the operation is complete, a callback to the observer will occur with the completion status. If the operation cannot be initiated, the call will return EPVSocketFailure and there will be no callback.

6.242.3.13 OSCL_IMPORT_REF int32 OsclUDPSocket::SetRecvBufferSize (uint32 *size*)

Set the buffer size of the socket This is a synchronous method.

Parameters:

size: buffer size

Returns:

Returns OsclErrNone for success, or a platform-specific error code. May throw an OsclErrNotSupported Exception.

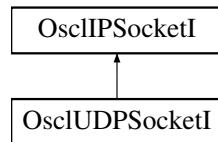
The documentation for this class was generated from the following file:

- [oscl_socket.h](#)

6.243 OsclUDPSocketI Class Reference

```
#include <oscl_udp_socket.h>
```

Inheritance diagram for OsclUDPSocketI::



Public Methods

- virtual ~OsclUDPSocketI ()
- int32 [Close \(\)](#)
- uint8 * [GetRecvData \(int32 *aLength\)](#)
- uint8 * [GetSendData \(int32 *aLength\)](#)
- [TPVSocketEvent BindAsync \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelBind \(\)](#)
- [TPVSocketEvent SendTo \(const uint8 *&aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelSendTo \(\)](#)
- [TPVSocketEvent RecvFrom \(uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiMaxLen=0, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen=NULL, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource=NULL\)](#)
- void [CancelRecvFrom \(\)](#)

Static Public Methods

- OsclUDPSocketI * [NewL \(Oscl_DefAlloc &a, OsclSocketServI *aServ, OsclSocketObserver *aObserver, uint32 aId\)](#)

6.243.1 Detailed Description

Internal implementation class for [OsclUDPSocket](#)

6.243.2 Constructor & Destructor Documentation

6.243.2.1 **virtual OsclUDPSocketI::~OsclUDPSocketI () [virtual]**

6.243.3 Member Function Documentation

6.243.3.1 **TPVSocketEvent OsclUDPSocketI::BindAsync (OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]**

6.243.3.2 **void OsclUDPSocketI::CancelBind () [inline]**

6.243.3.3 **void OsclUDPSocketI::CancelRecvFrom () [inline]**

6.243.3.4 **void OsclUDPSocketI::CancelSendTo () [inline]**

6.243.3.5 **int32 OsclUDPSocketI::Close () [virtual]**

Implements [OsclIPSocketI](#).

6.243.3.6 **uint8 * OsclUDPSocketI::GetRecvData (int32 * aLength) [inline, virtual]**

Implements [OsclIPSocketI](#).

6.243.3.7 **uint8 * OsclUDPSocketI::GetSendData (int32 * aLength) [inline, virtual]**

Implements [OsclIPSocketI](#).

6.243.3.8 **OsclUDPSocketI* OsclUDPSocketI::NewL (Oscl_DefAlloc & a, OsclSocketServI * aServ, OsclSocketObserver * aObserver, uint32 aId) [static]**

6.243.3.9 **TPVSocketEvent OsclUDPSocketI::RecvFrom (uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiMaxLen = 0, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen = NULL, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource = NULL) [inline]**

6.243.3.10 **TPVSocketEvent OsclUDPSocketI::SendTo (const uint8 *& aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]**

The documentation for this class was generated from the following file:

- [oscl_udp_socket.h](#)

6.244 OsclUuid Struct Reference

```
#include <oscl_uuid.h>
```

Public Methods

- [OsclUuid \(\)](#)
- [OsclUuid \(uint32 l, uint16 w1, uint16 w2, uint8 b1, uint8 b2, uint8 b3, uint8 b4, uint8 b5, uint8 b6, uint8 b7, uint8 b8\)](#)
- [OsclUuid \(const char *aUuidString\)](#)
- [OsclUuid \(const OsclUuid &uuid\)](#)
- [OsclUuid & operator= \(const OsclUuid &src\)](#)
- [bool operator== \(const OsclUuid &src\) const](#)
- [bool operator!= \(const OsclUuid &src\) const](#)

Data Fields

- uint32 [data1](#)
- uint16 [data2](#)
- uint16 [data3](#)
- uint8 [data4](#) [BYTES_IN_UUID_ARRAY]

6.244.1 Detailed Description

OSCL UUID structure used for unique identification of modules and interfaces.

6.244.2 Constructor & Destructor Documentation

6.244.2.1 OsclUuid::OsclUuid () [inline]

6.244.2.2 OsclUuid::OsclUuid (uint32 *l*, uint16 *w1*, uint16 *w2*, uint8 *b1*, uint8 *b2*, uint8 *b3*, uint8 *b4*, uint8 *b5*, uint8 *b6*, uint8 *b7*, uint8 *b8*) [inline]

6.244.2.3 OsclUuid::OsclUuid (const char * *aUuidString*) [inline]

6.244.2.4 OsclUuid::OsclUuid (const OsclUuid & *uuid*) [inline]

6.244.3 Member Function Documentation

6.244.3.1 bool OsclUuid::operator!= (const OsclUuid & *src*) const [inline]

6.244.3.2 OsclUuid& OsclUuid::operator= (const OsclUuid & *src*) [inline]

6.244.3.3 bool OsclUuid::operator== (const OsclUuid & *src*) const [inline]

6.244.4 Field Documentation

6.244.4.1 uint32 OsclUuid::data1

6.244.4.2 uint16 OsclUuid::data2

6.244.4.3 uint16 OsclUuid::data3

6.244.4.4 uint8 OsclUuid::data4[BYTES_IN_UUID_ARRAY]

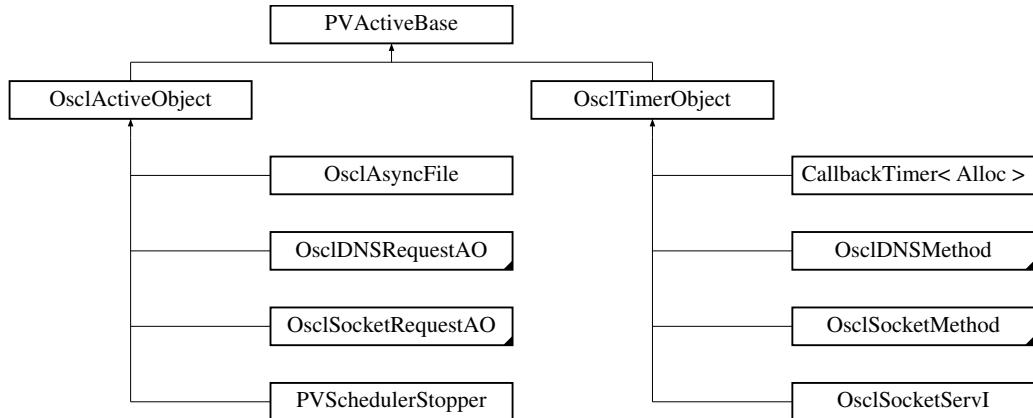
The documentation for this struct was generated from the following file:

- [oscl_uuid.h](#)

6.245 PVActiveBase Class Reference

```
#include <oscl_scheduler_aobase.h>
```

Inheritance diagram for PVActiveBase::



Public Methods

- [PVActiveBase](#) (const char name[], int32 pri)
- virtual [~PVActiveBase](#) ()
- bool [IsInAnyQ](#) ()
- virtual int32 [RunError](#) (int32 aError)=0
- virtual void [Run](#) ()=0
- virtual void [DoCancel](#) ()=0
- void [AddToScheduler](#) ()
- void [RemoveFromScheduler](#) ()
- void [Destroy](#) ()
- void [Activate](#) ()
- OSCL_IMPORT_REF bool [IsAdded](#) () const
- void [Cancel](#) ()

Data Fields

- uint32 [iAddedNum](#)
- [OsclNameString< PVEXECNAMELEN > iName](#)
- [PVThreadContext iThreadContext](#)
- [PVActiveStats * iPVAstats](#)
- [TReadyQueLink iPVReadyQLink](#)
- bool [iBusy](#)
- [OsclAOStatus iStatus](#)

Friends

- class [PVActiveStats](#)
- class [OsclSchedulerCommonBase](#)

- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [OsclReadyQ](#)
- class [OsclReadyCompare](#)
- class [OsclReadySetPosition](#)
- class [OsclExecScheduler](#)

6.245.1 Detailed Description

PV Scheduler internal AO base class. Both [OsclActiveObject](#) and [OsclTimerObject](#) derive from this class. For Symbian, this just container has the desired additions to the basic CTimer or OsclActiveObj functionality. For non-Symbian, this class contains the entire AO implementation.

6.245.2 Constructor & Destructor Documentation

6.245.2.1 PVActiveBase::PVActiveBase (const char *name*[], int32 *pri*)

6.245.2.2 virtual PVActiveBase::~PVActiveBase () [virtual]

6.245.3 Member Function Documentation

6.245.3.1 void PVActiveBase::Activate ()

6.245.3.2 void PVActiveBase::AddToScheduler ()

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

6.245.3.3 void PVActiveBase::Cancel ()

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

6.245.3.4 void PVActiveBase::Destroy ()

6.245.3.5 virtual void PVActiveBase::DoCancel () [pure virtual]

Implements cancellation of an outstanding request.

This function is called as part of the active object's [Cancel\(\)](#).

It must call the appropriate cancel function offered by the active object's asynchronous service provider. The asynchronous service provider's cancel is expected to act immediately.

[DoCancel\(\)](#) must not wait for event completion; this is handled by [Cancel\(\)](#).

Implemented in [OsclDNSRequestAO](#), [OsclSocketRequestAO](#), [OsclActiveObject](#), and [OsclTimerObject](#).

6.245.3.6 OSCL_IMPORT_REF bool PVActiveBase::IsAdded ()

6.245.3.7 bool PVActiveBase::IsInAnyQ () [inline]

6.245.3.8 void PVActiveBase::RemoveFromScheduler ()

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

6.245.3.9 virtual void PVActiveBase::Run () [pure virtual]

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implemented in [OsclDNSMethod](#), [OsclDNSRequestAO](#), [OsclSocketMethod](#), [OsclSocketRequestAO](#), and [CallbackTimer< Alloc >](#).

6.245.3.10 virtual int32 PVActiveBase::RunError (int32 *aError*) [pure virtual]

Virtual routine that gets called if the active object's Run routine leaves.

Parameters:

***aError*:** the leave code generated by the Run.

Returns:

:returns `OsclErrNone` if the error was handled, or returns the input *aError* value if not handled.

Implemented in [OsclActiveObject](#), and [OsclTimerObject](#).

6.245.4 Friends And Related Function Documentation

6.245.4.1 friend class OsclActiveObject [friend]

6.245.4.2 friend class OsclExecScheduler [friend]

6.245.4.3 friend class OsclReadyCompare [friend]

6.245.4.4 friend class OsclReadyQ [friend]

6.245.4.5 friend class OsclReadySetPosition [friend]

6.245.4.6 friend class OsclSchedulerCommonBase [friend]

6.245.4.7 friend class OsclTimerObject [friend]

6.245.4.8 friend class PVActiveStats [friend]

6.245.5 Field Documentation

6.245.5.1 uint32 PVActiveBase::iAddedNum

6.245.5.2 bool PVActiveBase::iBusy

6.245.5.3 OsclNameString<PVEXECNAMELEN> PVActiveBase::iName

6.245.5.4 PVActiveStats* PVActiveBase::iPVActiveStats

6.245.5.5 TReadyQueLink PVActiveBase::iPVReadyQLink

6.245.5.6 OsclAOStatus PVActiveBase::iStatus

The request status associated with an asynchronous request.

This is passed as a parameter to all asynchronous service providers.

The active scheduler uses this to check whether the active object's request has completed.

The function can use the completion code to judge the success or otherwise of the request.

Request status contains one of the values OSCL_REQUEST_ERR_NONE: request completed with no error, or request is not active. OSCL_REQUEST_PENDING: request is active & pending OSCL_REQUEST_ERR_CANCEL: request was canceled before completion. or any user-defined value.

6.245.5.7 PVThreadContext PVActiveBase::iThreadContext

The documentation for this class was generated from the following file:

- [oscl_scheduler_aobase.h](#)

6.246 PVActiveStats Class Reference

```
#include <oscl_scheduler_aobase.h>
```

Friends

- class [PVActiveBase](#)
- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [OsclReadyQ](#)

6.246.1 Detailed Description

PV AO statistics

6.246.2 Friends And Related Function Documentation

6.246.2.1 friend class OsclActiveObject [friend]

6.246.2.2 friend class OsclExecScheduler [friend]

6.246.2.3 friend class OsclExecSchedulerCommonBase [friend]

6.246.2.4 friend class OsclReadyQ [friend]

6.246.2.5 friend class OsclTimerObject [friend]

6.246.2.6 friend class PVActiveBase [friend]

The documentation for this class was generated from the following file:

- [oscl_scheduler_aobase.h](#)

6.247 PVLogger Class Reference

```
#include <pvlogger.h>
```

Public Types

- `typedef int32 log_level_type`
- `typedef int32 message_id_type`
- `typedef int32 filter_status_type`
- `typedef _OsclBasicAllocator alloc_type`

Public Methods

- `void SetLogLevel (log_level_type level)`
- `OSCL_IMPORT_REF void SetLogLevelAndPropagate (log_level_type level)`
- `log_level_type GetLogLevel ()`
- `void DisableAppenderInheritance ()`
- `void AddAppender (OsclSharedPtr< PVLoggerAppender > &appender)`
- `void RemoveAppender (OsclSharedPtr< PVLoggerAppender > &appender)`
- `void AddFilter (OsclSharedPtr< PVLoggerFilter > &filter)`
- `uint32 GetNumAppenders ()`
- `OSCL_IMPORT_REF bool IsActive (log_level_type level)`
- `OSCL_IMPORT_REF void LogMsgStringV (message_id_type msgID, const char *fmt, va_list arguments)`
- `OSCL_IMPORT_REF void LogMsgBuffersV (message_id_type msgID, int32 numPairs, va_list arguments)`
- `OSCL_IMPORT_REF void LogMsgString (message_id_type msgID, const char *fmt,...)`
- `OSCL_IMPORT_REF void LogMsgBuffers (message_id_type msgID, int32 numPairs,...)`
- `OSCL_IMPORT_REF PVLogger (const char *inputTag, log_level_type level, bool oAppenderInheritance)`
- `virtual ~PVLogger ()`

Static Public Methods

- `OSCL_IMPORT_REF void Init ()`
- `OSCL_IMPORT_REF void Cleanup ()`
- `OSCL_IMPORT_REF PVLogger * GetLoggerObject (const char *inputTag)`

Protected Methods

- `void SetParent (PVLogger *parentLogger)`
- `PVLogger * GetParent ()`

Friends

- class `PVLoggerRegistry`

6.247.1 Member Typedef Documentation

6.247.1.1 `typedef _OsclBasicAllocator PVLogger::alloc_type`

6.247.1.2 `typedef int32 PVLogger::filter_status_type`

6.247.1.3 `typedef int32 PVLogger::log_level_type`

6.247.1.4 `typedef int32 PVLogger::message_id_type`

6.247.2 Constructor & Destructor Documentation

6.247.2.1 `OSCL_IMPORT_REF PVLogger::PVLogger (const char * inputTag, log_level_type level, bool oAppenderInheritance)`

Logger Constructor

Parameters:

tag Logger tag, unique to a logging control point

level Active Log level of the logger

oAppenderInheritance

Returns:

NONE

6.247.2.2 `virtual PVLogger::~PVLogger () [inline, virtual]`

6.247.3 Member Function Documentation

6.247.3.1 `void PVLogger::AddAppender (OsclSharedPtr< PVLoggerAppender > & appender) [inline]`

This method adds an appender to the logging control point. Each logger maintains a list of appenders. Any msg to a logger if deemed active is logged to all the appenders.

Parameters:

appender pointer to the appender to add

Returns:

NONE

Exceptions:

leaves if out of memory

6.247.3.2 `void PVLogger::AddFilter (OsclSharedPtr< PVLoggerFilter > & filter) [inline]`

This method adds a message filter to the logging control point. Each logger maintains a list of filters. Any msg to a logger if deemed active is passed through the msg filters prior to logging.

Parameters:

msgFilter pointer to the filter to add

Returns:

NONE

Exceptions:

leaves if out of memory

6.247.3.3 OSCL_IMPORT_REF void PVLogger::Cleanup () [static]

Frees the PVLogger singleton used by the current thread. This must be called before thread exit. No messages can be logged after cleanup.

Returns:**6.247.3.4 void PVLogger::DisableAppenderInheritance () [inline]**

This method disables appender inheritance for the logging control point

6.247.3.5 OSCL_IMPORT_REF PVLogger* PVLogger::GetLoggerObject (const char * *inputTag*) [static]

This is a factory method to create a log control point, with a certain input tag. There is a central registry of all the loggers, with their corresponding tags, called PV Logger Registry. In case the logger with the specified tag exists in the global registry, it is returned, else a new one is created and a pointer to the same is returned.

Parameters:

inputTag logger tag, viz. "x.y.z"

level log level associated with the logging control point (All messages with log levels less than equal to the log level of the control point would be logged)

oAppenderInheritance

Returns:

PVLogger* Pointer to the logging control point

Exceptions:

leaves if out of memory

6.247.3.6 log_level_type PVLogger::GetLogLevel () [inline]

This method returns the log level of a control point. This could either have been set explicitly by the user (at the time of creation or later) or could have been inherited from one of its ancestors.

Returns:

log level associated with the logging control point

6.247.3.7 uint32 PVLogger::GetNumAppenders () [inline]

This method returns the number of appenders attached to the logging control point.

6.247.3.8 PVLogger* PVLogger::GetParent () [inline, protected]**6.247.3.9 OSCL_IMPORT_REF void PVLogger::Init () [static]**

PVLogger needs to be initialized once per thread. This creates the PVLogger singleton that is used throughout the duration of the thread. Initialization must occur before the first message is logged.

Exceptions:

leaves if out of memory

6.247.3.10 OSCL_IMPORT_REF bool PVLogger::IsActive (*log_level_type level*)

This method determines if a msg passed to the logging control point is active or not. Only messages that are deemed active are logged. Messages are considered not active if any of the following criteria are met:

- All logging is disabled at this logging control point

If all the log levels, leading upto the root log point are uninitialized

- If the log level of the incoming message is LESS THAN that of the active log level of the logging control point.

Returns:

BOOL

6.247.3.11 OSCL_IMPORT_REF void PVLogger::LogMsgBuffers (*message_id_type msgID, int32 numPairs, ...*)

This method logs opaque data buffers to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

Parameters:

msgID Message ID, that is unique to a message

numPairs Number of (ptr_len, ptr) pairs

arguments Variable list of arguments

Returns:

NONE

6.247.3.12 OSCL_IMPORT_REF void PVLogger::LogMsgBuffersV (*message_id_type msgID, int32 numPairs, va_list arguments*)

This method logs opaque data buffers to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

Parameters:

msgID Message ID, that is unique to a message

numPairs Number of (ptr_len, ptr) pairs

arguments Variable list of arguments

Returns:

NONE

6.247.3.13 OSCL_IMPORT_REF void PVLogger::LogMsgString (*message_id_type msgID, const char *fmt, ...*)

This method logs formatted text msg to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

Parameters:

msgID Message ID, that is unique to a message

fmt format string, similar to one taken by printf

arguments Variable list of arguments

Returns:

NONE

6.247.3.14 OSCL_IMPORT_REF void PVLogger::LogMsgStringV (*message_id_type msgID, const char *fmt, va_list arguments*)

This method logs formatted text msg to all the appenders, after running through the message filters. After logging the message to the appenders attached to the current control point, the message is passed up to the parent node, only if appender inheritance is enabled.

Parameters:

msgID Message ID, that is unique to a message

fmt format string, similar to one taken by printf

arguments Variable list of arguments

Returns:

NONE

**6.247.3.15 void PVLogger::RemoveAppender (*OsclSharedPtr< PVLoggerAppender >* &
appender) [inline]**

This method removes an appender from the logging control point. Each logger maintains a list of appenders. Any msg to a logger if deemed active is logged to all the appenders.

Parameters:

appender pointer to the appender to delete

Returns:

NONE

6.247.3.16 void PVLogger::SetLogLevel (*log_level_type level*) [inline]

This method is used to set the log level of a control point.

Parameters:

level log level associated with the logging control point

Returns:

NONE

6.247.3.17 OSCL_IMPORT_REF void PVLogger::SetLogLevelAndPropagate (*log_level_type level*)

This method is used to set the log level of a control point, as well as to propagate the level to all the descendants of this control point.

Parameters:

level log level associated with the logging control point

Returns:

NONE

6.247.3.18 void PVLogger::SetParent (*PVLogger *parentLogger*) [inline, protected]**6.247.4 Friends And Related Function Documentation****6.247.4.1 friend class PVLoggerRegistry [friend]**

The documentation for this class was generated from the following file:

- [pvlogger.h](#)

6.248 PVLoggerAppender Class Reference

```
#include <pvlogger_accessories.h>
```

Public Types

- `typedef PVLogger::message_id_type message_id_type`

Public Methods

- `virtual ~PVLoggerAppender ()`
- `virtual void AppendString (message_id_type msgID, const char *fmt, va_list va)=0`
- `virtual void AppendBuffers (message_id_type msgID, int32 numPairs, va_list va)=0`

6.248.1 Detailed Description

Base class for all message appenders. This class defines the interface to the message appenders. There are two kinds of msg appender APIs, one to append text messages, and other to append opaque message buffers.

6.248.2 Member Typedef Documentation

6.248.2.1 `typedef PVLogger::message_id_type PVLoggerAppender::message_id_type`

6.248.3 Constructor & Destructor Documentation

6.248.3.1 `virtual PVLoggerAppender::~PVLoggerAppender () [inline, virtual]`

6.248.4 Member Function Documentation

6.248.4.1 `virtual void PVLoggerAppender::AppendBuffers (message_id_type msgID, int32 numPairs, va_list va) [pure virtual]`

6.248.4.2 `virtual void PVLoggerAppender::AppendString (message_id_type msgID, const char *fmt, va_list va) [pure virtual]`

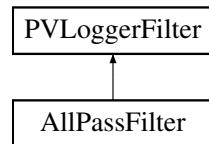
The documentation for this class was generated from the following file:

- `pvlogger_accessories.h`

6.249 PVLoggerFilter Class Reference

```
#include <pvlogger_accessories.h>
```

Inheritance diagram for PVLoggerFilter::



Public Types

- `typedef PVLogger::message_id_type message_id_type`
- `typedef PVLogger::log_level_type log_level_type`
- `typedef PVLogger::filter_status_type filter_status_type`

Public Methods

- `virtual ~PVLoggerFilter ()`
- `virtual filter_status_type FilterString (char *tag, message_id_type msgID, log_level_type level)=0`
- `virtual filter_status_type FilterOpaqueMessge (char *tag, message_id_type msgID, log_level_type level)=0`

6.249.1 Detailed Description

Base class for all message filters. This class defines the interface to the message filters. There are two kinds of msg filtering APIs, one to filter text messages, and other to filter opaque message buffers.

6.249.2 Member Typedef Documentation

6.249.2.1 `typedef PVLogger::filter_status_type PVLoggerFilter::filter_status_type`

Reimplemented in [AllPassFilter](#).

6.249.2.2 `typedef PVLogger::log_level_type PVLoggerFilter::log_level_type`

Reimplemented in [AllPassFilter](#).

6.249.2.3 `typedef PVLogger::message_id_type PVLoggerFilter::message_id_type`

Reimplemented in [AllPassFilter](#).

6.249.3 Constructor & Destructor Documentation

6.249.3.1 `virtual PVLoggerFilter::~PVLoggerFilter () [inline, virtual]`

6.249.4 Member Function Documentation

6.249.4.1 `virtual filter_status_type PVLoggerFilter::FilterOpaqueMessge (char * tag, message_id_type msgID, log_level_type level) [pure virtual]`

Implemented in [AllPassFilter](#).

6.249.4.2 `virtual filter_status_type PVLoggerFilter::FilterString (char * tag, message_id_type msgID, log_level_type level) [pure virtual]`

Implemented in [AllPassFilter](#).

The documentation for this class was generated from the following file:

- [pvlogger_accessories.h](#)

6.250 PVLoggerLayout Class Reference

```
#include <pvlogger_accessories.h>
```

Public Types

- `typedef PVLogger::message_id_type message_id_type`

Public Methods

- `virtual ~PVLoggerLayout ()`
- `virtual int32 FormatString (char *formatBuf, int32 formatBufSize, message_id_type msgID, const char *fmt, va_list va)=0`
- `virtual int32 FormatOpaqueMessage (char *formatBuf, int32 formatBufSize, message_id_type msgID, int32 numPairs, va_list va)=0`

6.250.1 Detailed Description

Base class for all message formatters. This class defines the interface to the message formatter. There are two kinds of msg formatting APIs, one to format text messages, and other to format opaque message buffers.

6.250.2 Member Typedef Documentation

6.250.2.1 `typedef PVLogger::message_id_type PVLoggerLayout::message_id_type`

6.250.3 Constructor & Destructor Documentation

6.250.3.1 `virtual PVLoggerLayout::~PVLoggerLayout () [inline, virtual]`

6.250.4 Member Function Documentation

6.250.4.1 `virtual int32 PVLoggerLayout::FormatOpaqueMessage (char *formatBuf, int32 formatBufSize, message_id_type msgID, int32 numPairs, va_list va) [pure virtual]`

Formats the data and copies it to the given buffer.

Returns:

The length of the buffer used.

6.250.4.2 `virtual int32 PVLoggerLayout::FormatString (char *formatBuf, int32 formatBufSize, message_id_type msgID, const char *fmt, va_list va) [pure virtual]`

Formats the string and copies it to the given buffer.

Returns:

The length of the string not including the trailing '\0'

The documentation for this class was generated from the following file:

- [pvlogger_accessories.h](#)

6.251 PVLoggerRegistry Class Reference

```
#include <pvlogger_registry.h>
```

Public Types

- `typedef PVLogger::log_level_type log_level_type`
- `typedef PVLogger::alloc_type alloc_type`

Public Methods

- `OSCL_IMPORT_REF PVLoggerRegistry()`
- `virtual OSCL_IMPORT_REF ~PVLoggerRegistry()`
- `OSCL_IMPORT_REF PVLogger * GetPVLoggerObject (const char *tagIn)`
- `OSCL_IMPORT_REF PVLogger * CreatePVLogger (const char *tagIn, log_level_type level, bool oAppenderInheritance)`
- `OSCL_IMPORT_REF bool SetNodeLogLevelExplicit (char *tagIn, log_level_type level)`
- `OSCL_IMPORT_REF void SetNodeLogLevelExplicit (Oscl_TagTree< PVLogger *, alloc_type >::node_type *node, log_level_type level)`

Static Public Methods

- `OSCL_IMPORT_REF PVLoggerRegistry * GetPVLoggerRegistry()`

6.251.1 Detailed Description

Class: PVLoggerRegistry

PVLoggerRegistry class, maintains a repository of all the loggers, along with their associated tags, in a tag tree. Any request for a log control point is serviced by this class.

Memory Ownership: Creates log control points for each tag, and holds these pointers in the tag tree. `PVLogger` registry is responsible for calling the destructor on each of these loggers.

6.251.2 Member Typedef Documentation

6.251.2.1 `typedef PVLogger::alloc_type PVLoggerRegistry::alloc_type`

6.251.2.2 `typedef PVLogger::log_level_type PVLoggerRegistry::log_level_type`

6.251.3 Constructor & Destructor Documentation

6.251.3.1 `OSCL_IMPORT_REF PVLoggerRegistry::PVLoggerRegistry()`

PVLoggerRegistry Constructor

6.251.3.2 `virtual OSCL_IMPORT_REF PVLoggerRegistry::~PVLoggerRegistry () [virtual]`

PVLoggerRegistry Destructor

6.251.4 Member Function Documentation

6.251.4.1 OSCL_IMPORT_REF PVLogger* PVLoggerRegistry::CreatePVLogger (const char * *tagIn*, log_level_type *level*, bool *oAppenderInheritance*)

This method creates a log control point, with specified tag, and level

Parameters:

inputTag logger tag, viz. "x.y.z"
level log level associated with the logging control point
oAppenderInheritance

Returns:

PVLogger<alloc_type, TheLock>* Pointer to the logging control point

6.251.4.2 OSCL_IMPORT_REF PVLogger* PVLoggerRegistry::GetPVLoggerObject (const char * *tagIn*)

PVLoggerRegistry method to get access to a logging control point, associated with a tag. In case the logger for this tag does not exist, it is created afresh, else pointer to the existing one is returned.

Parameters:

inputTag logger tag, viz. "x.y.z"
level log level associated with the logging control point
oAppenderInheritance

Returns:

PVLogger<Alloc, TheLock>* Pointer to the logging control point

6.251.4.3 OSCL_IMPORT_REF PVLoggerRegistry* PVLoggerRegistry::GetPVLoggerRegistry () [static]

Get the logger registry. There is only one logger registry instance per thread.

6.251.4.4 OSCL_IMPORT_REF void PVLoggerRegistry::SetNodeLogLevelExplicit (Oscl_TagTree< PVLogger *, alloc_type >::node_type * *node*, log_level_type *level*)

This method recursively propagates the log level to all the descendants, of a node.

Parameters:

node Node ptr, associated with a logger, from the tag tree.
level log level associated with the logging control point

Returns:

NONE

6.251.4.5 OSCL_IMPORT_REF bool PVLoggerRegistry::SetNodeLogLevelExplicit (char * *tagIn*, *log_level_type level*)

This method propagates the log level to all the descendants of the node, with a specified tag.

Parameters:

tagIn logger tag, viz. "x.y.z"

level log level associated with the logging control point

Returns:

true on success, else false.

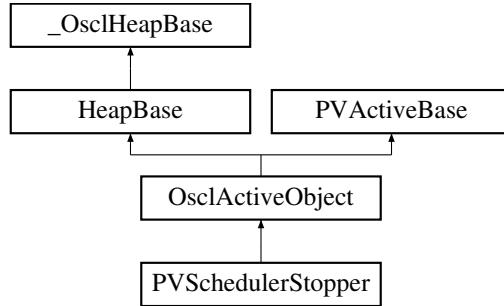
The documentation for this class was generated from the following file:

- [pvlogger_registry.h](#)

6.252 PVSchedulerStopper Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for PVSchedulerStopper::



Public Methods

- [PVSchedulerStopper \(\)](#)
- [~PVSchedulerStopper \(\)](#)

6.252.1 Detailed Description

Scheduler stopper AO class, for internal use by scheduler.

6.252.2 Constructor & Destructor Documentation

6.252.2.1 PVSchedulerStopper::PVSchedulerStopper ()

6.252.2.2 PVSchedulerStopper::~PVSchedulerStopper ()

The documentation for this class was generated from the following file:

- [oscl_scheduler.h](#)

6.253 PVSockBufRecv Class Reference

```
#include <oscl_socket_request.h>
```

Public Methods

- [PVSockBufRecv \(\)](#)
- [PVSockBufRecv \(uint8 *aPtr, uint32 aLen, uint32 aMax\)](#)
- [PVSockBufRecv \(const PVSockBufRecv &a\)](#)

Data Fields

- [uint8 * iPtr](#)
- [uint32 iLen](#)
- [uint32 iMaxLen](#)

6.253.1 Constructor & Destructor Documentation

6.253.1.1 PVSockBufRecv::PVSockBufRecv () [inline]

6.253.1.2 PVSockBufRecv::PVSockBufRecv (uint8 * *aPtr*, uint32 *aLen*, uint32 *aMax*) [inline]

6.253.1.3 PVSockBufRecv::PVSockBufRecv (const PVSockBufRecv & *a*) [inline]

6.253.2 Field Documentation

6.253.2.1 uint32 PVSockBufRecv::iLen

6.253.2.2 uint32 PVSockBufRecv::iMaxLen

6.253.2.3 uint8* PVSockBufRecv::iPtr

The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

6.254 PVSockBufSend Class Reference

```
#include <oscl_socket_request.h>
```

Public Methods

- [PVSockBufSend \(\)](#)
- [PVSockBufSend \(const uint8 *aPtr, uint32 aLen\)](#)
- [PVSockBufSend \(const PVSockBufSend &a\)](#)

Data Fields

- [const uint8 * iPtr](#)
- [uint32 iLen](#)

6.254.1 Constructor & Destructor Documentation

6.254.1.1 PVSockBufSend::PVSockBufSend () [inline]

6.254.1.2 PVSockBufSend::PVSockBufSend (const uint8 * aPtr, uint32 aLen) [inline]

6.254.1.3 PVSockBufSend::PVSockBufSend (const PVSockBufSend & a) [inline]

6.254.2 Field Documentation

6.254.2.1 uint32 PVSockBufSend::iLen

6.254.2.2 const uint8* PVSockBufSend::iPtr

The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

6.255 PVThreadContext Class Reference

```
#include <oscl_scheduler_threadcontext.h>
```

Public Methods

- OSCL_IMPORT_REF PVThreadContext ()
- OSCL_IMPORT_REF ~PVThreadContext ()
- OSCL_IMPORT_REF bool IsSameThreadContext ()
- OSCL_IMPORT_REF void EnterThreadContext ()
- OSCL_IMPORT_REF void ExitThreadContext ()

Static Public Methods

- OSCL_IMPORT_REF uint32 Id ()
- OSCL_IMPORT_REF bool ThreadHasScheduler ()

Friends

- class PVActiveBase
- class OsclActiveObject
- class OsclTimerObject
- class OsclExecScheduler
- class OsclCoeActiveScheduler
- class OsclExecSchedulerCommonBase
- class OsclExecSchedulerBase
- class OsclCoeActiveSchedulerBase

6.255.1 Constructor & Destructor Documentation

6.255.1.1 OSCL_IMPORT_REF PVThreadContext::PVThreadContext ()

6.255.1.2 OSCL_IMPORT_REF PVThreadContext::~PVThreadContext ()

6.255.2 Member Function Documentation

6.255.2.1 OSCL_IMPORT_REF void PVThreadContext::EnterThreadContext ()

enter and exit thread context.

6.255.2.2 OSCL_IMPORT_REF void PVThreadContext::ExitThreadContext ()

6.255.2.3 OSCL_IMPORT_REF uint32 PVThreadContext::Id () [static]

static routine to get a unique thread ID for caller's thread context.

6.255.2.4 OSCL_IMPORT_REF bool PVThreadContext::IsSameThreadContext ()

compare caller's thread context to this one.

6.255.2.5 OSCL_IMPORT_REF bool PVThreadContext::ThreadHasScheduler () [static]

a static utility to tell whether the calling thread has any scheduler— either Oscl scheduler or native scheduler.

6.255.3 Friends And Related Function Documentation**6.255.3.1 friend class OsclActiveObject [friend]****6.255.3.2 friend class OsclCoeActiveScheduler [friend]****6.255.3.3 friend class OsclCoeActiveSchedulerBase [friend]****6.255.3.4 friend class OsclExecScheduler [friend]****6.255.3.5 friend class OsclExecSchedulerBase [friend]****6.255.3.6 friend class OsclExecSchedulerCommonBase [friend]****6.255.3.7 friend class OsclTimerObject [friend]****6.255.3.8 friend class PVActiveBase [friend]**

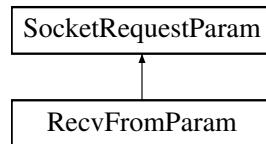
The documentation for this class was generated from the following file:

- [oscl_scheduler_threadcontext.h](#)

6.256 RecvFromParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for RecvFromParam::



Public Methods

- [RecvFromParam \(uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, uint32 flags, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource\)](#)

Data Fields

- [PVSockBufRecv iBufRecv](#)
- [uint32 iFlags](#)
- [OsclNetworkAddress & iAddr](#)
- [uint32 iMultiMaxLen](#)
- [Oscl_Vector< uint32, OsclMemAllocator > * iPacketLen](#)
- [Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * iPacketSource](#)

6.256.1 Constructor & Destructor Documentation

[6.256.1.1 RecvFromParam::RecvFromParam \(uint8 *& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, uint32 flags, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > * aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aPacketSource\) \[inline\]](#)

6.256.2 Field Documentation

[6.256.2.1 OsclNetworkAddress& RecvFromParam::iAddr](#)

[6.256.2.2 PVSockBufRecv RecvFromParam::iBufRecv](#)

[6.256.2.3 uint32 RecvFromParam::iFlags](#)

[6.256.2.4 uint32 RecvFromParam::iMultiMaxLen](#)

[6.256.2.5 Oscl_Vector<uint32, OsclMemAllocator>* RecvFromParam::iPacketLen](#)

[6.256.2.6 Oscl_Vector<OsclNetworkAddress, OsclMemAllocator>*> RecvFromParam::iPacketSource](#)

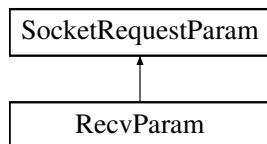
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

6.257 RecvParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for RecvParam::



Public Methods

- [RecvParam \(uint8 *&aPtr, uint32 aMaxLen, uint32 flags\)](#)

Data Fields

- [PVSockBufRecv iBufRecv](#)
- [uint32 iFlags](#)

6.257.1 Constructor & Destructor Documentation

6.257.1.1 RecvParam::RecvParam (uint8 *& aPtr, uint32 aMaxLen, uint32 flags) [inline]

6.257.2 Field Documentation

6.257.2.1 PVSockBufRecv RecvParam::iBufRecv

6.257.2.2 uint32 RecvParam::iFlags

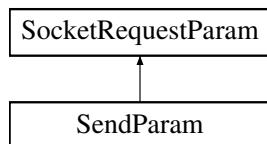
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

6.258 SendParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SendParam::



Public Methods

- [SendParam \(const uint8 *&aPtr, uint32 aLen, uint32 aFlags\)](#)

Data Fields

- [PVSockBufSend iBufSend](#)
- [uint32 iFlags](#)
- [uint32 iXferLen](#)

6.258.1 Detailed Description

Socket method parameter sets

6.258.2 Constructor & Destructor Documentation

6.258.2.1 [SendParam::SendParam \(const uint8 *& aPtr, uint32 aLen, uint32 aFlags\) \[inline\]](#)

6.258.3 Field Documentation

6.258.3.1 [PVSockBufSend SendParam::iBufSend](#)

6.258.3.2 [uint32 SendParam::iFlags](#)

6.258.3.3 [uint32 SendParam::iXferLen](#)

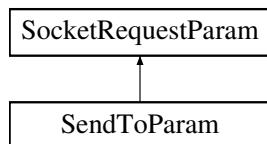
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

6.259 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](#)

6.259.1 Constructor & Destructor Documentation

6.259.1.1 SendToParam::SendToParam (const uint8 *& *aPtr*, uint32 *aLen*, OsclNetworkAddress & *anAddr*, uint32 *flags*) [inline]

6.259.1.2 SendToParam::~SendToParam () [inline]

6.259.2 Field Documentation

6.259.2.1 OsclNetworkAddress SendToParam::iAddr

6.259.2.2 PVSockBufSend SendToParam::iBufSend

6.259.2.3 uint32 SendToParam::iFlags

6.259.2.4 uint32 SendToParam::iXferLen

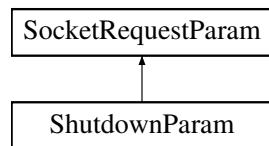
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

6.260 ShutdownParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for ShutdownParam::



Public Methods

- [ShutdownParam \(TPVSocketShutdown aHow\)](#)

Data Fields

- [TPVSocketShutdown iHow](#)

6.260.1 Constructor & Destructor Documentation

6.260.1.1 ShutdownParam::ShutdownParam ([TPVSocketShutdown aHow](#)) [inline]

6.260.2 Field Documentation

6.260.2.1 [TPVSocketShutdown ShutdownParam::iHow](#)

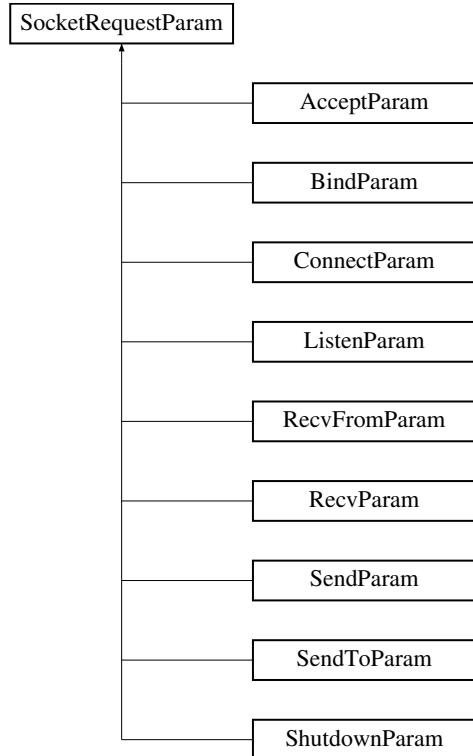
The documentation for this class was generated from the following file:

- [oscl_socket_request.h](#)

6.261 SocketRequestParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SocketRequestParam::



Public Methods

- [SocketRequestParam \(TPVSocketFxn aFxn\)](#)

Data Fields

- [TPVSocketFxn iFxn](#)

6.261.1 Detailed Description

Base class for all socket method parameter sets

6.261.2 Constructor & Destructor Documentation

6.261.2.1 `SocketRequestParam::SocketRequestParam (TPVSocketFxn aFxn) [inline]`

6.261.3 Field Documentation

6.261.3.1 `TPVSocketFxn SocketRequestParam::iFxn`

The documentation for this class was generated from the following file:

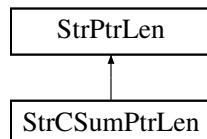
- `oscl_socket_request.h`

6.262 StrCSumPtrLen Struct Reference

same as [StrPtrLen](#), but includes checksum field and method to speed up querying

```
#include <oscl_str_ptr_len.h>
```

Inheritance diagram for StrCSumPtrLen::



Public Types

- [typedef int16 CheckSumType](#)

Public Methods

- [void setPtrLen \(const char *newPtr, uint32 newLen\)](#)
- [CheckSumType getCheckSum \(\) const](#)
- [OSCL_IMPORT_REF void setCheckSum \(\)](#)
- [StrCSumPtrLen \(\)](#)
- [StrCSumPtrLen \(const char *newPtr\)](#)
- [StrCSumPtrLen \(const char *newPtr, uint32 newLen\)](#)
- [StrCSumPtrLen \(const StrCSumPtrLen &rhs\)](#)
- [StrCSumPtrLen \(const StrPtrLen &rhs\)](#)
- [c_bool isCIEquivalentTo \(const StrCSumPtrLen &rhs\) const](#)
- [c_bool operator== \(const StrCSumPtrLen &rhs\) const](#)
- [c_bool operator!= \(const StrCSumPtrLen &rhs\) const](#)
- [StrCSumPtrLen & operator= \(const StrCSumPtrLen &rhs\)](#)
- [StrCSumPtrLen & operator= \(const StrPtrLen &rhs\)](#)
- [StrCSumPtrLen & operator= \(const char *rhs\)](#)

Protected Attributes

- [CheckSumType checkSum](#)

6.262.1 Detailed Description

same as [StrPtrLen](#), but includes checksum field and method to speed up querying

6.262.2 Member Typedef Documentation

6.262.2.1 `typedef int16 StrCSumPtrLen::CheckSumType`

6.262.3 Constructor & Destructor Documentation

6.262.3.1 `StrCSumPtrLen::StrCSumPtrLen () [inline]`

6.262.3.2 `StrCSumPtrLen::StrCSumPtrLen (const char * newPtr) [inline]`

6.262.3.3 `StrCSumPtrLen::StrCSumPtrLen (const char * newPtr, uint32 newLen) [inline]`

6.262.3.4 `StrCSumPtrLen::StrCSumPtrLen (const StrCSumPtrLen & rhs) [inline]`

6.262.3.5 `StrCSumPtrLen::StrCSumPtrLen (const StrPtrLen & rhs) [inline]`

6.262.4 Member Function Documentation

6.262.4.1 `CheckSumType StrCSumPtrLen::getCheckSum () const [inline]`

6.262.4.2 `c_bool StrCSumPtrLen::isCIEquivalentTo (const StrCSumPtrLen & rhs) const [inline]`

6.262.4.3 `c_bool StrCSumPtrLen::operator!= (const StrCSumPtrLen & rhs) const [inline]`

6.262.4.4 `StrCSumPtrLen& StrCSumPtrLen::operator= (const char * rhs) [inline]`

Reimplemented from [StrPtrLen](#).

6.262.4.5 `StrCSumPtrLen& StrCSumPtrLen::operator= (const StrPtrLen & rhs) [inline]`

Reimplemented from [StrPtrLen](#).

6.262.4.6 `StrCSumPtrLen& StrCSumPtrLen::operator= (const StrCSumPtrLen & rhs) [inline]`

6.262.4.7 `c_bool StrCSumPtrLen::operator== (const StrCSumPtrLen & rhs) const [inline]`

6.262.4.8 `OSCL_IMPORT_REF void StrCSumPtrLen::setCheckSum ()`

6.262.4.9 `void StrCSumPtrLen::setPtrLen (const char * newPtr, uint32 newLen) [inline]`

Reimplemented from [StrPtrLen](#).

6.262.5 Field Documentation

6.262.5.1 `CheckSumType StrCSumPtrLen::checkSum [protected]`

The documentation for this struct was generated from the following file:

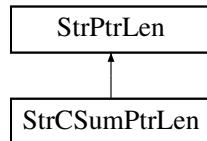
- [oscl_str_ptr_len.h](#)

6.263 StrPtrLen Struct Reference

This data structure encapsulates a set of functions used to perform.

```
#include <oscl_str_ptr_len.h>
```

Inheritance diagram for StrPtrLen::



Public Methods

- [StrPtrLen](#) (const char *newPtr)
- [StrPtrLen](#) (const char *newPtr, uint32 newLen)
- [StrPtrLen](#) ()
- [StrPtrLen](#) (const StrPtrLen &rhs)
- const char * [c_str](#) () const
- int32 [length](#) () const
- int32 [size](#) () const
- void [setPtrLen](#) (const char *newPtr, uint32 newLen)
- [c_bool](#) [isCIEquivalentTo](#) (const StrPtrLen &rhs) const
- [c_bool](#) [isCIPrefixOf](#) (const StrPtrLen &rhs) const
- int32 [operator==](#) (const StrPtrLen &rhs) const
- int32 [operator!=](#) (const StrPtrLen &rhs) const
- StrPtrLen & [operator=](#) (const StrPtrLen &rhs)
- StrPtrLen & [operator=](#) (const char *rhs)

Protected Methods

- bool [isLetter](#) (const char c) const

Protected Attributes

- const char * [ptr](#)
- int32 [len](#)

6.263.1 Detailed Description

This data structure encapsulates a set of functions used to perform.

standard string operations. It should be used for null-terminated constant (non-modifiable) strings of char type.

6.263.2 Constructor & Destructor Documentation

6.263.2.1 `StrPtrLen::StrPtrLen (const char * newPtr) [inline]`

6.263.2.2 `StrPtrLen::StrPtrLen (const char * newPtr, uint32 newLen) [inline]`

6.263.2.3 `StrPtrLen::StrPtrLen () [inline]`

6.263.2.4 `StrPtrLen::StrPtrLen (const StrPtrLen & rhs) [inline]`

6.263.3 Member Function Documentation

6.263.3.1 `const char* StrPtrLen::c_str () const [inline]`

6.263.3.2 `c_bool StrPtrLen::isCIEquivalentTo (const StrPtrLen & rhs) const [inline]`

6.263.3.3 `c_bool StrPtrLen::isCIPrefixOf (const StrPtrLen & rhs) const [inline]`

6.263.3.4 `bool StrPtrLen::isLetter (const char c) const [inline, protected]`

6.263.3.5 `int32 StrPtrLen::length () const [inline]`

6.263.3.6 `int32 StrPtrLen::operator!= (const StrPtrLen & rhs) const [inline]`

6.263.3.7 `StrPtrLen& StrPtrLen::operator= (const char * rhs) [inline]`

Reimplemented in [StrCSumPtrLen](#).

6.263.3.8 `StrPtrLen& StrPtrLen::operator= (const StrPtrLen & rhs) [inline]`

Reimplemented in [StrCSumPtrLen](#).

6.263.3.9 `int32 StrPtrLen::operator== (const StrPtrLen & rhs) const [inline]`

6.263.3.10 `void StrPtrLen::setPtrLen (const char * newPtr, uint32 newLen) [inline]`

Reimplemented in [StrCSumPtrLen](#).

6.263.3.11 `int32 StrPtrLen::size () const [inline]`

6.263.4 Field Documentation

6.263.4.1 `int32 StrPtrLen::len [protected]`

6.263.4.2 `const char* StrPtrLen::ptr [protected]`

The documentation for this struct was generated from the following file:

- [oscl_str_ptr_len.h](#)

6.264 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
- OSCL_COND_IMPORT_REF uint32 [get_usec](#) () 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 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+= \(const TimeValue &a\)](#)
+ = operator
- OSCL_COND_IMPORT_REF TimeValue & [operator-= \(const TimeValue &a\)](#)
- = operator
- OSCL_COND_IMPORT_REF TimeValue & [operator *= \(const int scale\)](#)
This operator scales the time value by a constant.
- OSCL_COND_IMPORT_REF OsclBasicTimeStruct * [get_timeval_ptr \(\)](#)

Friends

- class [NTPTime](#)
- OSCL_COND_IMPORT_REF friend bool [operator== \(const TimeValue &a, const TimeValue &b\)](#)
- OSCL_COND_IMPORT_REF friend bool [operator!= \(const TimeValue &a, const TimeValue &b\)](#)
- OSCL_COND_IMPORT_REF friend bool [operator<= \(const TimeValue &a, const TimeValue &b\)](#)
- OSCL_COND_IMPORT_REF friend bool [operator>= \(const TimeValue &a, const TimeValue &b\)](#)
- OSCL_COND_IMPORT_REF friend bool [operator< \(const TimeValue &a, const TimeValue &b\)](#)
- OSCL_COND_IMPORT_REF friend bool [operator> \(const TimeValue &a, const TimeValue &b\)](#)

6.264.1 Detailed Description

The TimeValue class represents a time value in a format native to the system.

This class provides common time functions independent of any OS. The actual representation used is native to the system that the class is compiled on to increase efficiency. Macros used in this class:

- OSCL_HAS_ANSI_STRING_SUPPORT:

Definitions to determine the type of basic time structure used to store the time

- OSCL_HAS_UNIX_TIME_FUNCS
- OSCL_HAS_SYMBIAN_SUPPORT
- OSCL_HAS_MSWIN_SUPPORT

6.264.2 Constructor & Destructor Documentation

6.264.2.1 OSCL_COND_IMPORT_REF TimeValue::TimeValue ()

Create a TimeValue representing the current time.

6.264.2.2 OSCL_COND_IMPORT_REF TimeValue::TimeValue (const TimeValue & *Tv*)

Copy constructor.

6.264.2.3 OSCL_COND_IMPORT_REF TimeValue::TimeValue (long *tv*, TimeUnits *units*)

Create a TimeValue representing an interval of tv units.

Parameters:

tv The number of units in the interval to be represented by this TimeValue.

units The units of the tv argument. Must be in the enumeration TimeUnits.

6.264.2.4 OSCL_COND_IMPORT_REF TimeValue::TimeValue (const OsclBasicTimeStruct & *in_tv*)

Create a TimeValue representing the absolute time specified by the BasicTimeStruct.

Parameters:

in_tv OsclBasicTimeStruct as defined for each platform.

6.264.2.5 OSCL_COND_IMPORT_REF TimeValue::TimeValue (OsclBasicDateTimeStruct *in_ts*)

Create a TimeValue representing the absolute time specified by the BasicDateTimeStruct.

Parameters:

in_ts OsclBasicDateTimeStruct as defined for each platform provides the date in a readable format (i.e. day, date , week etc.) Notes: Implementation incomplete (= not done) on Win32, Wince, Symbian

6.264.3 Member Function Documentation

6.264.3.1 OSCL_COND_IMPORT_REF int32 TimeValue::get_local_time ()

Get the local time after having adjusted for daylight saving.

Notes: Implementation incomplete (= not done) on Win32, Wince, Symbian

6.264.3.2 OSCL_IMPORT_REF int TimeValue::get_pv8601_str_time (PV8601timeStrBuf *time_strbuf*)

Get a PV extended text representation of the Time based on the ISO 8601 format.

Parameters:

time_strbuf A PV8601timeStrBuf object to which the text representation will be written,

Returns:

The number of characters copied to the buffer, not including the terminating null. The returned string is of the form "19850412T101530.047Z".

6.264.3.3 OSCL_IMPORT_REF char* TimeValue::get_rfc822_gmtime_str (int max_time_strlen, char * time_str)

Get a text representation of the time in the GMT timezone based on the RFC 822 / RFC 1123 (also described in the HTTP spec RFC 2068 and RFC 2616).

Parameters:

max_time_strlen The maximum number of characters that can be written to the buffer.

time_str A pointer to the buffer to which the characters will be written.

Returns:

Returns a pointer to the buffer (same as *time_str*) containing a null terminated (c-style) string of the form "Wed, 30 Jun 1993 21:49:08 GMT".

6.264.3.4 OSCL_COND_IMPORT_REF uint32 TimeValue::get_sec ()

Get a 32 bit value representing the seconds since the (system dependent) epoch.

Returns:

This call returns a 32 bit value representing the number of seconds since the epoch. On unix systems this represents the number of seconds since the unix epoch Jan 1 1970. On Win32 this represents the number of seconds since Jan 1, 1601. This is intended to be used for intervals rather than for absolute time. (On Win32 for example, a 32 bit value would be too small to represent the number of seconds from the epoch until the current time.)

6.264.3.5 OSCL_IMPORT_REF char* TimeValue::get_str_ctime (CtimeStrBuf ctime_strbuf)

Get a string containing a text representation of this TimeValue object.

Parameters:

ctime_strbuf A CtimeStrBuf object to which the text representation will be written,

Returns:

A pointer to the input CtimeStrBuf is returned. This string is null terminated of the form "Wed Jun 30 21:49:08 1993".

6.264.3.6 OSCL_COND_IMPORT_REF OsclBasicTimeStruct* TimeValue::get_timeval_ptr ()**6.264.3.7 OSCL_COND_IMPORT_REF uint32 TimeValue::get_usec ()**

Get a 32 bit value representing the number of microseconds in the time value.

Returns:

Returns a uint32 value representing the number of microseconds in the time value after subtracting off the whole seconds.

6.264.3.8 OSCL_COND_IMPORT_REF bool TimeValue::is_zero ()

Determine if the time value is zero.

6.264.3.9 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator *= (const int scale)

This operator scales the time value by a constant.

6.264.3.10 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator+= (const TimeValue & a)

+= operator

6.264.3.11 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator-= (const TimeValue & a)

-= operator

6.264.3.12 OSCL_COND_IMPORT_REF TimeValue& TimeValue::operator= (const TimeValue & a)

Assignment operator.

6.264.3.13 OSCL_COND_IMPORT_REF void TimeValue::set_from_ntp_time (const uint32 ntp_offset)

This method converts a 32-bit NTP offset to system time.

This method takes a 32-bit ntp offset which is the number of seconds from 0 h Jan 1, 1900 and converts it to the system time

6.264.3.14 OSCL_COND_IMPORT_REF void TimeValue::set_to_current_time ()

Set the time value to the current system time.

6.264.3.15 OSCL_COND_IMPORT_REF void TimeValue::set_to_zero ()

Set the time value to zero (represents a zero interval).

6.264.3.16 OSCL_COND_IMPORT_REF int32 TimeValue::to_msec ()

6.264.4 Friends And Related Function Documentation

6.264.4.1 friend class NTPTime [friend]

6.264.4.2 OSCL_COND_IMPORT_REF friend bool operator!= (const TimeValue & a, const TimeValue & b) [friend]

6.264.4.3 OSCL_COND_IMPORT_REF friend bool operator< (const TimeValue & a, const TimeValue & b) [friend]

6.264.4.4 OSCL_COND_IMPORT_REF friend bool operator<= (const TimeValue & a, const TimeValue & b) [friend]

6.264.4.5 OSCL_COND_IMPORT_REF friend bool operator== (const TimeValue & a, const TimeValue & b) [friend]

6.264.4.6 OSCL_COND_IMPORT_REF friend bool operator> (const TimeValue & a, const TimeValue & b) [friend]

6.264.4.7 OSCL_COND_IMPORT_REF friend bool operator>= (const TimeValue & a, const TimeValue & b) [friend]

The documentation for this class was generated from the following file:

- [oscl_time.h](#)

6.265 TLSStorageOps Class Reference

```
#include <oscl_tls.h>
```

Static Public Methods

- OSCL_IMPORT_REF void [save_registry \(TOsclTlsKey *key, OsclAny *ptr, int32 &\)](#)
- OSCL_IMPORT_REF [OsclAny * get_registry \(TOsclTlsKey *key\)](#)

6.265.1 Member Function Documentation

**6.265.1.1 OSCL_IMPORT_REF OsclAny* TLSStorageOps::get_registry (TOsclTlsKey * *key*)
[static]**

**6.265.1.2 OSCL_IMPORT_REF void TLSStorageOps::save_registry (TOsclTlsKey * *key*,
OsclAny * *ptr*, int32 &) [static]**

The documentation for this class was generated from the following file:

- [oscl_tls.h](#)

6.266 TReadyQueLink Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Public Methods

- [TReadyQueLink \(\)](#)

Data Fields

- int32 [iAOPriority](#)
- uint32 [iTimeToRunTicks](#)
- uint32 [iTimeQueuedTicks](#)
- uint32 [iSeqNum](#)
- OsclAny * [iIsIn](#)

6.266.1 Detailed Description

This class defines the queue link, which is common to both ready Q and timer Q. Each AO contains its own queue link object.

6.266.2 Constructor & Destructor Documentation

6.266.2.1 [TReadyQueLink::TReadyQueLink \(\) \[inline\]](#)

6.266.3 Field Documentation

6.266.3.1 [int32 TReadyQueLink::iAOPriority](#)

6.266.3.2 [OsclAny* TReadyQueLink::iIsIn](#)

6.266.3.3 [uint32 TReadyQueLink::iSeqNum](#)

6.266.3.4 [uint32 TReadyQueLink::iTimeQueuedTicks](#)

6.266.3.5 [uint32 TReadyQueLink::iTimeToRunTicks](#)

The documentation for this class was generated from the following file:

- [oscl_scheduler_readyq.h](#)

6.267 WStrPtrLen Struct Reference

This data structure encapsulates a set of functions used to perform.

```
#include <oscl_str_ptr_len.h>
```

Public Methods

- [WStrPtrLen \(const oscl_wchar *newPtr\)](#)
- [WStrPtrLen \(const oscl_wchar *newPtr, uint32 newLen\)](#)
- [WStrPtrLen \(\)](#)
- [WStrPtrLen \(const WStrPtrLen &rhs\)](#)
- [const oscl_wchar * c_str \(\) const](#)
- [int32 length \(\) const](#)
- [int32 size \(\) const](#)
- [void setPtrLen \(const oscl_wchar *newPtr, uint32 newLen\)](#)
- [c_bool isCIEquivalentTo \(const WStrPtrLen &rhs\) const](#)
- [int32 operator== \(const WStrPtrLen &rhs\) const](#)
- [int32 operator!= \(const WStrPtrLen &rhs\) const](#)
- [WStrPtrLen & operator= \(const WStrPtrLen &rhs\)](#)
- [WStrPtrLen & operator= \(const oscl_wchar *rhs\)](#)

Protected Attributes

- [const oscl_wchar * ptr](#)
- [int32 len](#)

6.267.1 Detailed Description

This data structure encapsulates a set of functions used to perform.

standard string operations. It should be used for null-terminated constant strings (non-modifiable) of wchar type.

6.267.2 Constructor & Destructor Documentation

- 6.267.2.1 `WStrPtrLen::WStrPtrLen (const oscl_wchar * newPtr) [inline]`
- 6.267.2.2 `WStrPtrLen::WStrPtrLen (const oscl_wchar * newPtr, uint32 newLen) [inline]`
- 6.267.2.3 `WStrPtrLen::WStrPtrLen () [inline]`
- 6.267.2.4 `WStrPtrLen::WStrPtrLen (const WStrPtrLen & rhs) [inline]`

6.267.3 Member Function Documentation

- 6.267.3.1 `const oscl_wchar* WStrPtrLen::c_str () const [inline]`
- 6.267.3.2 `c_bool WStrPtrLen::isCIEquivalentTo (const WStrPtrLen & rhs) const [inline]`
- 6.267.3.3 `int32 WStrPtrLen::length () const [inline]`
- 6.267.3.4 `int32 WStrPtrLen::operator!= (const WStrPtrLen & rhs) const [inline]`
- 6.267.3.5 `WStrPtrLen& WStrPtrLen::operator= (const oscl_wchar * rhs) [inline]`
- 6.267.3.6 `WStrPtrLen& WStrPtrLen::operator= (const WStrPtrLen & rhs) [inline]`
- 6.267.3.7 `int32 WStrPtrLen::operator== (const WStrPtrLen & rhs) const [inline]`
- 6.267.3.8 `void WStrPtrLen::setPtrLen (const oscl_wchar * newPtr, uint32 newLen) [inline]`
- 6.267.3.9 `int32 WStrPtrLen::size () const [inline]`

6.267.4 Field Documentation

- 6.267.4.1 `int32 WStrPtrLen::len [protected]`
- 6.267.4.2 `const oscl_wchar* WStrPtrLen::ptr [protected]`

The documentation for this struct was generated from the following file:

- `oscl_str_ptr_len.h`

Chapter 7

oscl File Documentation

7.1 oscl_aostatus.h File Reference

Some basic types used with active objects.

```
#include "osclconfig_proc.h"  
#include "oscl_base.h"  
#include "oscl_aostatus.inl"
```

Data Structures

- class [OsclAOStatus](#)

Variables

- const int32 [OSCL_REQUEST_ERR_NONE](#) = 0
- const int32 [OSCL_REQUEST_PENDING](#) = (-0x7fffffff)
- const int32 [OSCL_REQUEST_ERR_CANCEL](#) = (-1)
- const int32 [OSCL_REQUEST_ERR_GENERAL](#) = (-2)

7.1.1 Detailed Description

Some basic types used with active objects.

7.2 oscl_assert.h File Reference

The file [oscl_assert.h](#) provides an OSCL_ASSERT macro to document assumptions and test them during development.

```
#include "oscl_base.h"  
#include "oscl_assert.inl"
```

Defines

- #define [OSCL_ASSERT](#)(_expr) ((_expr)?((void)0):OSCL Assert(# _expr,__FILE__,__LINE__))

Functions

- OSCL_COND_IMPORT_REF void [_OSCL_Abort](#) ()
This function terminates the current process abnormally.
- OSCL_IMPORT_REF void [OSCL Assert](#) (const char *expr, const char *filename, int line_number)
OSCL_ASSERT macro evaluates an expression and when the result is false, prints a diagnostic message and aborts the program.

7.2.1 Detailed Description

The file [oscl_assert.h](#) provides an OSCL_ASSERT macro to document assumptions and test them during development.

7.3 oscl_base.h File Reference

The file [oscl_base.h](#) is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros.

```
#include "osclconfig.h"
#include "oscl_base_macros.h"
#include "oscl_types.h"
#include "osclconfig_check.h"
```

Defines

- #define [OSCL_HAS_SINGLETON_SUPPORT](#) 1

Functions

- void [PVOsclBase_Init](#) ()
- void [PVOsclBase_Cleanup](#) ()

7.3.1 Detailed Description

The file [oscl_base.h](#) is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros.

7.4 oscl_base_alloc.h File Reference

A basic allocator that does not rely on other modules.

```
#include "osclconfig.h"  
#include "oscl_defalloc.h"  
#include "osclconfig_memory.h"
```

Data Structures

- class [_OsclBasicAllocator](#)

7.4.1 Detailed Description

A basic allocator that does not rely on other modules.

7.5 oscl_base_macros.h File Reference

This file defines common macros and constants for basic compilation support.

```
#include "osclconfig.h"
```

Defines

- #define **NULL_TERM_CHAR** '\0'
The NULL_TERM_CHAR is used to terminate c-style strings.
- #define **NULL** (0)
if the NULL macro isn't already defined, then define it as zero.
- #define **OSCL_INLINE** inline
- #define **OSCL_COND_EXPORT_REF**
- #define **OSCL_COND_IMPORT_REF**
- #define **OSCL_CONST_CAST**(type, exp) ((type)(exp))
Type casting macros.
- #define **OSCL_STATIC_CAST**(type, exp) ((type)(exp))
- #define **OSCL_REINTERPRET_CAST**(type, exp) ((type)(exp))
- #define **OSCL_DYNAMIC_CAST**(type, exp) ((type)(exp))
- #define **OSCL_UNUSED_ARG**(vbl) (void)(vbl)
- #define **OSCL_UNUSED_RETURN**(value) return value
- #define **OSCL_MIN**(a, b) ((a) < (b) ? (a) : (b))
- #define **OSCL_MAX**(a, b) ((a) > (b) ? (a) : (b))
- #define **OSCL_ABS**(a) ((a) > (0) ? (a) : -(a))
- #define **OSCL_TEMPLATED_DESTRUCTOR_CALL**(type, simple_type) type :: ~simple_type ()
- #define **OSCL_UNSIGNED_CONST**(x) x
- #define **OSCL_PACKED_VAR** "error"
- #define **OSCL_BEGIN_PACKED** "error"
- #define **OSCL_END_PACKED** "error"

7.5.1 Detailed Description

This file defines common macros and constants for basic compilation support.

7.6 oscl_bin_stream.h File Reference

Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.

```
#include "oscl_base.h"
#include "oscl_bin_stream.inl"
```

Data Structures

- class [OsclBinIStream](#)
- class [OsclBinIStreamBigEndian](#)
- class [OsclBinIStreamLittleEndian](#)
- class [OsclBinOStream](#)

Class OsclBinOStream implements the basic stream functions for an output stream.

- class [OsclBinOStreamBigEndian](#)

Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering.

- class [OsclBinOStreamLittleEndian](#)

Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering.

- class [OsclBinStream](#)

7.6.1 Detailed Description

Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.

7.7 oscl_byte_order.h File Reference

This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).

```
#include "oscl_base.h"  
#include "oscl_byte_order.inl"
```

Functions

- void [little_endian_to_host](#) (char *data, uint32 size)
Convert little endian to host format.
- void [host_to_little_endian](#) (char *data, unsigned int size)
Convert host to little endian format.
- void [big_endian_to_host](#) (char *data, unsigned int size)
Convert big endian to host format.
- void [host_to_big_endian](#) (char *data, unsigned int size)
Convert host to big endian format.

7.7.1 Detailed Description

This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).

7.8 oscl_defalloc.h File Reference

The file defines simple default memory allocator classes. These allocators are used by the [Oscl_Vector](#) and [Oscl_Map](#) class, etc.

```
#include "oscl_base.h"  
#include "osclconfig_compiler_warnings.h"  
#include "oscl_mem_inst.h"
```

Data Structures

- class [Oscl_Alloc](#)
- class [Oscl_Dealloc](#)
- class [Oscl_DefAlloc](#)
- class [Oscl_TAlloc](#)
- class [OsclAllocDestructDealloc](#)
- class [OsclDestructDealloc](#)
- struct [rebind](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [ALLOCATE\(n\)](#) [allocate_fl\(n,__FILE__,__LINE__\)](#)
- #define [ALLOC_AND_CONSTRUCT\(n\)](#) [alloc_and_construct_fl\(n,__FILE__,__LINE__\)](#)

7.8.1 Detailed Description

The file defines simple default memory allocator classes. These allocators are used by the [Oscl_Vector](#) and [Oscl_Map](#) class, etc.

7.9 oscl_dll.h File Reference

Defines a DLL entry point.

```
#include "osclconfig.h"
```

Defines

- #define **OSCL_DLL_ENTRY_POINT()** void oscl_dll_entry_point() {}
- #define **OSCL_DLL_ENTRY_POINT_DEFAULT()**

7.9.1 Detailed Description

Defines a DLL entry point.

7.10 oscl_dns.h File Reference

The file [oscl_socket.h](#) defines the OSCL DNS APIs.

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_defalloc.h"
#include "oscl_socket.h"
```

Data Structures

- class [OsclIDNS](#)
- class [OsclDNSObserver](#)

Enumerations

- enum [TPVDNSFxn](#) { [EPVDNSGetHostByName](#) }
- enum [TPVDNSEvent](#) { [EPVDNSSuccess](#), [EPVDNSPending](#), [EPVDNSTimeout](#), [EPVDNSFailure](#), [EPVDNSCancel](#) }

7.10.1 Detailed Description

The file [oscl_socket.h](#) defines the OSCL DNS APIs.

7.11 oscl_dns_gethostbyname.h File Reference

```
#include "oscl_dns_method.h"
#include "oscl_dns.h"
#include "osclconfig_io.h"
```

Data Structures

- class [OsclGetHostByNameMethod](#)
- class [OsclGetHostByNameRequest](#)

7.12 oscl_dns_imp.h File Reference

```
#include "oscl_dns_tuneables.h"  
#include "oscl_dns_imp_pv.h"
```

7.13 oscl_dns_imp_base.h File Reference

```
#include "oscl_socket_imp.h"
#include "oscl_dns_request.h"
#include "oscl_dns.h"
```

Data Structures

- class [OsclDNSIBase](#)

7.14 oscl_dns_imp_pv.h File Reference

```
#include "oscl_socket_imp_base.h"
#include "oscl_dns_request.h"
#include "oscl_dns_imp_base.h"
```

Data Structures

- class [OsclDNSI](#)

7.15 oscl_dns_method.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_dns.h"
#include "pvlogger.h"
```

Data Structures

- class [OsclIDNSMethod](#)
- class [OsclDNSRequestAO](#)

7.16 oscl_dns_param.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_dns_tuneables.h"
#include "oscl_namestring.h"
#include "oscl_dns.h"
#include "oscl_mutex.h"
#include "oscl_semaphore.h"
```

Data Structures

- class [DNSRequestParam](#)
- class [GetHostByNameParam](#)

Typedefs

- typedef [OsclMemAllocator TDNSRequestParamAllocator](#)

7.16.1 Typedef Documentation

7.16.1.1 typedef [OsclMemAllocator TDNSRequestParamAllocator](#)

7.17 oscl_dns_request.h File Reference

```
#include "oscl_dns_tuneables.h"
#include "oscl_namestring.h"
#include "oscl_dns.h"
#include "oscl_socket_types.h"
```

Data Structures

- class [OsclDNSRequest](#)

7.18 oscl_dns_tuneables.h File Reference

```
#include "osclconfig_io.h"
#include "osclconfig_proc.h"
```

Defines

- #define PV_DNS_SERVER 1
- #define PV_DNS_IS_THREAD OSCL_HAS_THREAD_SUPPORT

7.18.1 Define Documentation

7.18.1.1 #define PV_DNS_IS_THREAD OSCL_HAS_THREAD_SUPPORT

PV_DNS_IS_THREAD chooses either the threaded or AO-based implementation of the PV DNS request.
Note: AO-based option is not good here, since some DNS requests will block the caller until completion.

7.18.1.2 #define PV_DNS_SERVER 1

Enable/disable the PV DNS server here.

7.19 oscl_double_list.h File Reference

Internal use types for scheduler.

```
#include "osclconfig_proc.h"
#include "oscl_base.h"
#include "oscl_assert.h"
#include "oscl_double_list.inl"
```

Data Structures

- class [OsclDoubleLink](#)
- class [OsclDoubleList](#)
- class [OsclDoubleListBase](#)
- class [OsclDoubleRunner](#)
- class [OsclPriorityLink](#)
- class [OsclPriorityList](#)

Defines

- #define [QUE_ITER_BEGIN](#)(_type, _qname)
- #define [QUE_ITER_END](#)(_qname)

Functions

- template<class T, class S> T * [OsclPtrAdd](#) (T *aPtr, S aVal)
- template<class T, class S> T * [OsclPtrSub](#) (T *aPtr, S aVal)

7.19.1 Detailed Description

Internal use types for scheduler.

7.20 oscl_errno.h File Reference

Defines functions to access additional information on errors where supported through an errno or similar service.

```
#include "oscl_base.h"
#include "osclconfig_error.h"
#include "oscl_errno.inl"
```

Functions

- OSCL_IMPORT_REF bool [OSCL_IsErrnoSupported \(\)](#)
This function determines if a particular system saves the error number that occurs on a system call.
- OSCL_IMPORT_REF int [OSCL_GetLastError \(\)](#)
This function returns the value of the system's global error number variable.
- OSCL_IMPORT_REF bool [OSCL_SetLastError \(int newVal\)](#)
This function sets the last error code for the system.
- OSCL_IMPORT_REF char * [OSCL_StrError \(int errnum\)](#)
This function maps an error number to an error-message string.

7.20.1 Detailed Description

Defines functions to access additional information on errors where supported through an errno or similar service.

7.21 oscl_error.h File Reference

OSCL Error trap and cleanup include file.

```
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_error_codes.h"
#include "oscl_singleton.h"
#include "oscl_assert.h"
#include "oscl_tls.h"
```

Data Structures

- class [OsclError](#)
- class [OsclErrorTrap](#)
- class [OsclTLSEx](#)
- class [OsclTLSRegistryEx](#)

Defines

- #define [OSCL_TRAPSTACK_PUSH\(a\)](#) OsclError::PushL(a)
- #define [OSCL_TRAPSTACK_POP\(\)](#) OsclError::Pop()
- #define [OSCL_TRAPSTACK_POPDEALLOC\(\)](#) OsclError::PopDealloc()

7.21.1 Detailed Description

OSCL Error trap and cleanup include file.

7.22 oscl_error_allocator.h File Reference

Defines a memory allocation class used by the oscl error layer.

```
#include "oscl_base.h"  
#include "oscl_base_macros.h"  
#include "osclconfig_error.h"  
#include "oscl_assert.h"  
#include "oscl_defalloc.h"
```

Data Structures

- class [OsclErrorAllocator](#)

This class provides static methods to invoke the user defined memory allocation routines.

7.22.1 Detailed Description

Defines a memory allocation class used by the oscl error layer.

7.23 oscl_error_codes.h File Reference

Defines basic error and leave codes.

Defines

- #define `OsclErrNone` 0
- #define `OsclErrGeneral` 100
- #define `OsclErrNoMemory` 101
- #define `OsclErrCancelled` 102
- #define `OsclErrNotSupported` 103
- #define `OsclErrArgument` 104
- #define `OsclErrBadHandle` 105
- #define `OsclErrAlreadyExists` 106
- #define `OsclErrBusy` 107
- #define `OsclErrNotReady` 108
- #define `OsclErrCorrupt` 109
- #define `OsclErrTimeout` 110
- #define `OsclErrOverflow` 111
- #define `OsclErrUnderflow` 112
- #define `OsclErrInvalidState` 113
- #define `OsclErrNoResources` 114
- #define `OsclErrNotInstalled` 115
- #define `OsclErrAlreadyInstalled` 116
- #define `OsclErrSystemCallFailed` 117
- #define `OsclErrNoHandler` 118
- #define `OsclErrThreadContextIncorrect` 119
- #define `OSCL_ERR_NONE` `OsclErrNone`
- #define `OSCL_BAD_ALLOC_EXCEPTION_CODE` `OsclErrNoMemory`
- #define `OsclSuccess` 0
- #define `OsclPending` 1
- #define `OsclFailure` -1

Typedefs

- typedef int32 `OsclLeaveCode`
- typedef int32 `OsclReturnCode`

7.23.1 Detailed Description

Defines basic error and leave codes.

7.24 oscl_error_imp.h File Reference

Internal error implementation support.

```
#include "osclconfig_error.h"  
#include "oscl_error_imp_jumps.h"
```

Defines

- #define PVERROR_IMP_JUMPS

7.24.1 Detailed Description

Internal error implementation support.

7.25 oscl_error_imp_cppexceptions.h File Reference

Implementation File for Leave using C++ exceptions.

```
#include "oscl_error_trapcleanup.h"
```

Data Structures

- class [internalLeave](#)

Defines

- #define [PVError_DoLeave\(\)](#) [internalLeave](#) __ilv;__ilv.a=0;throw(__ilv)
- #define [_PV_TRAP](#)(__r, __s)
- #define [_PV_TRAP_NO_TLS](#)(__trapimp, __r, __s)

7.25.1 Detailed Description

Implementation File for Leave using C++ exceptions.

7.26 oscl_error_imp_fatalerror.h File Reference

Implementation File for Leave using system fatal error.

```
#include "oscl_assert.h"
```

Defines

- #define PVError_DoLeave() _OSCL_Abort()
- #define _PV_TRAP(__r, __s)
- #define _PV_TRAP_NO_TLS(__tr, __r, __s)

7.26.1 Detailed Description

Implementation File for Leave using system fatal error.

7.26.2 Define Documentation

7.26.2.1 #define _PV_TRAP(__r, __s)

Value:

```
__r=OsclErrNone;\  
{__s;}
```

7.26.2.2 #define _PV_TRAP_NO_TLS(__tr, __r, __s)

Value:

```
__r=OsclErrNone;\  
{__s;}
```

7.26.2.3 #define PVError_DoLeave() _OSCL_Abort()

7.27 oscl_error_imp_jumps.h File Reference

Implementation of using Setjmp / Longjmp.

```
#include "oscl_error_trapcleanup.h"
#include "oscl_assert.h"
#include "osclconfig_error.h"
#include "oscl_defalloc.h"
#include "oscl_error.h"
```

Data Structures

- class [OsclJump](#)

Defines

- #define OSCL_JUMP_MAX_JUMP_MARKS OSCL_MAX_TRAP_LEVELS
- #define internalLeave (-1)
- #define PVError_DoLeave() OsclJump::StaticJump(internalLeave)
- #define _PV_TRAP(__r, __s)
- #define _PV_TRAP_NO_TLS(__trapimp, __r, __s)

7.27.1 Detailed Description

Implementation of using Setjmp / Longjmp.

7.27.2 Define Documentation

7.27.2.1 #define _PV_TRAP(__r, __s)

Value:

```
__r=OsclErrNone; \
{ \
    OsclErrorTrapImp* __trap=OsclErrorTrapImp::Trap(); \
    if(!__trap){__s;}else{ \
        int __tr=setjmp(*(__trap->iJumpData->Top())); \
        if (__tr==0)\ 
            {__s;} \
        else if (__tr==internalLeave)\ 
            {__r=__trap->iLeave;} \
            __trap->UnTrap();} \
}
```

7.27.2.2 #define _PV_TRAP_NO_TLS(__trapimp, __r, __s)

Value:

```
__r=OsclErrNone; \
{ \
    OsclErrorTrapImp* __trap=OsclErrorTrapImp::TrapNoTls(__trapimp); \
    if(!__trap){__s;}else{ \
        int __tr=setjmp(*(__trap->iJumpData->Top())); \
        if (__tr==0) \
            {__s;} \
        else if (__tr==internalLeave) \
            {__r=__trap->iLeave;} \
        __trap->UnTrap();} \
}
```

7.27.2.3 #define PVError_DoLeave() OsclJump::StaticJump(internalLeave)

7.28 oscl_error_trapcleanup.h File Reference

OSCL Error trap and cleanup implementation include file.

```
#include "osclconfig_error.h"  
#include "oscl_heapbase.h"  
#include "oscl_defalloc.h"  
#include "oscl_assert.h"  
#include "oscl_error.h"  
#include "oscl_base_alloc.h"  
#include "oscl_tls.h"  
#include "oscl_singleton.h"  
#include "oscl_error_imp.h"
```

Data Structures

- class [OsclErrorTrapImp](#)
- class [OsclTrapStack](#)
- class [OsclTrapStackItem](#)

Defines

- #define [OSCL_MAX_TRAP_LEVELS](#) 20
- #define [PVERRORTRAP_REGISTRY_ID](#) OSCL_TLS_ID_PVERRORTRAP
- #define [PVERRORTRAP_REGISTRY](#) OsclTLSRegistry

7.28.1 Detailed Description

OSCL Error trap and cleanup implementation include file.

7.29 oscl_exception.h File Reference

contains all the exception handling macros and classes

```
#include "oscl_error.h"
#include "oscl_error_imp.h"
```

Data Structures

- class **OsclException**

oscl_exception.h contains all the exception handling macros and classes This template class provides the base exception class that all exceptions derive from

Defines

- #define **OSCL_LEAVE(_leave_status)** OsclError::Leave(_leave_status)
Use this macro to cause a Leave. It terminates the execution of the current active function.
- #define **OSCL_TRY(_leave_status, _statements)** _PV_TRAP(_leave_status,_statements)
This macro will be used to set up a try block.
- #define **OSCL_TRY_NO_TLS(_trapimp, _leave_status, _statements)** _PV_TRAP_NO_TLS(_-
 $_trapimp, _leave_status, _statements)$
• #define **OSCL_FIRST_CATCH_ANY(_leave_status, _statements)** if (_leave_status!=OsclErrNone){ _statements; }
This section defines the macros to be used in the catch block following the try block Use this macro to call a function that handles all exception types thrown in the preceding try block.
- #define **OSCL_FIRST_CATCH(_leave_status, _catch_value, _statements)** if (_leave_status!=OsclErrNone && _leave_status == _catch_value){_statements;}
Use this macro to define a block of code that catches the first exception type thrown in the preceding try block.
- #define **OSCL_CATCH(_leave_status, _catch_value, _statements)** else if (_leave_status!=OsclErrNone && _leave_status == _catch_value){_statements;}
Use this macro to define a block of code for catching additional exception types.
- #define **OSCL_CATCH_ANY(_leave_status, _statements)** else if (_leave_status!=OsclErrNone){ _-
 $_statements;$
Use this macro to call a function that will catch all remaining exception types.
- #define **OSCL_LAST_CATCH(_leave_status)** else if (_leave_status!=OsclErrNone){OSCL_-
 $LEAVE(_leave_status);$
Use this macro if OSCL_CATCH_ANY has not been used. It will mark the end of the catch block.

7.29.1 Detailed Description

contains all the exception handling macros and classes

7.30 oscl_exclusive_ptr.h File Reference

This file defines the [OsclExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

```
#include "oscl_defalloc.h"
```

Data Structures

- class [OsclExclusiveArrayPtr](#)

The OsclExclusiveArrayPtr class is a template class that defines an array pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory.

- class [OsclExclusivePtr](#)

The OsclExclusivePtr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory.

- class [OsclExclusivePtrA](#)

The OsclExclusivePtrA class is a template class that defines any pointer like object intended to be assigned an address obtained (directly or or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory.

7.30.1 Detailed Description

This file defines the [OsclExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

7.31 oscl_file_async_read.h File Reference

```
#include "oscl_base.h"
#include "osclconfig_io.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_scheduler_ao.h"
#include "oscl_file_io.h"
#include "oscl_semaphore.h"
```

Data Structures

- class [OsclAsyncFile](#)
- class [OsclAsyncFileBuffer](#)
- class [OsclBuf](#)
- class [OsclPtr](#)
- class [OsclPtrC](#)

7.32 oscl_file_cache.h File Reference

The file [oscl_file_cache.h](#) defines the class [OsclFileCache](#).

```
#include "osclconfig_io.h"  
#include "oscl_base.h"  
#include "oscl_file_io.h"
```

Data Structures

- class [OsclFileCache](#)

7.32.1 Detailed Description

The file [oscl_file_cache.h](#) defines the class [OsclFileCache](#).

7.33 oscl_file_dir_utils.h File Reference

The file `oscl_file_dir_utils.h` defines some unix-style directory ops.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
```

Data Structures

- struct `oscl_fsstat`
- struct `oscl_stat_buf`

Typedefs

- typedef `oscl_fsstat` `OSCL_FSSTAT`
- typedef `oscl_stat_buf` `OSCL_STAT_BUF`

Enumerations

- enum `OSCL_FILEMGMT_PERMS` { `OSCL_FILEMGMT_PERMS_READ` = 0x1, `OSCL_FILEMGMT_PERMS_WRITE` = 0x2, `OSCL_FILEMGMT_PERMS_EXECUTE` = 0x4 }
- enum `OSCL_FILEMGMT_MODES` { `OSCL_FILEMGMT_MODE_DIR` = 0x1 }
- enum `OSCL_FILEMGMT_ERR_TYPE` { `OSCL_FILEMGMT_E_OK` = 0, `OSCL_FILEMGMT_E_PATH_TOO_LONG`, `OSCL_FILEMGMT_E_PATH_NOT_FOUND`, `OSCL_FILEMGMT_E_ALREADY_EXISTS`, `OSCL_FILEMGMT_E_NOT_EMPTY`, `OSCL_FILEMGMT_E_PERMISSION_DENIED`, `OSCL_FILEMGMT_E_NO_MATCH`, `OSCL_FILEMGMT_E_UNKNOWN`, `OSCL_FILEMGMT_E_SYS_SPECIFIC`, `OSCL_FILEMGMT_E_NOT_IMPLEMENTED` }

Functions

- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_getcwd (oscl_wchar *path, uint32 size)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_getcwd (char *path, uint32 size)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_stat (const oscl_wchar *path, OSCL_STAT_BUF *statbuf)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_stat (const char *path, OSCL_STAT_BUF *statbuf)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_mkdir (const oscl_wchar *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_mkdir (const char *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rmdir (const oscl_wchar *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rmdir (const char *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_chdir (const oscl_wchar *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_chdir (const char *path)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rename (const oscl_wchar *oldpath, const oscl_wchar *newpath)`
- `OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_rename (const char *oldpath, const char *newpath)`

- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_statfs (OSCL_FSSTAT *stats, const char *path)
- OSCL_IMPORT_REF OSCL_FILEMGMT_ERR_TYPE oscl_statfs (OSCL_FSSTAT *stats, const oscl_wchar *path)

7.33.1 Detailed Description

The file [oscl_file_dir_utils.h](#) defines some unix-style directory ops.

7.34 oscl_file_find.h File Reference

The file [oscl_file_find.h](#) defines the class [Oscl_FileFind](#).

```
#include "osclconfig_io.h"  
#include "oscl_base.h"  
#include "oscl_mem.h"  
#include "oscl_vector.h"  
#include "oscl_string_containers.h"
```

Data Structures

- class [Oscl_FileFind](#)

7.34.1 Detailed Description

The file [oscl_file_find.h](#) defines the class [Oscl_FileFind](#).

7.35 oscl_file_handle.h File Reference

The file [oscl_file_handle.h](#) defines the class [OsclFileHandle](#).

```
#include "osclconfig_io.h"  
#include "oscl_base.h"
```

Data Structures

- class [OsclFileHandle](#)

TypeDefs

- [typedef FILE * TOsclFileHandle](#)

7.35.1 Detailed Description

The file [oscl_file_handle.h](#) defines the class [OsclFileHandle](#).

7.36 oscl_file_io.h File Reference

The file [oscl_file_io.h](#) defines the class [Oscl_File](#). This is the public API to the basic file I/O operations.

```
#include "osclconfig_io.h"  
#include "oscl_base.h"  
#include "oscl_mem.h"  
#include "oscl_file_server.h"  
#include "oscl_file_find.h"  
#include "oscl_file_dir_utils.h"  
#include "oscl_file_handle.h"
```

Data Structures

- class [Oscl_File](#)

7.36.1 Detailed Description

The file [oscl_file_io.h](#) defines the class [Oscl_File](#). This is the public API to the basic file I/O operations.

7.37 oscl_file_native.h File Reference

The file [oscl_file_native.h](#) defines the class [OsclNativeFile](#). This is the porting layer for basic file I/O operations.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_aostatus.h"
#include "oscl_file_io.h"
#include "oscl_file_types.h"
```

Data Structures

- class [OsclNativeFile](#)

7.37.1 Detailed Description

The file [oscl_file_native.h](#) defines the class [OsclNativeFile](#). This is the porting layer for basic file I/O operations.

7.38 oscl_file_server.h File Reference

The file [oscl_file_server.h](#) defines the class [Oscl_FileServer](#). This is the porting layer for file server implementations.

```
#include "osclconfig_io.h"  
#include "oscl_base.h"
```

Data Structures

- class [Oscl_FileServer](#)

7.38.1 Detailed Description

The file [oscl_file_server.h](#) defines the class [Oscl_FileServer](#). This is the porting layer for file server implementations.

7.39 oscl_file_stats.h File Reference

File stats class.

```
#include "oscl_base.h"
```

Data Structures

- class [OsclFileStats](#)
- class [OsclFileStatsItem](#)

Defines

- #define [OSCL_FILE_STATS_LOGGER_NODE](#) "OsclFileStats"

Enumerations

- 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 }

7.39.1 Detailed Description

File stats class.

7.40 oscl_file_types.h File Reference

The file [oscl_file_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.

Data Structures

- class [OsclNativeFileParams](#)

Defines

- #define [OSCL_IO_FILENAME_MAXLEN](#) 512
- #define [OSCL_IO_EXTENSION_MAXLEN](#) 512
- #define [OSCL_FILE_WCHAR_PATH_DELIMITER](#) _STRLIT("/")
- #define [OSCL_FILE_CHAR_PATH_DELIMITER](#) _STRLIT_CHAR("/")

7.40.1 Detailed Description

The file [oscl_file_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.

7.41 oscl_heapbase.h File Reference

OSCL Heap Base include file.

```
#include "osclconfig_error.h"
#include "oscl_base.h"
#include "oscl_heapbase.inl"
```

Data Structures

- class [_OsclHeapBase](#)
- class [OsclTrapItem](#)

Typedefs

- [typedef void\(* OsclTrapOperation \)\(OsclAny *\)](#)

7.41.1 Detailed Description

OSCL Heap Base include file.

7.42 oscl_init.h File Reference

Global oscl initialization.

```
#include "oscl_base.h"  
#include <stdio.h>
```

Data Structures

- class [OsclInit](#)
- class [OsclSelect](#)

7.42.1 Detailed Description

Global oscl initialization.

7.43 oscl_int64_utils.h File Reference

```
#include "oscl_base.h"
```

Data Structures

- class [Oscl_Int64_Utils](#)
The Oscl_Int64_Utils class provides a wrapper for commonly used int64/uint64 operations.
- struct [OsclInteger64Transport](#)

Typedefs

- typedef [OsclInteger64Transport _OsclInteger64Transport](#)

7.43.1 Typedef Documentation

7.43.1.1 typedef struct [OsclInteger64Transport _OsclInteger64Transport](#)

[OsclInteger64Transport](#) Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

7.44 oscl_ip_socket.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

Data Structures

- class [OsclIPSocketI](#)

7.45 oscl_linked_list.h File Reference

The file [oscl_linked_list.h](#) defines the template class [Oscl_Linked_List](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
#include "oscl_opaque_type.h"
#include "oscl_assert.h"
```

Data Structures

- class [LinkedListElement](#)
- class [Oscl_Linked_List](#)
- class [Oscl_Linked_List_Base](#)
- class [Oscl_MTLinked_List](#)

7.45.1 Detailed Description

The file [oscl_linked_list.h](#) defines the template class [Oscl_Linked_List](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

7.46 oscl_lock_base.h File Reference

This file defines an abstract lock class, [OsclLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OsclNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OsclScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.

Data Structures

- class [OsclLockBase](#)
- class [OsclNullLock](#)
- class [OsclScopedLock](#)

The OsclScopedLock class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the OsclScopedLock goes out of scope.

7.46.1 Detailed Description

This file defines an abstract lock class, [OsclLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OsclNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OsclScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.

7.47 oscl_map.h File Reference

The file `oscl_map.h` defines the template class `Oscl_Map` which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_tree.h"
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- struct `Oscl_Less`
- class `Oscl_Map`
- struct `Oscl_Select1st`
- class `value_compare`

Defines

- `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

7.47.1 Detailed Description

The file `oscl_map.h` defines the template class `Oscl_Map` which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

7.47.2 Define Documentation

7.47.2.1 `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

7.48 oscl_math.h File Reference

Provides math functions.

```
#include "osclconfig_util.h"  
#include "oscl_base.h"  
#include "oscl_math.inl"
```

Functions

- OSCL_COND_IMPORT_REF double `oscl_log` (double value)
- OSCL_COND_IMPORT_REF double `oscl_log10` (double value)
- OSCL_COND_IMPORT_REF double `oscl_sqrt` (double value)
- OSCL_COND_IMPORT_REF double `oscl_pow` (double x, double y)
- OSCL_COND_IMPORT_REF double `oscl_exp` (double value)
- OSCL_COND_IMPORT_REF double `oscl_sin` (double value)
- OSCL_COND_IMPORT_REF double `oscl_cos` (double value)
- OSCL_COND_IMPORT_REF double `oscl_tan` (double value)
- OSCL_COND_IMPORT_REF double `oscl_asin` (double value)
- OSCL_COND_IMPORT_REF double `oscl_atan` (double value)
- OSCL_COND_IMPORT_REF double `oscl_floor` (double value)

7.48.1 Detailed Description

Provides math functions.

7.49 oscl_media_data.h File Reference

Defines a container class for media data made up of a collection of memory fragments.

```
#include "oscl_base.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_media_status.h"
```

Data Structures

- class [BufferFragment](#)
- class [BufferMgr](#)
- class [BufferState](#)
- class [BuffFragGroup](#)
- class [MediaData](#)
- class [MemAllocator](#)

Typedefs

- typedef void(* [BufferFreeFuncPtr](#))(void *)
- typedef uint32 [MediaTimestamp](#)

7.49.1 Detailed Description

Defines a container class for media data made up of a collection of memory fragments.

7.50 oscl_media_status.h File Reference

Defines a status values for the [MediaData](#) containers.

Data Structures

- class [BufFragStatusClass](#)
- class [MediaStatusClass](#)

Variables

- const int32 [APPEND_MEDIA_AT_END](#) = -1

7.50.1 Detailed Description

Defines a status values for the [MediaData](#) containers.

7.51 oscl_mem.h File Reference

This file contains basic memory definitions for common use across platforms.

```
#include "osclconfig_memory.h"
#include "oscl_base.h"
#include "oscl_types.h"
#include "oscl_assert.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_lock_base.h"
#include "osclconfig_compiler_warnings.h"
#include "oscl_mem_inst.h"
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_refcounter.h"
#include "oscl_error.h"
#include "oscl_exception.h"
#include "oscl_mem.inl"
```

Data Structures

- class [HeapBase](#)
- class [OsclAuditCB](#)
- class [OsclMem](#)
- class [OsclMemAllocator](#)
- class [OsclMemAllocDestructDealloc](#)
- class [OsclMemBasicAllocator](#)
- class [OsclMemBasicAllocDestructDealloc](#)
- class [OsclMemGlobalAuditObject](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [OSCL_HAS_GLOBAL_NEW_DELETE](#) 1
- #define [OSCL_CLEANUP_BASE_CLASS\(T\)](#) _OSCL_CLEANUP_BASE_CLASS(T)
- #define [OSCL_ALLOC_NEW\(T_allocator, T, params\)](#) new(T_allocator.allocate(1)) T params
- #define [OSCL_TRAP_ALLOC_NEW\(T_ptr, T_allocator, T, params\)](#) _OSCL_TRAP_NEW(T_allocator.allocate(1),T_allocator.deallocate,T_ptr,T,params)
- #define [OSCL_ALLOC_DELETE\(ptr, T_allocator, T\)](#)
- #define [OSCL_MALLOC\(count\)](#) _oscl_default_audit_malloc(count)
- #define [oscl_malloc\(a\)](#) OSCL_MALLOC(a)
- #define [OSCL_DEFAULT_MALLOC\(x\)](#) OSCL_MALLOC(x)
- #define [OSCL_AUDIT_MALLOC\(auditCB, count\)](#) _oscl_audit_malloc(count, auditCB)
- #define [OSCL_CALLOC\(num, size\)](#) _oscl_default_audit_calloc(num,size)
- #define [oscl_calloc\(a, b\)](#) OSCL_CALLOC(a,b)

- #define **OSCL_AUDIT_CALLOC**(auditCB, num, size) _oscl_audit_calloc(num,size, auditCB)
- #define **OSCL_REALLOC**(ptr, new_size) _oscl_default_audit_realloc(ptr,new_size)
- #define **oscl_realloc**(a, b) OSCL_REALLOC(a,b)
- #define **OSCL_AUDIT_REALLOC**(auditCB, ptr, new_size) _oscl_audit_realloc(ptr,new_size, auditCB)
- #define **OSCL_FREE**(ptr) _oscl_audit_free(ptr)
- #define **oscl_free**(x) OSCL_FREE(x)
- #define **OSCL_DEFAULT_FREE**(x) OSCL_FREE(x)
- #define **OSCL_NEW**(T, params) new T params
- #define **OSCL_PLACEMENT_NEW**(ptr, constructor) new(ptr) constructor
- #define **OSCL_TRAP_NEW**(T_ptr, T, params) _OSCL_TRAP_NEW(_oscl_default_audit_new(sizeof(T)),_oscl_audit_free,T_ptr,T,params)
- #define **OSCL_AUDIT_NEW**(auditCB, T, params) new(_oscl_audit_new(sizeof(T),auditCB)) T params
- #define **OSCL_TRAP_AUDIT_NEW**(T_ptr, auditCB, T, params) _OSCL_TRAP_NEW(_oscl_audit_new(sizeof(T),auditCB),_oscl_audit_free,T_ptr,T,params)
- #define **OSCL_DELETE**(ptr)
- #define **OSCL_AUDIT_ARRAY_NEW**(auditCB, T, count) new(_oscl_audit_new(sizeof(T)*(count),auditCB)) T
- #define **OSCL_ARRAY_NEW**(T, count) new T[count]
- #define **OSCL_ARRAY_DELETE**(ptr) delete [] ptr
- #define **OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE**
- #define **_OSCL_TRAP_NEW**(exp, freeFunc, T_ptr, T, params)
- #define **_OSCL_CLEANUP_BASE_CLASS**(T) this → T::~T()

Functions

- **OSCL_COND_IMPORT_REF** **uint** **oscl_mem_aligned_size** (**uint** **size**)
- **OSCL_IMPORT_REF** **void** **OsclMemInit** (**OsclAuditCB** &auditCB)
- **OSCL_IMPORT_REF** **void *** **_oscl_audit_malloc** (**size_t**, **OsclAuditCB** &, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_audit_calloc** (**size_t**, **size_t**, **OsclAuditCB** &, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_audit_realloc** (**void** *, **size_t**, **OsclAuditCB** &, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_audit_new** (**size_t**, **OsclAuditCB** &, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_default_audit_malloc** (**size_t**, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_default_audit_calloc** (**size_t**, **size_t**, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_default_audit_realloc** (**void** *, **size_t**, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void *** **_oscl_default_audit_new** (**size_t**, **const char** *f=NULL, **const int** l=0)
- **OSCL_IMPORT_REF** **void** **_oscl_audit_free** (**void** *)
- **void *** **operator new** (**size_t** aSize, **const char** *aFile, **int** aLine)
- **void *** **operator new** (**size_t** aSize)
- **void operator delete** (**void** *aPtr)
- **void *** **operator new[]** (**size_t** aSize, **const char** *aFile, **int** aLine)
- **void *** **operator new[]** (**size_t** aSize)
- **void operator delete[]** (**void** *aPtr)

7.51.1 Detailed Description

This file contains basic memory definitions for common use across platforms.

This is the main entry point header file for the OSCL memory library. It should be the only one users directly include. Basic memory copy, compare, and move functions are defined here as well as the allocation functions.

7.51.2 Define Documentation

7.51.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

Previously this was in oscl_mem_imp.h

7.51.3 Function Documentation

7.51.3.1 void operator delete (void * *aPtr*) [inline]

7.51.3.2 void* operator new (size_t *aSize*) [inline]

7.52 oscl_mem_align.h File Reference

7.53 oscl_mem_audit.h File Reference

This file contains the definition and partial implementation of MM_Audit class.

```
#include "oscl_lock_base.h"
#include "oscl_base_alloc.h"
#include "oscl_tagtree.h"
#include "oscl_mem.h"
#include "oscl_mem_auto_ptr.h"
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- struct [MM_AllocInfo](#)
- struct [MM_AllocNode](#)
- struct [MM_AllocQueryInfo](#)
- class [MM_Audit_Imp](#)
- struct [MM_AuditOverheadStats](#)
- struct [MM_FailInsertParam](#)
- struct [MM_Stats_CB](#)
- struct [MM_Stats_t](#)
- class [OsclMemAudit](#)
- class [OsclMemStatsNode](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [MM_ALLOC_MAX_QUERY_FILENAME_LEN](#) 128
- #define [MM_ALLOC_MAX_QUERY_TAG_LEN](#) 64
- #define [MM_AUDIT_VALIDATE_BLOCK](#) 1
- #define [MM_AUDIT_PREFILL_FLAG](#) 0x1
- #define [MM_AUDIT_POSTFILL_FLAG](#) 0x2
- #define [MM_AUDIT_VALIDATE_ALL_HEAP_FLAG](#) 0x4
- #define [MM_AUDIT_VALIDATE_ON_FREE_FLAG](#) 0x8
- #define [MM_AUDIT_ALLOC_NODE_ENABLE_FLAG](#) 0x10
- #define [MM_AUDIT_SUPPRESS_FILENAME_FLAG](#) 0x20
- #define [DEFAULT_MM_AUDIT_MODE](#) 0

Typedefs

- typedef [OSCLMemAutoPtr< char, Oscl_TAlloc< char, OsclMemBasicAllocator > >](#) [MMAudit_CharAutoPtr](#)
- typedef [OSCLMemAutoPtr< uint8, Oscl_TAlloc< uint8, _OsclBasicAllocator > >](#) [MMAudit_Uint8AutoPtr](#)
- typedef [OSCLMemAutoPtr< MM_AllocNode, Oscl_TAlloc< MM_AllocNode, OsclMemBasicAllocator > >](#) [MM_AllocNodeAutoPtr](#)
- typedef [OSCLMemAutoPtr< OsclMemStatsNode, Oscl_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > >](#) [MM_StatsNodeTagTreeType](#)

- `typedef OSCLMemAutoPtr< OsclMemStatsNode, Oscl_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > > OsclMemStatsNodeAutoPtr`
- `typedef Oscl_TAlloc< MM_StatsNodeTagTreeType, OsclMemBasicAllocator > TagTree_Allocator`
- `typedef Oscl_TagTree< MM_StatsNodeTagTreeType, TagTree_Allocator > OsclTagTreeType`

7.53.1 Detailed Description

This file contains the definition and partial implementation of MM_Audit class.

7.53.2 Define Documentation

7.53.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

7.54 oscl_mem_audit_internals.h File Reference

This file contains the internal definitions for the mem audit library.

```
#include "oscl_base.h"
#include "oscl_mem_audit.h"
#include "oscl_mem_inst.h"
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- struct [MM_AllocBlockFence](#)
- struct [MM_AllocBlockHdr](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [MM_AUDIT_ALLOC_NODE_SUPPORT](#) 1
- #define [MM_AUDIT_FENCE_SUPPORT](#) 0
- #define [MM_AUDIT_INCLUDE_ALL_HEAP_VALIDATION](#) 1
- #define [MM_AUDIT_FILL_SUPPORT](#) 0
- #define [MM_AUDIT_FAILURE_SIMULATION_SUPPORT](#) 1
- #define [FENCE_PATTERN](#) 0xAA
- #define [MIN_FENCE_SIZE](#) 4
- #define [MEM_ALIGN_SIZE](#) 8
- #define [COMPUTE_MEM_ALIGN_SIZE](#)(x, y, z) (y+(((x+y)%z) ? (z - (x+y)%z) : 0))
- #define [DEFAULT_PREFILL_PATTERN](#) 0x96
- #define [DEFAULT_POSTFILL_PATTERN](#) 0x5A

7.54.1 Detailed Description

This file contains the internal definitions for the mem audit library.

7.54.2 Define Documentation

7.54.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

7.55 oscl_mem_auto_ptr.h File Reference

This file defines the oscl_mem_auto_ptr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

```
#include "osclconfig_memory.h"  
#include "osclconfig_compiler_warnings.h"  
#include "oscl_mem.h"
```

Data Structures

- class [OSCLMemAutoPtr](#)

The oscl_auto_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the oscl_auto_ptr expires, its destructor uses delete to free the memory.

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)
- #define [OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT](#)

7.55.1 Detailed Description

This file defines the oscl_mem_auto_ptr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

7.55.2 Define Documentation

7.55.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

7.56 oscl_mem_basic_functions.h File Reference

This file contains prototypes for the basic memory functions.

```
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.inl"
```

Functions

- OSCL_COND_IMPORT_REF void * [_oscl_malloc](#) (int32 count)
- OSCL_COND_IMPORT_REF void * [_oscl_calloc](#) (int32 nelems, int32 size)
- OSCL_COND_IMPORT_REF void * [_oscl_realloc](#) (void *src, int32 count)
- OSCL_COND_IMPORT_REF void [_oscl_free](#) (void *src)
- OSCL_COND_IMPORT_REF void * [oscl_memcpy](#) (void *dest, const void *src, uint32 count)
- OSCL_COND_IMPORT_REF void * [oscl_memmove](#) (void *dest, const void *src, uint32 count)
- OSCL_COND_IMPORT_REF void * [oscl_memmove32](#) (void *dest, const void *src, uint32 count)
- OSCL_COND_IMPORT_REF void * [oscl_memset](#) (void *dest, uint8 val, uint32 count)
- OSCL_COND_IMPORT_REF int [oscl_memcmp](#) (const void *buf1, const void *buf2, uint32 count)

7.56.1 Detailed Description

This file contains prototypes for the basic memory functions.

7.57 oscl_mem_inst.h File Reference

The file defines default memory instrumentation level.

```
#include "osclconfig_memory.h"
```

Defines

- #define **PVMEM_INST_LEVEL** 1

7.57.1 Detailed Description

The file defines default memory instrumentation level.

7.58 oscl_mem_mempool.h File Reference

This file contains the definition of memory pool allocators.

```
#include "oscl_mem.h"  
#include "oscl_defalloc.h"  
#include "oscl_vector.h"
```

Data Structures

- struct [MemPoolBlockInfo](#)
- struct [MemPoolBufferInfo](#)
- class [OsclMemPoolFixedChunkAllocator](#)
- class [OsclMemPoolFixedChunkAllocatorObserver](#)
- class [OsclMemPoolResizableAllocator](#)
- class [OsclMemPoolResizableAllocatorMemoryObserver](#)
- class [OsclMemPoolResizableAllocatorObserver](#)

7.58.1 Detailed Description

This file contains the definition of memory pool allocators.

7.59 oscl_mempool_allocator.h File Reference

This file contains the definition of memory pool allocator for leave/trap.

```
#include "oscl_defalloc.h"
```

Data Structures

- class [OsclMemPoolAllocator](#)

7.59.1 Detailed Description

This file contains the definition of memory pool allocator for leave/trap.

7.60 oscl_mutex.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"  
#include "oscl_types.h"  
#include "oscl_base.h"  
#include "oscl_thread.h"  
#include "oscl_lock_base.h"
```

Data Structures

- class [OsclMutex](#)
- class [OsclThreadLock](#)

Typedefs

- typedef [OsclMutex OsclNoYieldMutex](#)

7.60.1 Detailed Description

This file provides implementation of mutex.

7.60.2 Typedef Documentation

7.60.2.1 typedef [OsclMutex OsclNoYieldMutex](#)

Class [OsclNoYieldMutex](#) can be used in use cases where there will be no CPU-yielding operation done while the Mutex is locked.

CPU-yielding operations include [OsclMutex::Lock](#), [OsclSemaphore::Wait](#), [OsclThread::Sleep](#), and [OsclBrewThreadUtil::BThreadYield](#).

The behavior of [OsclNoYieldMutex](#) depends on whether the threading model is pre-emptive or not. When threading is pre-emptive, it is identical to [OsclMutex](#). When threading is non-pre-emptive, it is a NO-OP.

An example of this type of use case is for simple data protection.

7.61 oscl_namestring.h File Reference

Name string class include file.

```
#include "oscl_base.h"
```

Data Structures

- class [OsclNameString](#)

7.61.1 Detailed Description

Name string class include file.

7.62 oscl_opaque_type.h File Reference

The file [oscl_opaque_type.h](#) defines pure virtual classes for working with opaque types.

```
#include "oscl_base.h"
```

Data Structures

- class [Oscl_Opaque_Type_Alloc](#)
- class [Oscl_Opaque_Type_Alloc_LL](#)
- class [Oscl_Opaque_Type_Compare](#)

7.62.1 Detailed Description

The file [oscl_opaque_type.h](#) defines pure virtual classes for working with opaque types.

7.63 oscl_priqueue.h File Reference

Implements a priority queue data structure similar to STL.

```
#include "oscl_base.h"  
#include "oscl_vector.h"
```

Data Structures

- class [OsclCompareLess](#)
- class [OsclPriorityQueue](#)
- class [OsclPriorityQueueBase](#)

7.63.1 Detailed Description

Implements a priority queue data structure similar to STL.

Implements a priority queue data structure similar to the STL class. The properties of the class include O(Log_2(N)) insertion and deletion complexity and O(1) complexity to access the top priority item.

7.64 oscl_procstatus.h File Reference

Data Structures

- class [OsclProcStatus](#)

7.65 oscl_queue.h File Reference

The file [oscl_queue.h](#) defines the template class [Oscl_Queue](#). It is similar to the STL::queue class, with some differences: - less complete - based on array rather than a deque - some interfaces modeled on oscl_vector, for ease of transition Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.h"  
#include "oscl_assert.h"  
#include "oscl_opaque_type.h"
```

Data Structures

- class [Oscl_Queue](#)
- class [Oscl_Queue_Base](#)

7.65.1 Detailed Description

The file [oscl_queue.h](#) defines the template class [Oscl_Queue](#). It is similar to the STL::queue class, with some differences: - less complete - based on array rather than a deque - some interfaces modeled on oscl_vector, for ease of transition Memory allocation is abstracted through the use of an allocator template parameter.

7.66 oscl_rand.h File Reference

Provides pseudo-random number generation.

```
#include "osclconfig_util.h"  
#include "oscl_base.h"  
#include "oscl_mem_basic_functions.h"  
#include "oscl_rand.inl"
```

Data Structures

- class [OsclRand](#)

7.66.1 Detailed Description

Provides pseudo-random number generation.

7.67 oscl_refcounter.h File Reference

A general purpose reference counter to object lifetimes.

```
#include "oscl_assert.h"  
#include "oscl_defalloc.h"
```

Data Structures

- class [Oscl_DefAllocWithRefCounter](#)
- class [OsclRefCounter](#)
- class [OsclRefCounterDA](#)
- class [OsclRefCounterMTDA](#)
- class [OsclRefCounterMTSA](#)
- class [OsclRefCounterSA](#)

7.67.1 Detailed Description

A general purpose reference counter to object lifetimes.

7.68 oscl_refcounter_memfrag.h File Reference

This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount.

```
#include "oscl_base.h"  
#include "oscl_refcounter.h"
```

Data Structures

- class [OsclRefCounterMemFrag](#)

7.68.1 Detailed Description

This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount.

7.69 oscl_registry_access_client.h File Reference

Client-side implementation Registry Access implementation.

```
#include "oscl_registry_types.h"
#include "oscl_string_containers.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

Data Structures

- class [OsclRegistryAccessClient](#)

7.69.1 Detailed Description

Client-side implementation Registry Access implementation.

7.70 oscl_registry_client.h File Reference

Client-side implementation of OsclRegistry.

```
#include "oscl_registry_types.h"
#include "oscl_mem.h"
#include "oscl_string.h"
```

Data Structures

- class [OsclRegistryClient](#)

7.70.1 Detailed Description

Client-side implementation of OsclRegistry.

7.71 oscl_registry_client_impl.h File Reference

Client-side implementation of OsclRegistryInterface.

```
#include "oscl_base.h"
#include "osclconfig_proc.h"
#include "oscl_vector.h"
#include "oscl_string.h"
#include "oscl_registry_types.h"
#include "oscl_registry_serv_impl_tls.h"
```

Data Structures

- class [OsclRegistryAccessClientImpl](#)
- class [OsclRegistryAccessClientTlsImpl](#)
- class [OsclRegistryClientImpl](#)
- class [OsclRegistryClientTlsImpl](#)

7.71.1 Detailed Description

Client-side implementation of OsclRegistryInterface.

7.72 oscl_registry_serv_impl.h File Reference

Server-side implementation of OsclRegistry interfaces.

```
#include "oscl_base.h"
#include "osclconfig_proc.h"
#include "oscl_registry_types.h"
#include "oscl_string.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
```

Data Structures

- class [OsclComponentRegistry](#)
- class [OsclComponentRegistryData](#)
- class [OsclComponentRegistryElement](#)

7.72.1 Detailed Description

Server-side implementation of OsclRegistry interfaces.

7.73 oscl_registry_serv_impl_global.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_base.h"
```

7.74 oscl_registry_serv_impl_tls.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_registry_serv_impl.h"
#include "oscl_registry_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

Data Structures

- class [OsclRegistryServTlsImpl](#)

7.75 oscl_registry_types.h File Reference

Common types used in Oscl registry interfaces.

```
#include "oscl_types.h"  
#include "oscl_string_containers.h"
```

Data Structures

- class [OsclRegistryAccessElement](#)

TypeDefs

- typedef [OsclAny](#) * [OsclComponentFactory](#)

7.75.1 Detailed Description

Common types used in Oscl registry interfaces.

7.76 oscl_scheduler.h File Reference

```
#include "oscl_scheduler_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_defalloc.h"
#include "oscl_mem.h"
```

Data Structures

- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclScheduler](#)
- class [OsclSchedulerObserver](#)
- class [PVSchedulerStopper](#)

Defines

- #define [PVSCHEDNAMELEN](#) 30

7.77 oscl_scheduler_ao.h File Reference

Oscl Scheduler user execution object classes.

```
#include "oscl_scheduler_aobase.h"
#include "oscl_mem.h"
#include "oscl_scheduler_types.h"
```

Data Structures

- class [OsclActiveObject](#)
- class [OsclTimerObject](#)

7.77.1 Detailed Description

Oscl Scheduler user execution object classes.

7.78 oscl_scheduler_aobase.h File Reference

Oscl Scheduler internal active object classes.

```
#include "oscl_namestring.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_scheduler_readyq.h"
#include "oscl_string_containers.h"
#include "oscl_scheduler_types.h"
```

Data Structures

- class [PVActiveBase](#)
- class [PVActiveStats](#)

Defines

- #define [OSCL_ZEROIZE](#)(ptr, size) oscl_memset(ptr, 0, size)
- #define [PVEEXECNAMELEN](#) 30

7.78.1 Detailed Description

Oscl Scheduler internal active object classes.

7.79 oscl_scheduler_readyq.h File Reference

ready q types for oscl scheduler

```
#include "oscl_scheduler_tuneables.h"
#include "oscl_priqueue.h"
#include "oscl_base_alloc.h"
#include "oscl_semaphore.h"
#include "oscl_mem.h"
#include "oscl_string_containers.h"
#include "oscl_scheduler_types.h"
#include "oscl_mutex.h"
```

Data Structures

- class [OsclReadyAlloc](#)
- class [OsclReadyCompare](#)
- class [OsclReadyQ](#)
- class [OsclTimerCompare](#)
- class [OsclTimerQ](#)
- class [TReadyQueLink](#)

Typedefs

- typedef [PVActiveBase](#) * TOsclReady

7.79.1 Detailed Description

ready q types for oscl scheduler

7.80 oscl_scheduler_threadcontext.h File Reference

Thread context functions needed by oscl scheduler.

```
#include "oscl_double_list.h"  
#include "oscl_mutex.h"  
#include "oscl_aostatus.h"
```

Data Structures

- class [PVThreadContext](#)

Enumerations

- enum [TPVThreadContext](#) { [EPVThreadContext_InThread](#), [EPVThreadContext_OsclThread](#), [EPVThreadContext_NonOsclThread](#), [EPVThreadContext_Undetermined](#) }

7.80.1 Detailed Description

Thread context functions needed by oscl scheduler.

7.81 oscl_scheduler_tuneables.h File Reference

Tunable settings for Oscl Scheduler.

```
#include "osclconfig_proc.h"
```

Defines

- #define PV_SCHED_ENABLE_AO_STATS 1
- #define PV_SCHED_ENABLE_LOOP_STATS 0
- #define PV_SCHED_ENABLE_PERF_LOGGING 1
- #define PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS 1
- #define PV_SCHED_LOG_Q 0
- #define PV_SCHED_CHECK_Q 0
- #define PV_SCHED_FAIR_SCHEDULING 1
- #define OSCL_PERF_SUMMARY_LOGGING 0

7.81.1 Detailed Description

Tunable settings for Oscl Scheduler.

7.82 oscl_scheduler_types.h File Reference

Scheduler common types include file.

```
#include "osclconfig_proc.h"  
#include "oscl_aostatus.h"  
#include "oscl_heapbase.h"
```

Data Structures

- class [OsclExecSchedulerBase](#)

7.82.1 Detailed Description

Scheduler common types include file.

7.83 oscl_semaphore.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"  
#include "oscl_thread.h"
```

Data Structures

- class [OsclSemaphore](#)

7.83.1 Detailed Description

This file provides implementation of mutex.

7.84 oscl_shared_ptr.h File Reference

This file defines a template class [OsclSharedPtr](#) which is a "smart pointer" to the parameterized type.

```
#include "oscl_base.h"
#include "oscl_refcounter.h"
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- class [OsclSharedPtr](#)
A parameterized smart pointer class.

Defines

- #define [OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT](#)

7.84.1 Detailed Description

This file defines a template class [OsclSharedPtr](#) which is a "smart pointer" to the parameterized type.

7.85 oscl_singleton.h File Reference

This file defines the [OsclSingleton](#) class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

Data Structures

- class [OsclSingleton](#)
- class [OsclSingletonRegistry](#)
- class [SingletonTable](#)

Variables

- const uint32 [OSCL_SINGLETON_ID_TEST](#) = 0
- const uint32 [OSCL_SINGLETON_ID_OSCLMEM](#) = 1
- const uint32 [OSCL_SINGLETON_ID_PVLOGGER](#) = 2
- const uint32 [OSCL_SINGLETON_ID_PVSCHEDULER](#) = 3
- const uint32 [OSCL_SINGLETON_ID_PVERRORTRAP](#) = 4
- const uint32 [OSCL_SINGLETON_ID_SDPMEDIAPARSER](#) = 5
- const uint32 [OSCL_SINGLETON_ID_PAYLOADPARSER](#) = 6
- const uint32 [OSCL_SINGLETON_ID_CPM_PLUGIN](#) = 7
- const uint32 [OSCL_SINGLETON_ID_PVMFRECOGNIZER](#) = 8
- const uint32 [OSCL_SINGLETON_ID_OSCLREGISTRY](#) = 9
- const uint32 [OSCL_SINGLETON_ID_OMX](#) = 10
- const uint32 [OSCL_SINGLETON_ID_OMXMASTERCORE](#) = 11
- const uint32 [OSCL_SINGLETON_ID_TICKCOUNT](#) = 12
- const uint32 [OSCL_SINGLETON_ID_LAST](#) = 13

7.85.1 Detailed Description

This file defines the [OsclSingleton](#) class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

[OsclSingleton](#) is initialized in [OsclBase::Init](#).

7.85.2 Variable Documentation

- 7.85.2.1 `const uint32 OSCL_SINGLETON_ID_CPM_PLUGIN = 7`
- 7.85.2.2 `const uint32 OSCL_SINGLETON_ID_LAST = 13`
- 7.85.2.3 `const uint32 OSCL_SINGLETON_ID_OMX = 10`
- 7.85.2.4 `const uint32 OSCL_SINGLETON_ID_OMXMASTERCORE = 11`
- 7.85.2.5 `const uint32 OSCL_SINGLETON_ID_OSCLMEM = 1`
- 7.85.2.6 `const uint32 OSCL_SINGLETON_ID_OSCLREGISTRY = 9`
- 7.85.2.7 `const uint32 OSCL_SINGLETON_ID_PAYLOADPARSER = 6`
- 7.85.2.8 `const uint32 OSCL_SINGLETON_ID_PVERRORTRAP = 4`
- 7.85.2.9 `const uint32 OSCL_SINGLETON_ID_PVLOGGER = 2`
- 7.85.2.10 `const uint32 OSCL_SINGLETON_ID_PVMFRECOGNIZER = 8`
- 7.85.2.11 `const uint32 OSCL_SINGLETON_ID_PVSCHEDULER = 3`
- 7.85.2.12 `const uint32 OSCL_SINGLETON_ID_SDPMEDIAPARSER = 5`
- 7.85.2.13 `const uint32 OSCL_SINGLETON_ID_TEST = 0`
- 7.85.2.14 `const uint32 OSCL_SINGLETON_ID_TICKCOUNT = 12`

7.86 oscl_snprintf.h File Reference

Provides a portable implementation of snprintf.

```
#include "oscl_base.h"  
#include "osclconfig_util.h"
```

Functions

- OSCL_IMPORT_REF int32 [oscl_snprintf](#) (char *str, uint32 count, const char *fmt,...)
- OSCL_IMPORT_REF int32 [oscl_snprintf](#) ([oscl_wchar](#) *str, uint32 count, const [oscl_wchar](#) *fmt,...)
- OSCL_IMPORT_REF int32 [oscl_vsnprintf](#) (char *str, uint32 count, const char *fmt, va_list args)
- OSCL_IMPORT_REF int32 [oscl_vsnprintf](#) ([oscl_wchar](#) *str, uint32 count, const [oscl_wchar](#) *fmt, va_list args)

7.86.1 Detailed Description

Provides a portable implementation of snprintf.

7.87 oscl_socket.h File Reference

The file [oscl_socket.h](#) defines the OSCL Socket APIs.

```
#include "osclconfig_io.h"  
#include "oscl_heapbase.h"  
#include "oscl_defalloc.h"  
#include "oscl_vector.h"  
#include "oscl_mem.h"  
#include "oscl_socket_types.h"
```

Data Structures

- class [OsclSocketServ](#)
- class [OsclTCPSocket](#)
- class [OsclUDPSocket](#)

7.87.1 Detailed Description

The file [oscl_socket.h](#) defines the OSCL Socket APIs.

7.88 oscl_socket_accept.h File Reference

```
#include "oscl_socket_imp.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclAcceptMethod](#)
- class [OsclAcceptRequest](#)

7.89 oscl_socket_bind.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclBindMethod](#)
- class [OsclBindRequest](#)

7.90 oscl_socket_connect.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclConnectMethod](#)
- class [OsclConnectRequest](#)

7.91 oscl_socket_imp.h File Reference

```
#include "oscl_socket_tuneables.h"  
#include "oscl_socket_imp_pv.h"
```

7.92 oscl_socket_imp_base.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_request.h"
#include "oscl_defalloc.h"
#include "oscl_mutex.h"
#include "oscl_socket_stats.h"
#include "oscl_base.h"
```

Data Structures

- class [OsclSocketIBase](#)

7.93 oscl_socket_imp_pv.h File Reference

```
#include "oscl_socket_imp_base.h"
```

Data Structures

- class [OsclSocketI](#)

Defines

- #define [PVSOCK_ERR_BAD_PARAM](#) (-1)
- #define [PVSOCK_ERR SOCK_NOT_OPEN](#) (-2)
- #define [PVSOCK_ERR SOCK_NO_SERV](#) (-3)
- #define [PVSOCK_ERR SERV_NOT_CONNECTED](#) (-4)
- #define [PVSOCK_ERR SOCK_NOT_CONNECTED](#) (-5)
- #define [PVSOCK_ERR NOT_IMPLEMENTED](#) (-6)

7.93.1 Define Documentation

7.93.1.1 #define PVSOCK_ERR_BAD_PARAM (-1)

some error codes for request completion these are negative so they won't conflict with errors from the OS socket layer.

7.93.1.2 #define PVSOCK_ERR_NOT_IMPLEMENTED (-6)

7.93.1.3 #define PVSOCK_ERR_SERV_NOT_CONNECTED (-4)

7.93.1.4 #define PVSOCK_ERR SOCK_NO_SERV (-3)

7.93.1.5 #define PVSOCK_ERR SOCK_NOT_CONNECTED (-5)

7.93.1.6 #define PVSOCK_ERR SOCK_NOT_OPEN (-2)

7.94 oscl_socket_listen.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclListenMethod](#)
- class [OsclListenRequest](#)

Defines

- #define [OSCL_SOCKET_LISTEN_H_INCLUDEDd](#)

7.94.1 Define Documentation

7.94.1.1 #define OSCL_SOCKET_LISTEN_H_INCLUDEDd

7.95 oscl_socket_method.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_socket_request.h"
#include "pvlogger.h"
#include "oscl_socket_tuneables.h"
#include "oscl_ip_socket.h"
#include "oscl_socket_imp.h"
```

Data Structures

- class [OsclSocketMethod](#)
- class [OsclSocketRequestAO](#)

Defines

- #define [MSEC_TO_MICROSEC](#) 1000

7.95.1 Define Documentation

7.95.1.1 #define MSEC_TO_MICROSEC 1000

7.96 oscl_socket_recv.h File Reference

```
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclRecvMethod](#)
- class [OsclRecvRequest](#)

7.97 oscl_socket_recv_from.h File Reference

```
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclRecvFromMethod](#)
- class [OsclRecvFromRequest](#)

7.98 oscl_socket_request.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_tuneables.h"
```

Data Structures

- class [AcceptParam](#)
- class [BindParam](#)
- class [ConnectParam](#)
- class [ListenParam](#)
- class [OsclSocketRequest](#)
- class [PVSockBufRecv](#)
- class [PVSockBufSend](#)
- class [RecvFromParam](#)
- class [RecvParam](#)
- class [SendParam](#)
- class [SendToParam](#)
- class [ShutdownParam](#)
- class [SocketRequestParam](#)

7.99 oscl_socket_send.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclSendMethod](#)
- class [OsclSendRequest](#)

7.100 oscl_socket_send_to.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclSendToMethod](#)
- class [OsclSendToRequest](#)

7.101 oscl_socket_serv_imp.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_tuneables.h"
#include "oscl_socket_serv_imp_pv.h"
```

7.102 oscl_socket_serv_imp_base.h File Reference

```
#include "oscl_base.h"
#include "oscl_socket_stats.h"
```

Data Structures

- class [OsclSocketServIBase](#)

7.103 oscl_socket_serv_imp_pv.h File Reference

```
#include "oscl_socket_serv_imp_base.h"
#include "oscl_socket_serv_imp_reqlist.h"
#include "oscl_socket_tuneables.h"
#include "oscl_scheduler_ao.h"
```

Data Structures

- class [OsclSocketServI](#)

Defines

- #define [OSCL_READSET_FLAG](#) 0x04
- #define [OSCL_WRITESET_FLAG](#) 0x02
- #define [OSCL_EXCEPTSET_FLAG](#) 0x01

7.103.1 Define Documentation

7.103.1.1 #define OSCL_EXCEPTSET_FLAG 0x01

7.103.1.2 #define OSCL_READSET_FLAG 0x04

A bitmask for socket select operations

7.103.1.3 #define OSCL_WRITESET_FLAG 0x02

7.104 oscl_socket_serv_imp_reqlist.h File Reference

```
#include "oscl_socket_tuneables.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

Data Structures

- class [OsclSocketServRequestList](#)
- class [OsclSocketServRequestQElem](#)

7.105 oscl_socket_shutdown.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_method.h"
```

Data Structures

- class [OsclShutdownMethod](#)
- class [OsclShutdownRequest](#)

7.106 oscl_socket_stats.h File Reference

```
#include "oscl_base.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
#include "oscl_socket_tuneables.h"
```

Enumerations

- enum TOsclSocketStatEvent { EOscSocket_RequestAO_Success, EOscSocket_RequestAO_Canceled, EOscSocket_RequestAO_Error, EOscSocket_RequestAO_Timeout, EOscSocket_ServRequestIssued, EOscSocket_ServPoll, EOscSocket_OS, EOscSocket_Readable, EOscSocket_Writable, EOscSocket_Except, EOscSocket_DataRecv, EOscSocket_DataSent, EOscSocket_ServRequestComplete, EOscSocket_ServRequestCancelIssued, EOscSocketServ_LoopsockOk, EOscSocketServ_LoopsockError }
- enum TOsclSocketServStatEvent { EOscSocketServ_SelectNoActivity = 0, EOscSocketServ_SelectActivity, EOscSocketServ_SelectRescheduleAsap, EOscSocketServ_SelectReschedulePoll, EOscSocketServ_LastEvent }

7.106.1 Enumeration Type Documentation

7.106.1.1 enum TOsclSocketServStatEvent

Enumeration values:

EOscSocketServ_SelectNoActivity
EOscSocketServ_SelectActivity
EOscSocketServ_SelectRescheduleAsap
EOscSocketServ_SelectReschedulePoll
EOscSocketServ_LastEvent

7.106.1.2 enum TOsclSocketStatEvent

Socket diagnostics.

Enumeration values:

EOscSocket_RequestAO_Success
EOscSocket_RequestAO_Canceled
EOscSocket_RequestAO_Error
EOscSocket_RequestAO_Timeout
EOscSocket_ServRequestIssued
EOscSocket_ServPoll
EOscSocket_OS
EOscSocket_Readable
EOscSocket_Writable

EOselSocket_Except
EOselSocket_DataRecv
EOselSocket_DataSent
EOselSocket_ServRequestComplete
EOselSocket_ServRequestCancelIssued
EOselSocketServ_LoopsockOk
EOselSocketServ_LoopsockError

7.107 oscl_socket_tuneables.h File Reference

```
#include "osclconfig_io.h"
#include "osclconfig_proc.h"
```

Defines

- #define PV_SOCKET_REQUEST_AO_PRIORITY OsclActiveObject::EPriorityNominal
- #define PV_OSCL_SOCKET_STATS_LOGGING 0
- #define PV_SOCKET_SERVER 1
- #define PV_SOCKET_SERVER_IS_THREAD OSCL_HAS_THREAD_SUPPORT
- #define PV_SOCKET_SERVER_SELECT 0
- #define PV_SOCKET_SERVER_THREAD_PRIORITY ThreadPriorityAboveNormal
- #define PV_SOCKET_SERVER_SELECT_TIMEOUT_MSEC (-1)
- #define PV_SOCKET_SERVER_SELECT_LOOPBACK_SOCKET 0
- #define PV_SOCKET_SERVER_AO_PRIORITY (OsclActiveObject::EPriorityNominal)
- #define PV_SOCKET_SERVER_AO_INTERVAL_MSEC 5
- #define PV_OSCL_SOCKET_SERVER_LOGGER_OUTPUT 0
- #define PV_OSCL_SOCKET_1MB_RECV_BUF 0
- #define PV_SOCKET_SERVI_STATS 0

7.107.1 Define Documentation

7.107.1.1 #define PV_OSCL_SOCKET_1MB_RECV_BUF 0

Set this to 0 or 1 to enable/disable setting the socket receive buffer size to 1 MB in the Bind call. This setting only affects PV socket server implementations.

When set to 1, the code will use the OsclSetRecvBufferSize macro to set the buffer size in the Bind call.

This setting was found to improve streaming performance on WinMobile devices, but should not generally be used.

7.107.1.2 #define PV_OSCL_SOCKET_SERVER_LOGGER_OUTPUT 0

Set this to 0 or 1 to enable/disable [PVLogger](#) output from PV socket server. Note that socket server logging will appear in a different file when running threaded mode of socket server. This is quite a bit of logging, so it should generally be disabled.

7.107.1.3 #define PV_OSCL_SOCKET_STATS_LOGGING 0

Set this to 0 or 1 to enable/disable socket stats logging with "OsclSocketStats" node. This feature is fairly costly so should be off in production code.

7.107.1.4 #define PV_SOCKET_REQUEST_AO_PRIORITY OsclActiveObject::EPriority-Nominal

PV_SOCKET_REQUEST_AO_PRIORITY sets the priority of the socket request completion AOs.

7.107.1.5 #define PV_SOCKET_SERVER 1

Enable/disable the PV socket server here.

7.107.1.6 #define PV_SOCKET_SERVER_AO_INTERVAL_MSEC 5

PV_SOCKET_SERVER_AO_INTERVAL_MSEC sets the AO scheduling interval of the PV socket server AO for non-threaded implementations.

7.107.1.7 #define PV_SOCKET_SERVER_AO_PRIORITY (OsclActiveObject::EPriority-Nominal)

PV_SOCKET_SERVER_AO_PRIORITY sets priority of the PV socket server AO for non-threaded implementations.

7.107.1.8 #define PV_SOCKET_SERVER_IS_THREAD OSCL_HAS_THREAD_SUPPORT

PV_SOCKET_SERVER_IS_THREAD chooses either the threaded or AO-based implementation of the PV socket server

7.107.1.9 #define PV_SOCKET_SERVER_SELECT 0

PV_SOCKET_SERVER_SELECT chooses whether to use "select" call or not. In threaded mode, select call is required and is forced to "1". In AO mode, "select" call is an option that defaults to "0". Avoiding any "select" call was found to greatly reduce CPU usage on WinMobile devices.

7.107.1.10 #define PV_SOCKET_SERVER_SELECT_LOOPBACK_SOCKET 0

PV_SOCKET_SERVER_SELECT_LOOPBACK_SOCKET enables the feature to wakeup the select call by writing to a loopback socket each time a new request comes in. This option is required to support the blocking select option of threaded server mode. This option is forced to "0" in AO mode.

7.107.1.11 #define PV_SOCKET_SERVER_SELECT_TIMEOUT_MSEC (-1)

PV_SOCKET_SERVER_SELECT_TIMEOUT_MSEC sets duration of the select call in the PV socket server thread for the polling select loop implementation. When the timeout is -1, the select call will block forever waiting on a new request and will use a loopback socket to signal a new request. Note: if infinite wait is selected, but loopback socket is not available, the implementation will poll at 10 msec intervals.

7.107.1.12 #define PV_SOCKET_SERVER_THREAD_PRIORITY ThreadPriorityAboveNormal

PV_SOCKET_SERVER_THREAD_PRIORITY sets the priority of the PV socket server thread.

7.107.1.13 #define PV_SOCKET_SERVI_STATS 0

For detailed performance breakdown of time spend in [OsclSocketServI](#) AO. Output is logged under "OsclSchedulerPerfStats" node. Should be off in production code. This option is forced to "0" in threaded mode.

7.108 oscl_socket_types.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_types.h"
#include "oscl_scheduler_types.h"
#include "oscl_namestring.h"
#include "oscl_stdstring.h"
```

Data Structures

- class [OsclNetworkAddress](#)
- class [OsclSocketObserver](#)

Defines

- #define [PVNETWORKADDRESS_LEN](#) 50

Enumerations

- enum [TPVSocketFxn](#) { [EPVSocketSend](#) = 0, [EPVSocketSendTo](#), [EPVSocketRecv](#), [EPVSocketRecvFrom](#), [EPVSocketConnect](#), [EPVSocketAccept](#), [EPVSocketShutdown](#), [EPVSocketBind](#), [EPVSocketListen](#), [EPVSocket_Last](#) }
- enum [TPVSocketEvent](#) { [EPVSocketSuccess](#), [EPVSocketPending](#), [EPVSocketTimeout](#), [EPVSocketFailure](#), [EPVSocketCancel](#) }
- enum [TPVSocketShutdown](#) { [EPVSocketSendShutdown](#), [EPVSocketRecvShutdown](#), [EPVSocketBothShutdown](#) }

7.108.1 Define Documentation

7.108.1.1 #define PVNETWORKADDRESS_LEN 50

7.108.2 Enumeration Type Documentation

7.108.2.1 enum TPVSocketEvent

Return codes for asynchronous APIs

Enumeration values:

EPVSocketSuccess
EPVSocketPending
EPVSocketTimeout
EPVSocketFailure
EPVSocketCancel

7.108.2.2 enum TPVSocketFxn

Enumeration values:

- EPVSocketSend**
- EPVSocketSendTo**
- EPVSocketRecv**
- EPVSocketRecvFrom**
- EPVSocketConnect**
- EPVSocketAccept**
- EPVSocketShutdown**
- EPVSocketBind**
- EPVSocketListen**
- EPVSocket_Last**

7.108.2.3 enum TPVSocketShutdown

Enumeration values:

- EPVSocketSendShutdown**
- EPVSocketRecvShutdown**
- EPVSocketBothShutdown**

7.109 oscl_stdstring.h File Reference

This file provides standard string operations such as strlen, strcpy, etc.

```
#include "oscl_base.h"
```

Functions

- OSCL_IMPORT_REF uint32 `oscl_strlen` (const char *str)
- OSCL_IMPORT_REF uint32 `oscl_strlen` (const `oscl_wchar` *str)
- OSCL_IMPORT_REF char * `oscl_strncpy` (char *dest, const char *src, uint32 count)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_strncpy` (`oscl_wchar` *dest, const `oscl_wchar` *src, uint32 count)
- OSCL_IMPORT_REF int32 `oscl_strcmp` (const char *str1, const char *str2)
- OSCL_IMPORT_REF int32 `oscl_strcmp` (const `oscl_wchar` *str1, const `oscl_wchar` *str2)
- OSCL_IMPORT_REF int32 `oscl_strncmp` (const char *str1, const char *str2, uint32 count)
- OSCL_IMPORT_REF int32 `oscl_strncmp` (const `oscl_wchar` *str1, const `oscl_wchar` *str2, uint32 count)
- OSCL_IMPORT_REF char * `oscl_strncat` (char *dest, const char *src, uint32 count)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_strncat` (`oscl_wchar` *dest, const `oscl_wchar` *src, uint32 count)
- OSCL_IMPORT_REF const char * `oscl_strchr` (const char *str, int32 c)
- OSCL_IMPORT_REF char * `oscl_strchr` (char *str, int32 c)
- OSCL_IMPORT_REF const `oscl_wchar` * `oscl_strchr` (const `oscl_wchar` *str, int32 c)
- OSCL_IMPORT_REF `oscl_wchar` * `oscl_strchr` (`oscl_wchar` *str, int32 c)
- OSCL_IMPORT_REF const char * `oscl 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_Clstrcmp` (const char *str1, const char *str2)
- OSCL_IMPORT_REF int32 `oscl_Clstrcmp` (const `oscl_wchar` *str1, const `oscl_wchar` *str2)
- OSCL_IMPORT_REF int32 `oscl_Clstrncmp` (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)

7.109.1 Detailed Description

This file provides standard string operations such as strlen, strcpy, etc.

7.110 oscl_str_ptr_len.h File Reference

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

```
#include "oscl_base.h"
#include "oscl_stdstring.h"
```

Data Structures

- struct [StrCSumPtrLen](#)
same as [StrPtrLen](#), but includes checksum field and method to speed up querying
- struct [StrPtrLen](#)
This data structure encapsulates a set of functions used to perform.
- struct [WStrPtrLen](#)
This data structure encapsulates a set of functions used to perform.

Typedefs

- typedef StrPtrLen [StrPtrLen](#)
This data structure encapsulates a set of functions used to perform.
- typedef WStrPtrLen [WStrPtrLen](#)
This data structure encapsulates a set of functions used to perform.
- typedef StrCSumPtrLen [StrCSumPtrLen](#)
same as [StrPtrLen](#), but includes checksum field and method to speed up querying
- typedef [WStrPtrLen](#) [OSCL_TStrPtrLen](#)

Variables

- const uint8 [OSCL_ASCII_CASE_MAGIC_BIT](#) = 0x20

7.110.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

7.111 oscl_string.h File Reference

Provides a standardized set of string containers that can be used in place of character arrays.

```
#include "oscl_base.h"  
#include "oscl_mem.h"
```

Data Structures

- class [OSCL_String](#)
- class [OSCL_wString](#)

7.111.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

7.112 oscl_string_containers.h File Reference

Provides a standardized set of string containers that can be used in place of character arrays.

```
#include "oscl_string.h"  
#include "oscl_defalloc.h"  
#include "oscl_refcounter.h"  
#include "oscl_error.h"  
#include "oscl_string_rep.h"  
#include "oscl_stdstring.h"  
#include "oscl_mem.h"
```

Data Structures

- class [OSCL_FastString](#)
- class [OSCL_HeapString](#)
- class [OSCL_HeapStringA](#)
- class [OSCL_StackString](#)
- class [OSCL_wFastString](#)
- class [OSCL_wHeapString](#)
- class [OSCL_wHeapStringA](#)
- class [OSCL_wStackString](#)

7.112.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

7.113 oscl_string_rep.h File Reference

Contains some internal implementation for string containers.

```
#include "oscl_defalloc.h"
```

Data Structures

- class [CFastRep](#)
- class [CHheapRep](#)
- class [CStackRep](#)

7.113.1 Detailed Description

Contains some internal implementation for string containers.

7.114 oscl_string_uri.h File Reference

Utilities to unescape URIs.

```
#include "oscl_base.h"  
#include "oscl_string.h"
```

Functions

- OSCL_IMPORT_REF bool [oscl_str_unescape_uri](#) (const char *str_buf_in, char *str_buf_out, uint32 max_out_buf_bytes, uint32 max_bytes, uint32 &out_buf_len)
unescape any of the special escape sequence in the uri string
- OSCL_IMPORT_REF bool [oscl_str_unescape_uri](#) (const [OSCL_String](#) &oscl_str_in, [OSCL_String](#) &oscl_str_out, uint32 &out_buf_len)
unescape any of the special escape sequence in the uri string

7.114.1 Detailed Description

Utilities to unescape URIs.

7.115 oscl_string_utf8.h File Reference

Utilities to validate and truncate UTF-8 encoded strings.

```
#include "oscl_base.h"
```

Functions

- OSCL_IMPORT_REF bool `oscl_str_is_valid_utf8` (const uint8 *str_buf, uint32 &num_valid_characters, uint32 max_bytes=0, uint32 max_char_2_valid=0, uint32 *num_byte_4_char=NULL)

Check if the input string contains any illegal UTF-8 character. The function scans the string and validate that each character is a valid utf-8. It stops at the first NULL character, invalid character or the max_byte value. The string is valid if and only if every character is a valid utf-8 character and the scanning stopped on a character boundary.

- OSCL_IMPORT_REF int32 `oscl_str_truncate_utf8` (uint8 *str_buf, uint32 max_char, uint32 max_bytes=0)

Truncates the UTF-8 string upto the required size.

7.115.1 Detailed Description

Utilities to validate and truncate UTF-8 encoded strings.

7.116 oscl_string_utils.h File Reference

Utilities to parse and convert strings.

```
#include "oscl_base.h"
```

Defines

- #define `oscl_isdigit(c)` ((c) >= '0' && (c) <= '9')

Functions

- OSCL_IMPORT_REF const char * `skip_whitespace` (const char *ptr)
- OSCL_IMPORT_REF char * `skip_whitespace` (char *ptr)
- OSCL_IMPORT_REF const char * `skip_whitespace` (const char *start, const char *end)
- OSCL_IMPORT_REF const char * `skip_to_whitespace` (const char *start, const char *end)
- OSCL_IMPORT_REF const char * `skip_to_line_term` (const char *start_ptr, const char *end_ptr)
- OSCL_IMPORT_REF const char * `skip_whitespace_and_line_term` (const char *start, const char *end)
- OSCL_IMPORT_REF int `extract_string` (const char *in_ptr, char *outstring, int maxsize)
- OSCL_IMPORT_REF int `extract_string` (const char *start, const char *end, char *outstring, int maxsize)
- OSCL_IMPORT_REF bool `PV_atoi` (const char *buf, const char new_format, uint32 &value)
- OSCL_IMPORT_REF bool `PV_atoi` (const char *buf, const char new_format, int length, uint32 &value)
- OSCL_IMPORT_REF bool `PV_atoi` (const char *buf, const char new_format, int length, `uint64` &value)
- OSCL_IMPORT_REF bool `PV_atof` (const char *buf, `OsclFloat` &value)
- OSCL_IMPORT_REF bool `PV_atof` (const char *buf, int length, `OsclFloat` &value)
- OSCL_IMPORT_REF int `oscl_abs` (int aVal)

7.116.1 Detailed Description

Utilities to parse and convert strings.

7.117 oscl_string_xml.h File Reference

Utilities to escape special characters in XML strings.

```
#include "oscl_base.h"
```

Functions

- OSCL_IMPORT_REF bool [oscl_str_need_escape_xml](#) (const char *str_buf, uint32 &num_escape_bytes, uint32 max_bytes=0)

Check if the input string contains any special ASCII character like &, <, >, ', ". The function scans the string and check if each character is a special character. It stops at the first NULL character (if max_bytes = 0), or the max_byte value.

- OSCL_IMPORT_REF int32 [oscl_str_escape_xml](#) (const char *str_buf_in, char *str_buf_out, uint32 max_out_buf_bytes, uint32 max_bytes=0, uint32 *num_bytes_written=NULL)

Escape any of the following special characters in the string Special ASCII characters: &, <, >, ', ".

7.117.1 Detailed Description

Utilities to escape special characters in XML strings.

7.118 oscl_tagtree.h File Reference

The file [oscl_tagtree.h](#) ...

```
#include "oscl_base.h"
#include "oscl_map.h"
#include "oscl_vector.h"
#include "oscl_stdstring.h"
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- struct [const_iterator](#)
- struct [iterator](#)
- struct [Node](#)
- struct [Oscl_Tag](#)
- struct [Oscl_Tag_Base](#)
- class [Oscl_TagTree](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)

7.118.1 Detailed Description

The file [oscl_tagtree.h](#) ...

7.118.2 Define Documentation

7.118.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

7.119 oscl_tcp_socket.h File Reference

```
#include "oscl_ip_socket.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_listen.h"
#include "oscl_socket_recv.h"
#include "oscl_socket_send.h"
#include "oscl_socket_accept.h"
#include "oscl_socket_shutdown.h"
#include "oscl_socket_connect.h"
#include "oscl_socket_bind.h"
```

Data Structures

- class [OsclTCPSocketI](#)

7.120 oscl_thread.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_procstatus.h"
#include "oscl_base.h"
```

Data Structures

- class [OsclThread](#)

Typedefs

- typedef [TOsclThreadFuncRet](#)(OSCL_THREAD_DECL * [TOsclThreadFuncPtr](#))([TOsclThreadFuncArg](#))

Enumerations

- enum [OsclThread_State](#) { [Start_on_creation](#), [Suspend_on_creation](#) }
- enum [OsclThreadPriority](#) { [ThreadPriorityLowest](#), [ThreadPriorityLow](#), [ThreadPriorityBelowNormal](#), [ThreadPriorityNormal](#), [ThreadPriorityAboveNormal](#), [ThreadPriorityHighest](#), [ThreadPriorityTimeCritical](#) }

7.120.1 Detailed Description

. This file provides THREAD implementation that can be ported
to three OS LINUX, SYMBIAN, WIN32

7.120.2 Typedef Documentation

7.120.2.1 [typedef TOsclThreadFuncRet\(OSCL_THREAD_DECL * TOsclThreadFuncPtr\)\(TOsclThreadFuncArg\)](#)

7.120.3 Enumeration Type Documentation

7.120.3.1 enum [OsclThread_State](#)

Enumeration values:

[Start_on_creation](#)

[Suspend_on_creation](#)

7.120.3.2 enum [OsclThreadPriority](#)

Enumeration values:

[ThreadPriorityLowest](#)

[ThreadPriorityLow](#)

[ThreadPriorityBelowNormal](#)

ThreadPriorityNormal

ThreadPriorityAboveNormal

ThreadPriorityHighest

ThreadPriorityTimeCritical

7.121 oscl_tickcount.h File Reference

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

```
#include "oscl_base.h"
#include "oscl_tickcount.inl"
```

Data Structures

- class [OsclTickCount](#)

Defines

- #define [OSCLTICKCOUNT_MAX_TICKS](#) 0xffffffff

7.121.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

7.122 oscl_time.h File Reference

The file `oscl_time.h` defines two classes `NTPTime` and `TimeValue` for getting, manipulating, and formatting time values. The `TimeValue` class is based on the native system time format while `NTPTime` is used for the standard Network Time Protocol format.

```
#include "oscl_base.h"
#include "osclconfig_time.h"
#include "oscl_int64_utils.h"
#include "oscl_time.inl"
```

Data Structures

- class `NTPTime`

The NTPTime class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900.

- class `TimeValue`

The TimeValue class represents a time value in a format native to the system.

Typedefs

- typedef char `CtimeStrBuf [CTIME_BUFFER_SIZE]`
- typedef char `PV8601timeStrBuf [PV8601TIME_BUFFER_SIZE]`

Enumerations

- enum `TimeUnits` { `SECONDS` = 0, `MILLISECONDS` = 1, `MICROSECONDS` = 2 }

The TimeUnits enum can be used when constructing a `TimeValue` class.

Functions

- OSCL_IMPORT_REF void `PV8601ToRFC822` (`PV8601timeStrBuf` `pv8601_buffer`, `CtimeStrBuf` `ctime_buffer`)
- OSCL_IMPORT_REF void `RFC822ToPV8601` (`CtimeStrBuf` `ctime_buffer`, `PV8601timeStrBuf`)
- OSCL_COND_IMPORT_REF `TimeValue operator-` (`const TimeValue &a`, `const TimeValue &b`)

Variables

- const int `CTIME_BUFFER_SIZE` = 26
- const int `PV8601TIME_BUFFER_SIZE` = 21
- const long `USEC_PER_SEC` = 1000000
- const long `MSEC_PER_SEC` = 1000
- const uint32 `unix_ntp_offset` = 2208988800U

7.122.1 Detailed Description

The file `oscl_time.h` defines two classes `NTPTime` and `TimeValue` for getting, manipulating, and formatting time values. The `TimeValue` class is based on the native system time format while `NTPTime` is used for the standard Network Time Protocol format.

7.123 oscl_timer.h File Reference

```
#include "oscl_base.h"
#include "osclconfig_util.h"
#include "oscl_vector.h"
#include "oscl_tickcount.h"
#include "oscl_rand.h"
#include "oscl_scheduler_ao.h"
```

Data Structures

- struct [_TimerEntry](#)
- class [CallbackTimer](#)
- class [CallbackTimerObserver](#)
- class [OsclTimer](#)
- class [OsclTimerObserver](#)

7.124 oscl_tls.h File Reference

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

Data Structures

- class [OsclTLS](#)
- class [OsclTLSRegistry](#)
- class [TLSStorageOps](#)

Defines

- #define [OSCL_TLS_BASE_SLOTS](#) OSCL_TLS_ID_BASE_LAST +1
- #define [OSCL_TLS_EXTERNAL_SLOTS](#) 0
- #define [OSCL_TLS_MAX_SLOTS](#) (OSCL_TLS_BASE_SLOTS + OSCL_TLS_EXTERNAL_SLOTS)

Typedefs

- typedef [OsclAny](#) TOsclTlsKey

Variables

- const uint32 [OSCL_TLS_ID_MAGICNUM](#) = 0
- const uint32 [OSCL_TLS_ID_ERRORHOOK](#) = 1
- const uint32 [OSCL_TLS_ID_PVLOGGER](#) = 2
- const uint32 [OSCL_TLS_ID_TEST](#) = 3
- const uint32 [OSCL_TLS_ID_PVSCHEDULER](#) = 4
- const uint32 [OSCL_TLS_ID_PVERRORTRAP](#) = 5
- const uint32 [OSCL_TLS_ID_SDPMEDIAPARSER](#) = 6
- const uint32 [OSCL_TLS_ID_PAYLOADPARSER](#) = 7
- const uint32 [OSCL_TLS_ID_PVMFRECOGNIZER](#) = 8
- const uint32 [OSCL_TLS_ID_WMDRM](#) = 9
- const uint32 [OSCL_TLS_ID_OSCLREGISTRY](#) = 10
- const uint32 [OSCL_TLS_ID_SQLITE3](#) = 11
- const uint32 [OSCL_TLS_ID_BASE_LAST](#) = 11

7.125 oscl_tree.h File Reference

The file [oscl_tree.h](#) defines the template class [Oscl_Rb_Tree](#) which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the [Oscl_Map](#) class. Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_defalloc.h"  
#include "osclconfig_compiler_warnings.h"
```

Data Structures

- struct [Oscl_Pair](#)
- class [Oscl_Rb_Tree](#)
- class [Oscl_Rb_Tree_Base](#)
- struct [Oscl_Rb_Tree_Const_Iterator](#)
- struct [Oscl_Rb_Tree_Iterator](#)
- struct [Oscl_Rb_Tree_Node](#)
- struct [Oscl_Rb_Tree_Node_Base](#)

Defines

- #define [OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE](#)

7.125.1 Detailed Description

The file [oscl_tree.h](#) defines the template class [Oscl_Rb_Tree](#) which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the [Oscl_Map](#) class. Memory allocation is abstracted through the use of an allocator template parameter.

7.125.2 Define Documentation

7.125.2.1 #define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE

7.126 oscl_types.h File Reference

This file contains basic type definitions for common use across platforms.

```
#include "osclconfig.h"
```

Data Structures

- struct [OsclMemoryFragment](#)

Typedefs

- [typedef int c_bool](#)
The c_bool type is mapped to an integer to provide a bool type for C interfaces.
- [typedef void OsclAny](#)
The OsclAny is meant to be used the context of a generic pointer (i.e., no specific type).
- [typedef char mbchar](#)
mbchar is multi-byte char (e.g., UTF-8) with null termination.
- [typedef unsigned int uint](#)
The uint type is a convenient abbreviation for unsigned int.
- [typedef uint8 octet](#)
The octet type is meant to be used for referring to a byte or collection bytes without suggesting anything about the underlying meaning of the bytes.
- [typedef float OsclFloat](#)
The Float type defined as OsclFloat.
- [typedef OSCL_NATIVE_INT64_TYPE int64](#)
- [typedef OSCL_NATIVE_UINT64_TYPE uint64](#)
- [typedef OSCL_NATIVE_WCHAR_TYPE oscl_wchar](#)
- [typedef oscl_wchar OSCL_TCHAR](#)
define OSCL_TCHAR

7.126.1 Detailed Description

This file contains basic type definitions for common use across platforms.

7.127 oscl_udp_socket.h File Reference

```
#include "oscl_ip_socket.h"
#include "oscl_defalloc.h"
#include "oscl_socket_recv_from.h"
#include "oscl_socket_send_to.h"
#include "oscl_socket_bind.h"
```

Data Structures

- class [OsclUDPSocketI](#)

7.128 oscl_utf8conv.h File Reference

Utilities to convert unicode to utf8 and vice versa.

```
#include "oscl_base.h"
```

Functions

- OSCL_IMPORT_REF int32 [oscl_UTF8ToUnicode](#) (const char *input, int32 inLength, oscl_wchar *output, int32 outLength)

Convert UTF8 byte sequence to Unicode string.

- OSCL_IMPORT_REF int32 [oscl_UncodeToUTF8](#) (const oscl_wchar *input, int32 inLength, char *output, int32 outLength)

Convert Unicode string to UTF8 byte sequence.

7.128.1 Detailed Description

Utilities to convert unicode to utf8 and vice versa.

7.129 oscl_uuid.h File Reference

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32.

```
#include "oscl_base.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_string_utils.h"
#include "oscl_stdstring.h"
```

Data Structures

- struct [OsclUuid](#)

Defines

- #define [BYTES_IN_UUID_ARRAY](#) 8

Typedefs

- typedef uint32 [OsclUid32](#)

Variables

- const char [PV_CHAR_CLOSE_BRACKET](#) = ')
- const char [PV_CHAR_COMMA](#) = ','

7.129.1 Detailed Description

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32.

7.129.2 Define Documentation

7.129.2.1 #define BYTES_IN_UUID_ARRAY 8

7.129.3 Typedef Documentation

7.129.3.1 typedef uint32 OsclUid32

7.129.4 Variable Documentation

7.129.4.1 const char PV_CHAR_CLOSE_BRACKET = ')

7.129.4.2 const char PV_CHAR_COMMA = ','

7.130 oscl_vector.h File Reference

The file `oscl_vector.h` defines the template class `Oscl_Vector` which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_mem_basic_functions.h"
#include "oscl_assert.h"
#include "oscl_opaque_type.h"
#include "oscl_defalloc.h"
#include "oscl_base.h"
```

Data Structures

- class `Oscl_Vector`
- class `Oscl_Vector_Base`

7.130.1 Detailed Description

The file `oscl_vector.h` defines the template class `Oscl_Vector` which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

7.131 osclconfig.h File Reference

This file contains configuration information for the linux platform.

```
#include <dirent.h>
#include <dlfcn.h>
#include "osclconfig_limits_typedefs.h"
#include "osclconfig_unix_nj.h"
#include "osclconfig_ix86.h"
#include "osclconfig_check.h"
```

Defines

- #define OSCL_HAS_NJ_SUPPORT 1
- #define OSCL_HAS_NJ_FILE_IO_SUPPORT 1
- #define OSCL_RELEASE_BUILD 0
- #define OSCL_UNSIGNED_CONST(x) x##u
- #define OSCL_NATIVE_UINT64_TYPE u_int64_t
- #define OSCL_TEMPLATED_DESTRUCTOR_CALL(type, simple_type) ~type()
- #define __TFS__ <>
- #define OSCL_BEGIN_PACKED
- #define OSCL_PACKED_VAR(x) x __attribute__((packed))
- #define OSCL_PACKED_STRUCT_BEGIN
- #define OSCL_PACKED_STRUCT_END __attribute__((packed))
- #define OSCL_END_PACKED
- #define OSCL_ASSERT_ALWAYS 0

7.131.1 Detailed Description

This file contains configuration information for the linux platform.

7.131.2 Define Documentation

7.131.2.1 `#define __TFS__ <>`

7.131.2.2 `#define OSCL_BEGIN_PACKED`

7.131.2.3 `#define OSCL_END_PACKED`

7.131.2.4 `#define OSCL_HAS_NJ_FILE_IO_SUPPORT 1`

7.131.2.5 `#define OSCL_HAS_NJ_SUPPORT 1`

7.131.2.6 `#define OSCL_NATIVE_UINT64_TYPE u_int64_t`

7.131.2.7 `#define OSCL_PACKED_STRUCT_BEGIN`

7.131.2.8 `#define OSCL_PACKED_STRUCT_END __attribute__((packed))`

7.131.2.9 `#define OSCL_PACKED_VAR(x) x __attribute__((packed))`

7.131.2.10 `#define OSCL_RELEASE_BUILD 0`

7.131.2.11 `#define OSCL_TEMPLATED_DESTRUCTOR_CALL(type, simple_type) ~type()`

7.131.2.12 `#define OSCL_UNSIGNED_CONST(x) x##u`

7.132 osclconfig_ansi_memory.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include <memory.h>
```

Defines

- #define OSCL_HAS_ANSI_MEMORY_FUNCS 1

Typedefs

- typedef size_t oscl_memsize_t

7.132.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

7.132.2 Define Documentation

7.132.2.1 #define OSCL_HAS_ANSI_MEMORY_FUNCS 1

7.132.3 Typedef Documentation

7.132.3.1 typedef size_t oscl_memsize_t

7.133 osclconfig_check.h File Reference

Typedefs

- `typedef int8 __int8__check__`
- `typedef uint8 __uint8__check__`
- `typedef int16 __int16__check__`
- `typedef uint16 __uint16__check__`
- `typedef int32 __int32__check__`
- `typedef uint32 __uint32__check__`

7.134 osclconfig_compiler_warnings.h File Reference

This file contains the ability to turn off/on compiler warnings.

Defines

- #define OSCL_FUNCTION_PTR(x) (&x)

7.134.1 Detailed Description

This file contains the ability to turn off/on compiler warnings.

7.134.2 Define Documentation

7.134.2.1 #define OSCL_FUNCTION_PTR(x) (&x)

7.135 osclconfig_error.h File Reference

This file contains the common typedefs and header files needed to compile osclerror.

```
#include "osclconfig.h"  
#include <setjmp.h>  
#include <errno.h>  
#include "osclconfig_error_check.h"
```

Defines

- #define OSCL_HAS_EXCEPTIONS 1
- #define OSCL_HAS_ERRNO_H 1
- #define OSCL_HAS_SYMBIAN_ERRORTRAP 0
- #define OSCL_HAS_SETJMP_H 1

7.135.1 Detailed Description

This file contains the common typedefs and header files needed to compile osclerror.

7.135.2 Define Documentation

- 7.135.2.1 #define OSCL_HAS_ERRNO_H 1
- 7.135.2.2 #define OSCL_HAS_EXCEPTIONS 1
- 7.135.2.3 #define OSCL_HAS_SETJMP_H 1
- 7.135.2.4 #define OSCL_HAS_SYMBIAN_ERRORTRAP 0

7.136 osclconfig_error_check.h File Reference

7.137 osclconfig_global_new_delete.h File Reference

Functions

- void * **operator new** (size_t)
- void **operator delete** (void *)

7.138 osclconfig_global_placement_new.h File Reference

Functions

- void * [operator new](#) (size_t, void *ptr)

7.138.1 Function Documentation

7.138.1.1 void* operator new (size_t, void *ptr) [inline]

7.139 osclconfig_io.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include "osclconfig.h"  
  
#include <stdio.h>  
#include <stdlib.h>  
#include <stdarg.h>  
#include <sys/socket.h>  
#include <netinet/in.h>  
#include <arpa/inet.h>  
#include <fcntl.h>  
#include <signal.h>  
#include <netdb.h>  
#include <sys/mman.h>  
#include <sys/types.h>  
#include <errno.h>  
#include <sys/vfs.h>  
#include <dirent.h>  
#include <sys/stat.h>  
#include "osclconfig_io_check.h"
```

Defines

- #define OSCL_HAS_GLOB 0
- #define OSCL_HAS_ANSI_FILE_IO_SUPPORT 1
- #define OSCL_HAS_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_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

Typedefs

- typedef int **TOsclSocket**
- typedef sockaddr_in **TOsclSockAddr**
- typedef socklen_t **TOsclSockAddrLen**
- typedef hostent **TOsclHostent**

7.139.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

7.139.2 Define Documentation

7.139.2.1 #define OSCL_AF_INET AF_INET

7.139.2.2 #define OSCL_FILE_BUFFER_MAX_SIZE 32768

7.139.2.3 #define OSCL_HAS_ANSI_FILE_IO_SUPPORT 1

7.139.2.4 #define OSCL_HAS_BERKELEY_SOCKETS 1

7.139.2.5 #define OSCL_HAS_GLOB 0

7.139.2.6 #define OSCL_HAS_NATIVE_FILE_CACHE_ENABLE 1

7.139.2.7 #define OSCL_HAS_PV_FILE_CACHE 0

7.139.2.8 #define OSCL_HAS_SOCKET_SUPPORT 1

7.139.2.9 #define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0

7.139.2.10 #define OSCL_HAS_SYMBIAN_DNS_SERVER 0

7.139.2.11 #define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0

7.139.2.12 #define OSCL IPPROTO_TCP IPPROTO_TCP

7.139.2.13 #define OSCL IPPROTO_UDP IPPROTO_UDP

7.139.2.14 #define OSCL_SD_BOTH SHUT_RDWR

7.139.2.15 #define OSCL_SD_RECEIVE SHUT_RD

7.139.2.16 #define OSCL_SD_SEND SHUT_WR

7.139.2.17 #define OSCL_SOCK_DGRAM SOCK_DGRAM

7.139.2.18 #define OSCL_SOCK_STREAM SOCK_STREAM

7.139.2.19 #define OsclAccept(s, accept_s, ok, err, wouldblock)

Value:

```
accept_s=accept(s,NULL,NULL); \
ok=(accept_s!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

7.139.2.20 #define OsclBind(s, addr, ok, err)

Value:

```
TosclSockAddr* tmpadr = &addr; \
sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr); \
```

```
ok=(bind(s,sadr,sizeof(addr))!=(-1));\
if (!ok)err=errno
```

7.139.2.21 #define OsclCloseSocket(s, ok, err)

Value:

```
ok=(close(s)!=(-1));\
if (!ok)err=errno
```

7.139.2.22 #define OsclConnect(s, addr, ok, err, wouldblock)

Value:

```
TOsclSockAddr* tmpadr = &addr;\nsockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\nok=(connect(s,sadr,sizeof(addr))!=(-1));\
if (!ok){err=errno;wouldblock=(err==EINPROGRESS);}
```

7.139.2.23 #define OsclConnectComplete(s, wset, eset, success, fail, ok, err)

Value:

```
success=fail=false;\
if (FD_ISSET(s,&eset))\
{fail=true;OsclGetAsyncSockErr(s,ok,err);}\nelse if (FD_ISSET(s,&wset))\
{OsclGetAsyncSockErr(s,ok,err);if (ok && err==0)success=true;else fail=true;}
```

7.139.2.24 #define OsclGetAsyncSockErr(s, ok, err)

Value:

```
int opterr;socklen_t optlen(sizeof(opterr));\nok=(getsockopt(s,SOL_SOCKET,SO_ERROR,(void *)&opterr,&optlen)!=(-1));\
if(ok)err=opterr;else err=errno;
```

7.139.2.25 #define OsclGetDottedAddr(hostent, dottedaddr, ok)

Value:

```
long *_hostaddr=(long*)hostent->h_addr_list[0];\nstruct in_addr _inaddr;\n_inaddr.s_addr=_hostaddr;\ndottedaddr=inet_ntoa(_inaddr);\nok=(dottedaddr!=NULL);
```

7.139.2.26 #define OsclGethostbyname(name, hostent, ok, err)

Value:

```
hostent=gethostbyname((const char*)name); \
ok=(hostent!=NULL); \
if (!ok)err=errno;
```

7.139.2.27 #define OsclJoin(s, addr, ok, err)

Value:

```
{ \
    struct ip_mreq mreq; \
    void* p = &addr; \
    ok=(bind(s,(sockaddr*)p,sizeof(addr))!=(-1)); \
    mreq.imr_multiaddr.s_addr = addr.sin_addr.s_addr ; \
    mreq.imr_interface.s_addr = htonl(INADDR_ANY); \
    ok=(setsockopt(s, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq, sizeof(struct ip_mreq))!=(-1)); \
    if (!ok)err=errno; \
}
```

7.139.2.28 #define OsclListen(s, size, ok, err)

Value:

```
ok=(listen(iSocket,qSize)!=(-1)); \
if (!ok)err=errno
```

7.139.2.29 #define OsclMakeSockAddr(sockaddr, port, addrstr, ok)

Value:

```
sockaddr.sin_family=OSCL_AF_INET; \
sockaddr.sin_port=htons(port); \
int32 result/inet_aton((const char*)addrstr,&sockaddr.sin_addr); \
ok=(result!=0);
```

7.139.2.30 #define OsclRecv(s, buf, len, ok, err, nbytes, wouldblock)

Value:

```
nbytes=recv(s,(void *)buf),(size_t)(len),0); \
ok=(nbytes!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN);}
```

7.139.2.31 #define OsclRecvFrom(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)

Value:

```
\ 
void* p=paddr;\n
nbytes=recvfrom(s,(void*)(buf),(size_t)(len),0,(struct sockaddr*)p,paddrlen);\n
    ok=(nbytes!=(-1));\n
    if (!ok){err=errno;wouldblock=(err==EAGAIN);}\n
}
```

7.139.2.32 #define OsclSend(s, buf, len, ok, err, nbytes, wouldblock)

Value:

```
nbytes=send(s,(const void*)(buf),(size_t)(len),0);\n
    ok=(nbytes!=(-1));\n
    if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

7.139.2.33 #define OsclSendTo(s, buf, len, addr, ok, err, nbytes, wouldblock)

Value:

```
TOsclSockAddr* tmpadr = &addr;\n
sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr);\n
nbytes=sendto(s,(const void*)(buf),(size_t)(len),0,sadr,(socklen_t)sizeof(addr));\n
ok=(nbytes!=(-1));\n
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

7.139.2.34 #define OsclSetNonBlocking(s, ok, err)

Value:

```
ok=(fcntl(s,F_SETFL,O_NONBLOCK)!=(-1));\n
if (!ok)err=errno
```

7.139.2.35 #define OsclSetRecvBufferSize(s, val, ok, err)

Value:

```
ok=(setsockopt(s,SOL_SOCKET,SO_RCVBUF,(char*)&val, sizeof(int)) !=-1);\n
if (!ok)err=errno
```

7.139.2.36 #define OsclShutdown(s, how, ok, err)

Value:

```
ok=(shutdown(iSocket,how)!=(-1));\n
if (!ok)err=errno
```

7.139.2.37 #define OsclSocket(s, fam, type, prot, ok, err)**Value:**

```
s=socket(fam,type,prot);\
ok=(s!=(-1));\
if (!ok)err=errno
```

7.139.2.38 #define OsclSocketCleanup(ok)**Value:**

```
signal(SIGPIPE,SIG_DFL);\
ok=true
```

7.139.2.39 #define OsclSocketSelect(nfds, rd, wr, ex, timeout, ok, err, nhandles)**Value:**

```
nhandles=select(nfds,&rd,&wr,&ex,&timeout);\
ok=(nhandles!=(-1));\
if (!ok)err=errno
```

7.139.2.40 #define OsclSocketStartup(ok)**Value:**

```
signal(SIGPIPE,SIG_IGN);\
ok=true
```

7.139.2.41 #define OsclUnMakeSockAddr(sockaddr, addrstr) addrstr=inet_ntoa(sockaddr.sin_addr);**7.139.2.42 #define OsclValidInetAddr(addr) (inet_addr(addr)!=INADDR_NONE)**

7.139.3 Typedef Documentation

7.139.3.1 typedef struct hostent TOsclHostent**7.139.3.2 typedef struct sockaddr_in TOsclSockAddr****7.139.3.3 typedef socklen_t TOsclSockAddrLen****7.139.3.4 typedef int TOsclSocket**

7.140 osclconfig_io_check.h File Reference

7.141 osclconfig_ix86.h File Reference

This file contains configuration information for the ix86 processor family.

Defines

- #define OSCL_INTEGERS_WORD_ALIGNED 1
- #define OSCL_BYTE_ORDER_BIG_ENDIAN 0
- #define OSCL_BYTE_ORDER_LITTLE_ENDIAN 1

7.141.1 Detailed Description

This file contains configuration information for the ix86 processor family.

7.142 osclconfig_lib.h File Reference

This file contains configuration information for the ANSI build.

```
#include "osclconfig_lib_check.h"
```

Defines

- #define OSCL_HAS_RUNTIME_LIB_LOADING_SUPPORT 1
- #define PV_RUNTIME_LIB_FILENAME_EXTENSION "so"
- #define OSCL_LIB_READ_DEBUG_LIBS 1
- #define PV_DYNAMIC_LOADING_CONFIG_FILE_PATH "./"

7.142.1 Detailed Description

This file contains configuration information for the ANSI build.

7.142.2 Define Documentation

7.142.2.1 #define OSCL_HAS_RUNTIME_LIB_LOADING_SUPPORT 1

7.142.2.2 #define OSCL_LIB_READ_DEBUG_LIBS 1

7.142.2.3 #define PV_DYNAMIC_LOADING_CONFIG_FILE_PATH "./"

7.142.2.4 #define PV_RUNTIME_LIB_FILENAME_EXTENSION "so"

7.143 osclconfig_lib_check.h File Reference

7.144 osclconfig_limits_typedefs.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include <limits.h>
```

Defines

- #define OSCL_CHAR_IS_UNSIGNED 1
- #define OSCL_CHAR_IS_SIGNED 0

7.144.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header.

This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

7.144.2 Define Documentation

7.144.2.1 #define OSCL_CHAR_IS_SIGNED 0

7.144.2.2 #define OSCL_CHAR_IS_UNSIGNED 1

7.145 osclconfig_memory.h File Reference

```
#include "osclconfig.h"
#include "osclconfig_ansi_memory.h"
#include "osclconfig_memory_check.h"
```

Defines

- #define OSCL_BYPASS_MEMMGT 1
- #define OSCL_HAS_GLOBAL_NEW_DELETE 1
- #define PVMEM_INST_LEVEL 1
- #define OSCL_HAS_HEAP_BASE_SUPPORT 1
- #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0

7.145.1 Define Documentation

7.145.1.1 #define OSCL_BYPASS_MEMMGT 1

7.145.1.2 #define OSCL_HAS_GLOBAL_NEW_DELETE 1

7.145.1.3 #define OSCL_HAS_HEAP_BASE_SUPPORT 1

7.145.1.4 #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0

7.145.1.5 #define PVMEM_INST_LEVEL 1

7.146 osclconfig_memory_check.h File Reference

7.147 osclconfig_no_os.h File Reference

Defines

- #define OSCL_HAS_UNIX_SUPPORT 0
- #define OSCL_HAS_MSWIN_SUPPORT 0
- #define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_SUPPORT 0
- #define OSCL_HAS_SAVAJE_SUPPORT 0
- #define OSCL_HAS_PV_C_OS_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_ERRORTRAP 0
- #define OSCL_HAS_SYMBIAN_MEMORY_FUNCS 0
- #define OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS 0
- #define OSCL_HAS_PV_C_OS_TIME_FUNCS 0
- #define OSCL_HAS_UNIX_TIME_FUNCS 0
- #define OSCL_HAS_SYMBIAN_TIMERS 0
- #define OSCL_HAS_SYMBIAN_MATH 0
- #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0
- #define OSCL_HAS_PTHREAD_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0
- #define OSCL_HAS_SAVAJE_IO_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0
- #define OSCL_HAS_SYMBIAN_DNS_SERVER 0
- #define OSCL_HAS_BERKELEY_SOCKETS 0

7.148 osclconfig_proc.h File Reference

This file contains configuration information for the linux platform.

```
#include "osclconfig.h"  
#include "osclconfig_proc_unix_nj.h"  
#include "osclconfig_proc_check.h"
```

7.148.1 Detailed Description

This file contains configuration information for the linux platform.

7.149 osclconfig_proc_check.h File Reference

Typedefs

- `typedef TOsclThreadId __verify__TOsclThreadId_defined__`
- `typedef TOsclThreadFuncRet __verify__TOsclThreadFuncRet_defined__`
- `typedef TOsclThreadFuncArg __verify__TOsclThreadFuncArg_defined__`
- `typedef TOsclThreadObject __verify__TOsclThreadObject_defined__`
- `typedef TOsclMutexObject __verify__TOsclMutexObject_defined__`
- `typedef TOsclSemaphoreObject __verify__TOsclSemaphoreObject_defined__`
- `typedef TOsclConditionObject __verify__TOsclConditionObject_defined__`

7.149.1 Typedef Documentation

7.149.1.1 `typedef TOsclConditionObject __verify__TOsclConditionObject_defined__`

type TOsclConditionObject should be defined as the type used as a condition variable on the target platform.
Example: `typedef pthread_cond_t TOsclConditionObject;`

Note: Condition variables are only used with certain semaphore implementations. If the semaphore implementation does not require a condition variable, then this type can be defined as 'int' as follows: `typedef int TOsclConditionObject; //not used`

7.149.1.2 `typedef TOsclMutexObject __verify__TOsclMutexObject_defined__`

type TOsclMutexObject should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef pthread_mutex_t TOsclMutexObject;`

7.149.1.3 `typedef TOsclSemaphoreObject __verify__TOsclSemaphoreObject_defined__`

type TOsclSemaphoreObject should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef sem_t TOsclSemaphoreObject;`

7.149.1.4 `typedef TOsclThreadFuncArg __verify__TOsclThreadFuncArg_defined__`

type TOsclThreadFuncArg should be defined as the type used as a thread function argument on the target platform. Example: `typedef LPVOID TOsclThreadFuncArg;`

7.149.1.5 `typedef TOsclThreadFuncRet __verify__TOsclThreadFuncRet_defined__`

type TOsclThreadFuncRet should be defined as the type used as a thread function return value on the target platform. Example: `typedef DWORD TOsclThreadFuncRet;`

7.149.1.6 `typedef TOsclThreadId __verify__TOsclThreadId_defined__`

type TOsclThreadId should be defined as the type used as a thread ID on the target platform. Example: `typedef DWORD TOsclThreadId;`

7.149.1.7 **typedef TOsclThreadObject __verify__TOsclThreadObject_defined__**

type TOsclThreadObject should be defined as the type used as a thread object or handle on the target platform. Example: `typedef pthread_t TOsclThreadObject;`

7.150 osclconfig_proc_unix_common.h File Reference

```
#include <time.h>
#include <semaphore.h>
#include <pthread.h>
#include <errno.h>
```

Defines

- #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- #define OSCL_HAS_THREAD_SUPPORT 1
- #define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0
- #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 1
- #define OSCL_HAS_PTHREAD_SUPPORT 1
- #define OSCL_THREAD_DECL

Typedefs

- typedef pthread_t TOsclThreadId
- typedef void * TOsclThreadFuncArg
- typedef void * TOsclThreadFuncRet
- typedef pthread_t TOsclThreadObject
- typedef pthread_mutex_t TOsclMutexObject
- typedef sem_t TOsclSemaphoreObject
- typedef pthread_cond_t TOsclConditionObject

7.150.1 Define Documentation

- 7.150.1.1 `#define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0`
- 7.150.1.2 `#define OSCL_HAS_PTHREAD_SUPPORT 1`
- 7.150.1.3 `#define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 1`
- 7.150.1.4 `#define OSCL_HAS_SYMBIAN_SCHEDULER 0`
- 7.150.1.5 `#define OSCL_HAS_THREAD_SUPPORT 1`
- 7.150.1.6 `#define OSCL_THREAD_DECL`

7.150.2 Typedef Documentation

- 7.150.2.1 `typedef pthread_cond_t TOsclConditionObject`
- 7.150.2.2 `typedef pthread_mutex_t TOsclMutexObject`
- 7.150.2.3 `typedef sem_t TOsclSemaphoreObject`
- 7.150.2.4 `typedef void* TOsclThreadFuncArg`
- 7.150.2.5 `typedef void* TOsclThreadFuncRet`
- 7.150.2.6 `typedef pthread_t TOsclThreadId`
- 7.150.2.7 `typedef pthread_t TOsclThreadObject`

7.151 osclconfig_proc_unix_nj.h File Reference

```
#include <pthread.h>
#include <errno.h>
#include <signal.h>
```

Defines

- #define OSCL_HAS_SYMBIAN_SCHEDULER 0
- #define OSCL_HAS_THREAD_SUPPORT 1
- #define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0
- #define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0
- #define OSCL_HAS_PTHREAD_SUPPORT 1
- #define OSCL_THREAD_DECL

TypeDefs

- typedef pthread_t TOsclThreadId
- typedef void * TOsclThreadFuncArg
- typedef void * TOsclThreadFuncRet
- typedef pthread_t TOsclThreadObject
- typedef pthread_mutex_t TOsclMutexObject
- typedef int TOsclSemaphoreObject
- typedef pthread_cond_t TOsclConditionObject

7.151.1 Define Documentation

- 7.151.1.1 `#define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0`
- 7.151.1.2 `#define OSCL_HAS_PTHREAD_SUPPORT 1`
- 7.151.1.3 `#define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0`
- 7.151.1.4 `#define OSCL_HAS_SYMBIAN_SCHEDULER 0`
- 7.151.1.5 `#define OSCL_HAS_THREAD_SUPPORT 1`
- 7.151.1.6 `#define OSCL_THREAD_DECL`

7.151.2 Typedef Documentation

- 7.151.2.1 `typedef pthread_cond_t TOsclConditionObject`
- 7.151.2.2 `typedef pthread_mutex_t TOsclMutexObject`
- 7.151.2.3 `typedef int TOsclSemaphoreObject`
- 7.151.2.4 `typedef void* TOsclThreadFuncArg`
- 7.151.2.5 `typedef void* TOsclThreadFuncRet`
- 7.151.2.6 `typedef pthread_t TOsclThreadId`
- 7.151.2.7 `typedef pthread_t TOsclThreadObject`

7.152 osclconfig_time.h File Reference

```
#include "osclconfig.h"
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include "osclconfig_time_check.h"
```

Defines

- #define OSCL_HAS_UNIX_TIME_FUNCS 1

TypeDefs

- typedef timeval OsclBasicTimeStruct
- typedef tm OsclBasicDateTimeStruct

7.152.1 Define Documentation

7.152.1.1 #define OSCL_HAS_UNIX_TIME_FUNCS 1

7.152.2 Typedef Documentation

7.152.2.1 typedef tm OsclBasicDateTimeStruct

7.152.2.2 typedef struct timeval OsclBasicTimeStruct

7.153 osclconfig_time_check.h File Reference

Typedefs

- `typedef OsclBasicTimeStruct __Validate__BasicTimeStruct__`
- `typedef OsclBasicDateTimeStruct __Validate__BasicTimeDateStruct__`

7.153.1 Typedef Documentation

7.153.1.1 `typedef OsclBasicDateTimeStruct __Validate__BasicTimeDateStruct__`

`OsclBasicDateTimeStruct` type should be defined to the platform-specific date + time type.

7.153.1.2 `typedef OsclBasicTimeStruct __Validate__BasicTimeStruct__`

`OsclBasicTimeStruct` type should be defined to the platform-specific time of day type.

7.154 osclconfig_unix_common.h File Reference

```
#include <stdlib.h>
#include <stdarg.h>
#include <sys/types.h>
#include <stdio.h>
#include <wchar.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ctype.h>
#include <math.h>
```

Defines

- #define OSCL_DISABLE_INLINES 0
- #define OSCL_HAS_ANSI_STDLIB_SUPPORT 1
- #define OSCL_HAS_ANSI_MATH_SUPPORT 1
- #define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1
- #define OSCL_HAS_ANSI_STRING_SUPPORT 1
- #define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 1
- #define OSCL_HAS_ANSI_STDIO_SUPPORT 1
- #define OSCL_MEMFRAG_PTR_BEFORE_LEN 1
- #define OSCL_HAS_UNIX_SUPPORT 1
- #define OSCL_HAS_MSWIN_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_SUPPORT 0
- #define OSCL_HAS_NATIVE_INT64_TYPE 1
- #define OSCL_HAS_NATIVE_UINT64_TYPE 1
- #define OSCL_NATIVE_INT64_TYPE int64_t
- #define OSCL_NATIVE_UINT64_TYPE uint64_t
- #define INT64(x) x##LL
- #define UINT64(x) x##ULL
- #define INT64_HILO(high, low) (((high##LL))<<32)|low)
- #define UINT64_HILO(high, low) (((high##ULL))<<32)|low)
- #define OSCL_HAS_UNICODE_SUPPORT 1
- #define OSCL_NATIVE_WCHAR_TYPE wchar_t
- #define _STRLIT(x) L ## x
- #define _STRLIT_CHAR(x) x
- #define _STRLIT_WCHAR(x) L ## x
- #define OSCL_HAS_TLS_SUPPORT 1
- #define OSCL_TLS_IS_KEYED 1
- #define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)
- #define OSCL_TLS_KEY_DELETE_FUNC(key) pthread_key_delete(key)
- #define OSCL_TLS_STORE_FUNC(key, ptr) (pthread_setspecific(key,(const void*)ptr)==0)
- #define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)
- #define OSCL_HAS_BASIC_LOCK 1

Typedefs

- `typedef pthread_key_t TOsclTlsKey`
- `typedef pthread_mutex_t TOsclBasicLockObject`

7.154.1 Define Documentation

- 7.154.1.1 `#define _STRLIT(x) L ## x`
- 7.154.1.2 `#define _STRLIT_CHAR(x) x`
- 7.154.1.3 `#define _STRLIT_WCHAR(x) L ## x`
- 7.154.1.4 `#define INT64(x) x##LL`
- 7.154.1.5 `#define INT64_HILO(high, low) (((high##LL))<<32)|low)`
- 7.154.1.6 `#define OSCL_DISABLE_INLINES 0`
- 7.154.1.7 `#define OSCL_HAS_ANSI_MATH_SUPPORT 1`
- 7.154.1.8 `#define OSCL_HAS_ANSI_STDIO_SUPPORT 1`
- 7.154.1.9 `#define OSCL_HAS_ANSI_STDLIB_SUPPORT 1`
- 7.154.1.10 `#define OSCL_HAS_ANSI_STRING_SUPPORT 1`
- 7.154.1.11 `#define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 1`
- 7.154.1.12 `#define OSCL_HAS_BASIC_LOCK 1`
- 7.154.1.13 `#define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1`
- 7.154.1.14 `#define OSCL_HAS_MSWIN_SUPPORT 0`
- 7.154.1.15 `#define OSCL_HAS_NATIVE_INT64_TYPE 1`
- 7.154.1.16 `#define OSCL_HAS_NATIVE_UINT64_TYPE 1`
- 7.154.1.17 `#define OSCL_HAS_SYMBIAN_SUPPORT 0`
- 7.154.1.18 `#define OSCL_HAS_TLS_SUPPORT 1`
- 7.154.1.19 `#define OSCL_HAS_UNICODE_SUPPORT 1`
- 7.154.1.20 `#define OSCL_HAS_UNIX_SUPPORT 1`
- 7.154.1.21 `#define OSCL_MEMFRAG_PTR_BEFORE_LEN 1`
- 7.154.1.22 `#define OSCL_NATIVE_INT64_TYPE int64_t`
- 7.154.1.23 `#define OSCL_NATIVE_UINT64_TYPE uint64_t`
- 7.154.1.24 `#define OSCL_NATIVE_WCHAR_TYPE wchar_t`
- 7.154.1.25 `#define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)`
- 7.154.1.26 `#define OSCL_TLS_IS_KEYED 1`
- 7.154.1.27 `#define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)`

7.155 osclconfig_unix_nj.h File Reference

```
#include <stdlib.h>
#include <stdarg.h>
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ctype.h>
#include <math.h>
```

Defines

- #define OSCL_DISABLE_INLINES 0
- #define OSCL_HAS_ANSI_STDLIB_SUPPORT 1
- #define OSCL_HAS_ANSI_MATH_SUPPORT 1
- #define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1
- #define OSCL_HAS_ANSI_STRING_SUPPORT 1
- #define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 0
- #define OSCL_HAS_ANSI_STDIO_SUPPORT 1
- #define OSCL_MEMFRAG_PTR_BEFORE_LEN 1
- #define OSCL_HAS_UNIX_SUPPORT 1
- #define OSCL_HAS_MSWIN_SUPPORT 0
- #define OSCL_HAS_SYMBIAN_SUPPORT 0
- #define OSCL_HAS_NATIVE_INT64_TYPE 1
- #define OSCL_HAS_NATIVE_UINT64_TYPE 1
- #define OSCL_NATIVE_INT64_TYPE int64_t
- #define OSCL_NATIVE_UINT64_TYPE uint64_t
- #define INT64(x) x##LL
- #define UINT64(x) x##ULL
- #define INT64_HILO(high, low) (((high##LL))<<32)|low)
- #define UINT64_HILO(high, low) (((high##ULL))<<32)|low)
- #define OSCL_HAS_UNICODE_SUPPORT 1
- #define OSCL_NATIVE_WCHAR_TYPE wchar_t
- #define _STRLIT(x) L ## x
- #define _STRLIT_CHAR(x) x
- #define _STRLIT_WCHAR(x) L ## x
- #define OSCL_HAS_TLS_SUPPORT 1
- #define OSCL_TLS_IS_KEYED 1
- #define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)
- #define OSCL_TLS_KEY_DELETE_FUNC(key) pthread_key_delete(key)
- #define OSCL_TLS_STORE_FUNC(key, ptr) (pthread_setspecific(key,(const void*)ptr)==0)
- #define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)
- #define OSCL_HAS_BASIC_LOCK 1

Typedefs

- `typedef pthread_key_t TOsclTlsKey`
- `typedef pthread_mutex_t TOsclBasicLockObject`

7.155.1 Define Documentation

7.155.1.1 #define _STRLIT(x) L ## x
7.155.1.2 #define _STRLIT_CHAR(x) x
7.155.1.3 #define _STRLIT_WCHAR(x) L ## x
7.155.1.4 #define INT64(x) x##LL
7.155.1.5 #define INT64_HILO(high, low) (((high##LL))<<32)|low
7.155.1.6 #define OSCL_DISABLE_INLINES 0
7.155.1.7 #define OSCL_HAS_ANSI_MATH_SUPPORT 1
7.155.1.8 #define OSCL_HAS_ANSI_STDIO_SUPPORT 1
7.155.1.9 #define OSCL_HAS_ANSI_STDLIB_SUPPORT 1
7.155.1.10 #define OSCL_HAS_ANSI_STRING_SUPPORT 1
7.155.1.11 #define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 0
7.155.1.12 #define OSCL_HAS_BASIC_LOCK 1
7.155.1.13 #define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1
7.155.1.14 #define OSCL_HAS_MSWIN_SUPPORT 0
7.155.1.15 #define OSCL_HAS_NATIVE_INT64_TYPE 1
7.155.1.16 #define OSCL_HAS_NATIVE_UINT64_TYPE 1
7.155.1.17 #define OSCL_HAS_SYMBIAN_SUPPORT 0
7.155.1.18 #define OSCL_HAS_TLS_SUPPORT 1
7.155.1.19 #define OSCL_HAS_UNICODE_SUPPORT 1
7.155.1.20 #define OSCL_HAS_UNIX_SUPPORT 1
7.155.1.21 #define OSCL_MEMFRAG_PTR_BEFORE_LEN 1
7.155.1.22 #define OSCL_NATIVE_INT64_TYPE int64_t
7.155.1.23 #define OSCL_NATIVE_UINT64_TYPE uint64_t
7.155.1.24 #define OSCL_NATIVE_WCHAR_TYPE wchar_t
7.155.1.25 #define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)
7.155.1.26 #define OSCL_TLS_IS_KEYED 1
7.155.1.27 #define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)

7.156 osclconfig_util.h File Reference

```
#include "osclconfig.h"
#include <stdio.h>
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include "osclconfig_util_check.h"
```

Defines

- #define OSCL_CLOCK_HAS_DRIFT_CORRECTION 0
- #define OSCL_HAS_SYMBIAN_TIMERS 0
- #define OSCL_HAS_SYMBIAN_MATH 0
- #define OSCL RAND_MAX RAND_MAX
- #define SLEEP_ONE_SEC sleep(1)

7.156.1 Define Documentation

- 7.156.1.1 #define OSCL_CLOCK_HAS_DRIFT_CORRECTION 0
- 7.156.1.2 #define OSCL_HAS_SYMBIAN_MATH 0
- 7.156.1.3 #define OSCL_HAS_SYMBIAN_TIMERS 0
- 7.156.1.4 #define OSCL RAND_MAX RAND_MAX
- 7.156.1.5 #define SLEEP_ONE_SEC sleep(1)

7.157 osclconfig_util_check.h File Reference

7.158 pvlogger.h File Reference

This file contains basic logger interfaces for common use across platforms.

```
#include "oscl_base.h"
#include "oscl_vector.h"
#include "oscl_shared_ptr.h"
#include "oscl_base_alloc.h"
```

Data Structures

- class [PVLogger](#)

Defines

- #define [PVLOGMSG_INST_REL](#) 0
- #define [PVLOGMSG_INST_PROF](#) 1
- #define [PVLOGMSG_INST_HLDBG](#) 2
- #define [PVLOGMSG_INST_MLDBG](#) 3
- #define [PVLOGMSG_INST_LLDBG](#) 4
- #define [PVLOGGER_INST_LEVEL](#) 5
- #define [_PVLOGGER_LOGMSG](#)(LOGGER, LEVEL, MESSAGE)
- #define [_PVLOGGER_LOGMSG_V](#)(LOGGER, LEVEL, MESSAGE)
- #define [_PVLOGGER_LOGBIN](#)(LOGGER, LEVEL, MESSAGE)
- #define [_PVLOGGER_LOGBIN_V](#)(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_INST_LEVEL_SUPPORT](#) 1
- #define [PVLOGGER_LOGMSG_PVLOGMSG_INST_REL](#)(LOGGER, LEVEL, MESSAGE) _-
PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGMSG_V_PVLOGMSG_INST_REL](#)(LOGGER, LEVEL, MESSAGE) _-
PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGBIN_PVLOGMSG_INST_REL](#)(LOGGER, LEVEL, MESSAGE) _-
PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGBIN_V_PVLOGMSG_INST_REL](#)(LOGGER, LEVEL, MESSAGE) _-
PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGMSG_PVLOGMSG_INST_PROF](#)(LOGGER, LEVEL, MESSAGE) _-
PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGMSG_V_PVLOGMSG_INST_PROF](#)(LOGGER, LEVEL, MESSAGE)
_PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGBIN_PVLOGMSG_INST_PROF](#)(LOGGER, LEVEL, MESSAGE) _-
PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGBIN_V_PVLOGMSG_INST_PROF](#)(LOGGER, LEVEL, MESSAGE)
_PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGMSG_PVLOGMSG_INST_HLDBG](#)(LOGGER, LEVEL, MESSAGE)
_PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGMSG_V_PVLOGMSG_INST_HLDBG](#)(LOGGER, LEVEL, MESSAGE)
_PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGBIN_PVLOGMSG_INST_HLDBG](#)(LOGGER, LEVEL, MESSAGE)
_PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define [PVLOGGER_LOGBIN_V_PVLOGMSG_INST_HLDBG](#)(LOGGER, LEVEL, MESSAGE)
_PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)

- #define **PVLOGGER_LOGMSG_PVLOGMSG_INST_MLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGMSG_V_PVLOGMSG_INST_MLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN_PVLOGMSG_INST_MLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN_V_PVLOGMSG_V_INST_MLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGMSG_PVLOGMSG_INST_LLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGMSG_V_PVLOGMSG_INST_LLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN_PVLOGMSG_INST_LLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN_V_PVLOGMSG_INST_LLDBG**(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGMSG**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGMSG_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGMSG_V**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGMSG_V_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGBIN_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOGBIN_V**(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGBIN_V_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER_LOG_USE_ONLY**(x) x
- #define **PVLOGGER_ENABLE** 1

Variables

- const int32 **PVLOGGER_LEVEL_UNINITIALIZED** = -1
- const **PVLogger::log_level_type** **PVLOGMSG_EMERG** = 0
- const **PVLogger::log_level_type** **PVLOGMSG_ALERT** = 1
- const **PVLogger::log_level_type** **PVLOGMSG_CRIT** = 2
- const **PVLogger::log_level_type** **PVLOGMSG_ERR** = 3
- const **PVLogger::log_level_type** **PVLOGMSG_WARNING** = 4
- const **PVLogger::log_level_type** **PVLOGMSG_NOTICE** = 5
- const **PVLogger::log_level_type** **PVLOGMSG_INFO** = 6
- const **PVLogger::log_level_type** **PVLOGMSG_STACK_TRACE** = 7
- const **PVLogger::log_level_type** **PVLOGMSG_DEBUG** = 8
- const **PVLogger::log_level_type** **PVLOGMSG_FATAL_ERROR** = **PVLOGMSG_EMERG**
- const **PVLogger::log_level_type** **PVLOGMSG_NONFATAL_ERROR** = **PVLOGMSG_ERR**
- const **PVLogger::log_level_type** **PVLOGMSG_STATISTIC** = **PVLOGMSG_INFO**
- const **PVLogger::log_level_type** **PVLOGMSG_VERBOSE** = **PVLOGMSG_DEBUG**

7.158.1 Detailed Description

This file contains basic logger interfaces for common use across platforms.

This is the main entry point header file for the logger library. It should be the only one users directly include.

7.158.2 Define Documentation

7.158.2.1 #define _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgBuffers MESSAGE; \
    }\
  }\
}
```

7.158.2.2 #define _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgBuffersV MESSAGE; \
    }\
  }\
}
```

7.158.2.3 #define _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgString MESSAGE; \
    }\
  }\
}
```

7.158.2.4 #define _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)

Value:

```
{\
  if (LOGGER) \
  {\
    if (LOGGER->IsActive(LEVEL)) \
    {\
      LOGGER->LogMsgStringV MESSAGE; \
    }\
  }\
}
```

7.158.2.5 #define PVLOGGER_ENABLE 1

In case logging is compiled out, there is no need to compile the logger runtime code either.

7.158.2.6 #define PVLOGGER_INST_LEVEL 5**7.158.2.7 #define PVLOGGER_INST_LEVEL_SUPPORT 1****7.158.2.8 #define PVLOGGER_LOG_USE_ONLY(x) x**

Used to compile in/out lines of code that are used only for **PVLogger** macros.

This code will be removed at compile time when **PVLogger** is disabled, i.e. Release mode. So do not put in any code that is necessary for correct functionality of the module

**7.158.2.9 #define PVLOGGER_LOGBIN(IL, LOGGER, LEVEL, MESSAGE)
PVLOGGER_LOGBIN_ ## IL (LOGGER, LEVEL, MESSAGE)**

This is a binary API to log messages

Parameters:

IL Instrumentation level.

LOGGER Pointer to the logger object, that acts as the logging control/interface point

LEVEL Log level of the message

MESSAGE Log Message which includes the message id, and message buffers that need to be logged.

Example Usage: `PVLOGGER_LOGBIN (PVLOGMSG_INST_LLDBG, logger_1, PVLOGMSG_WARNING, (10, 3, msgBuf1Size, msgBuf1, msgBuf2Size, msgBuf2, msgBuf3Size, msgBuf3));`

-This message contains THREE (ptr_len, ptr) pairs. Log level of this msg is PVLOGMSG_WARNING, message id is 10.

- 7.158.2.10 #define PVLOGGER_LOGBIN_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 7.158.2.11 #define PVLOGGER_LOGBIN_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 7.158.2.12 #define PVLOGGER_LOGBIN_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 7.158.2.13 #define PVLOGGER_LOGBIN_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 7.158.2.14 #define PVLOGGER_LOGBIN_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 7.158.2.15 #define PVLOGGER_LOGBIN_V(IL, LOGGER, LEVEL, MESSAGE)
PVLOGGER_LOGBIN_V_## IL (LOGGER, LEVEL, MESSAGE)
- 7.158.2.16 #define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- 7.158.2.17 #define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- 7.158.2.18 #define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- 7.158.2.19 #define PVLOGGER_LOGBIN_V_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- 7.158.2.20 #define PVLOGGER_LOGBIN_V_PVLOGMSG_V_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGBIN_V(LOGGER, LEVEL, MESSAGE)
- 7.158.2.21 #define PVLOGGER_LOGMSG(IL, LOGGER, LEVEL, MESSAGE)
PVLOGGER_LOGMSG_## IL (LOGGER, LEVEL, MESSAGE)

This is the text based API to log messages

Parameters:

IL Instrumentation level.

LOGGER Pointer to the logger object, that acts as the logging control/interface point

LEVEL Log level of the message

MESSAGE Log Message which includes the message id, and any kind of formatting information

Example Usage: PVLOGGER_LOGMSG(PVLOGMSG_INST_LLDBG, logger_1, PVLOGMSG_WARNING, (13, "Test Messsage to Node 1

")); -This message of log level PVLOGMSG_WARNING, and has a message id of 13

- 7.158.2.22 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.23 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.24 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.25 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.26 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.27 `#define PVLOGGER_LOGMSG_V(IL, LOGGER, LEVEL, MESSAGE)
PVLOGGER_LOGMSG_V_## IL(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.28 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.29 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.30 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.31 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.32 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 7.158.2.33 `#define PVLOGMSG_INST_HLDBG 2`

High Level Debug Layer

This layer should contain messages that have very minimal impact on performance, but are at lower level (i.e., provide more information) than would be appropriate in a shipping product. The messages are probably used to gather information and validate proper functionality at a high level as might be appropriate for IOT, stress testing, or QA testing.

7.158.2.34 `#define PVLOGMSG_INST_LLDBG 4`

Low Level Debug Layer

This layer should contain messages for early functional testing. The messages are typically at a very low-level and allow testing the functionality of individual modules and components. Messages at this layer will typically have a performance impact (sometimes significant) due to the fact that they are at such a low level.

7.158.2.35 #define PVLOGMSG_INST_MLDBG 3

Mid Level Debug Layer

This layer should contain messages that are useful in the middle stages of the development cycle where major components are being integrated. The components themselves should already be well-tested so the emphasis is on interfaces between these components and integration testing. Messages at this layer may have some performance impact.

7.158.2.36 #define PVLOGMSG_INST_PROF 1

Profile Layer

The profile layer is used for messages and information related to measuring and reporting performance-related information.

7.158.2.37 #define PVLOGMSG_INST_REL 0

Release Layer

The release layer should only be used for messages that should remain in the final release. In certain cases all messaging may be disabled depending on customer requirements. However, when allowed the release layer should contain information that will be useful diagnosing problems in a released product (perhaps after entering a diagnostic mode), but with absolutely minimal performance impact when disabled at runtime.

7.158.3 Variable Documentation

7.158.3.1 const int32 PVLOGGER_LEVEL_UNINITIALIZED = -1

7.158.3.2 const PVLogger::log_level_type PVLOGMSG_ALERT = 1

action must be taken immediately

7.158.3.3 const PVLogger::log_level_type PVLOGMSG_CRIT = 2

critical conditions

7.158.3.4 const PVLogger::log_level_type PVLOGMSG_DEBUG = 8

debug-level messages

7.158.3.5 const PVLogger::log_level_type PVLOGMSG_EMERG = 0

system is unusable

7.158.3.6 const PVLogger::log_level_type PVLOGMSG_ERR = 3

error conditions

7.158.3.7 const **PVLogger::log_level_type** PVLOGMSG_FATAL_ERROR =
PVLOGMSG_EMERG

7.158.3.8 const **PVLogger::log_level_type** PVLOGMSG_INFO = 6

informational

7.158.3.9 const **PVLogger::log_level_type** PVLOGMSG_NONFATAL_ERROR =
PVLOGMSG_ERR

7.158.3.10 const **PVLogger::log_level_type** PVLOGMSG_NOTICE = 5

normal but significant condition

7.158.3.11 const **PVLogger::log_level_type** PVLOGMSG_STACK_TRACE = 7

function enter and exit

7.158.3.12 const **PVLogger::log_level_type** PVLOGMSG_STATISTIC = **PVLOGMSG_INFO**

7.158.3.13 const **PVLogger::log_level_type** PVLOGMSG_VERBOSE = **PVLOGMSG_DEBUG**

7.158.3.14 const **PVLogger::log_level_type** PVLOGMSG_WARNING = 4

warning conditions

7.159 pvlogger_accessories.h File Reference

```
#include "oscl_base.h"
#include "pvlogger.h"
```

Data Structures

- class [AllPassFilter](#)
- class [PVLoggerAppender](#)
- class [PVLoggerFilter](#)
- class [PVLoggerLayout](#)

Variables

- const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_ACCEPT = 1](#)
- const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_REJECT = 2](#)
- const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_NEUTRAL = 3](#)

7.159.1 Variable Documentation

7.159.1.1 const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_ACCEPT = 1](#)

7.159.1.2 const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_NEUTRAL = 3](#)

7.159.1.3 const [PVLoggerFilter::filter_status_type PVLOGGER_FILTER_REJECT = 2](#)

7.160 pvlogger_c.h File Reference

This file contains basic logger interfaces for common use across platforms. C-callable version.

```
#include "osclconfig.h"
```

Defines

- #define PVLOGGER_C_INST_LEVEL 5
- #define PVLOGMSG_C_INST_REL 0
- #define PVLOGMSG_C_INST_PROF 1
- #define PVLOGMSG_C_INST_HLDBG 2
- #define PVLOGMSG_C_INST_MLDBG 3
- #define PVLOGMSG_C_INST_LLDBG 4
- #define PVLOGMSG_C_EMERG 0
- #define PVLOGMSG_C_ALERT 1
- #define PVLOGMSG_C_CRIT 2
- #define PVLOGMSG_C_ERR 3
- #define PVLOGMSG_C_WARNING 4
- #define PVLOGMSG_C_NOTICE 5
- #define PVLOGMSG_C_INFO 6
- #define PVLOGMSG_C_STACK_TRACE 7
- #define PVLOGMSG_C_STACK_DEBUG 8

Functions

- OSCL_IMPORT_REF void * [pvLogger_GetLoggerObject](#) (const char *tag)
- OSCL_IMPORT_REF int [pvLogger_IsActive](#) (void *logger, int log_level)
- OSCL_IMPORT_REF void [pvLogger_LogMsgString](#) (void *logger, int msgID, const char *fmt,...)

7.160.1 Detailed Description

This file contains basic logger interfaces for common use across platforms. C-callable version.

This is the main entry point header file for the logger library. It should be the only one users directly include.

7.160.2 Define Documentation

- 7.160.2.1 `#define PVLOGGER_C_INST_LEVEL 5`
- 7.160.2.2 `#define PVLOGMSG_C_ALERT 1`
- 7.160.2.3 `#define PVLOGMSG_C_CRIT 2`
- 7.160.2.4 `#define PVLOGMSG_C_EMERG 0`
- 7.160.2.5 `#define PVLOGMSG_C_ERR 3`
- 7.160.2.6 `#define PVLOGMSG_C_INFO 6`
- 7.160.2.7 `#define PVLOGMSG_C_INST_HLDBG 2`
- 7.160.2.8 `#define PVLOGMSG_C_INST_LLDBG 4`
- 7.160.2.9 `#define PVLOGMSG_C_INST_MLDBG 3`
- 7.160.2.10 `#define PVLOGMSG_C_INST_PROF 1`
- 7.160.2.11 `#define PVLOGMSG_C_INST_REL 0`
- 7.160.2.12 `#define PVLOGMSG_C_NOTICE 5`
- 7.160.2.13 `#define PVLOGMSG_C_STACK_DEBUG 8`
- 7.160.2.14 `#define PVLOGMSG_C_STACK_TRACE 7`
- 7.160.2.15 `#define PVLOGMSG_C_WARNING 4`

7.160.3 Function Documentation

- 7.160.3.1 `OSCL_IMPORT_REF void* pvLogger_GetLoggerObject (const char * tag)`
- 7.160.3.2 `OSCL_IMPORT_REF int pvLogger_IsActive (void * logger, int log_level)`
- 7.160.3.3 `OSCL_IMPORT_REF void pvLogger_LogMsgString (void * logger, int msgID, const char * fmt, ...)`

7.161 pvlogger_registry.h File Reference

```
#include "pvlogger.h"
#include "oscl_tagtree.h"
```

Data Structures

- class [PVLoggerRegistry](#)

Index

~AllPassFilter
 AllPassFilter, 111
~BufFragGroup
 BufFragGroup, 117
~BufferMgr
 BufferMgr, 114
~CallbackTimer
 CallbackTimer, 120
~CallbackTimerObserver
 CallbackTimerObserver, 122
~DNSRequestParam
 DNSRequestParam, 129
~GetHostByNameParam
 GetHostByNameParam, 131
~HeapBase
 HeapBase, 133
~MM_AllocInfo
 MM_AllocInfo, 145
~MM_AllocNode
 MM_AllocNode, 146
~MM_Audit_Imp
 MM_Audit_Imp, 149
~MediaData
 MediaData, 138
~MemAllocator
 MemAllocator, 141
~OSCLMemAutoPtr
 OSCLMemAutoPtr, 416
~OSCL_FastString
 OSCL_FastString, 171
~OSCL_HeapString
 osclutil, 80
~OSCL_HeapStringA
 OSCL_HeapStringA, 192
~OSCL_StackString
 osclutil, 80
~OSCL_String
 OSCL_String, 249
~OSCL_wFastString
 OSCL_wFastString, 283
~OSCL_wHeapString
 osclutil, 80
~OSCL_wHeapStringA
 OSCL_wHeapStringA, 288
~OSCL_wStackString
 osclutil, 80
osclutil, 80
~OSCL_wString
 OSCL_wString, 293
~OsclAcceptMethod
 OsclAcceptMethod, 296
~OsclActiveObject
 OsclActiveObject, 299
~OsclAllocDestructDealloc
 OsclAllocDestructDealloc, 302
~OsclAsyncFile
 OsclAsyncFile, 305
~OsclAsyncFileBuffer
 OsclAsyncFileBuffer, 308
~OsclBinIStream
 OsclBinIStream, 312
~OsclBinOStream
 OsclBinOStream, 319
~OsclBindMethod
 OsclBindMethod, 310
~OsclComponentRegistry
 OsclComponentRegistry, 332
~OsclComponentRegistryElement
 OsclComponentRegistryElement, 334
~OsclConnectMethod
 OsclConnectMethod, 336
~OsclDNS
 OsclDNS, 339
~OsclDNSI
 OsclDNSI, 341
~OsclDNSIBase
 OsclDNSIBase, 344
~OsclDNSObserver
 OsclDNSObserver, 349
~OsclDNSRequest
 OsclDNSRequest, 350
~OsclExclusiveArrayPtr
 OsclExclusiveArrayPtr, 369
~OsclExclusivePtr
 OsclExclusivePtr, 372
~OsclExclusivePtrA
 OsclExclusivePtrA, 375
~OsclExecSchedulerCommonBase
 OsclExecSchedulerCommonBase, 383
~OsclFileCache
 OsclFileCache, 390

- ~OsclGetHostByNameMethod
 - OsclGetHostByNameMethod, [394](#)
- ~OsclIPSocketI
 - OsclIPSocketI, [399](#)
- ~OsclJump
 - OsclJump, [401](#)
- ~OsclListenMethod
 - OsclListenMethod, [402](#)
- ~OsclLockBase
 - OsclLockBase, [404](#)
- ~OsclMemAudit
 - OsclMemAudit, [409](#)
- ~OsclMemPoolAllocator
 - OsclMemPoolAllocator, [423](#)
- ~OsclMemPoolFixedChunkAllocator
 - OsclMemPoolFixedChunkAllocator, [425](#)
- ~OsclMemPoolFixedChunkAllocatorObserver
 - OsclMemPoolFixedChunkAllocator-
Observer, [428](#)
- ~OsclMemPoolResizableAllocator
 - OsclMemPoolResizableAllocator, [430](#)
- ~OsclMemPoolResizableAllocatorMemoryObserver
 - OsclMemPoolResizableAllocatorMemory-
Observer, [437](#)
- ~OsclMemPoolResizableAllocatorObserver
 - OsclMemPoolResizableAllocatorObserver,
[438](#)
- ~OsclMemStatsNode
 - OsclMemStatsNode, [439](#)
- ~OsclMutex
 - OsclMutex, [440](#)
- ~OsclNativeFile
 - OsclNativeFile, [444](#)
- ~OsclNullLock
 - OsclNullLock, [448](#)
- ~OsclPriorityQueue
 - OsclPriorityQueue, [452](#)
- ~OsclPriorityQueueBase
 - OsclPriorityQueueBase, [455](#)
- ~OsclRecvFromMethod
 - OsclRecvFromMethod, [467](#)
- ~OsclRecvMethod
 - OsclRecvMethod, [471](#)
- ~OsclRefCounter
 - OsclRefCounter, [473](#)
- ~OsclRefCounterDA
 - OsclRefCounterDA, [475](#)
- ~OsclRefCounterMTDA
 - OsclRefCounterMTDA, [479](#)
- ~OsclRefCounterMTSA
 - OsclRefCounterMTSA, [481](#)
- ~OsclRefCounterMemFrag
 - OsclRefCounterMemFrag, [477](#)
- ~OsclRefCounterSA
 - OsclRefCounterSA, [483](#)
- ~OsclRegistryAccessClient
 - OsclRegistryAccessClient, [485](#)
- ~OsclRegistryClient
 - OsclRegistryClient, [490](#)
- ~OsclRegistryServTlsImpl
 - OsclRegistryServTlsImpl, [496](#)
- ~OsclSchedulerObserver
 - OsclSchedulerObserver, [498](#)
- ~OsclScopedLock
 - OsclScopedLock, [499](#)
- ~OsclSemaphore
 - OsclSemaphore, [502](#)
- ~OsclSendMethod
 - OsclSendMethod, [504](#)
- ~OsclSendToMethod
 - OsclSendToMethod, [506](#)
- ~OsclSharedPtr
 - OsclSharedPtr, [509](#)
- ~OsclShutdownMethod
 - OsclShutdownMethod, [511](#)
- ~OsclSingleton
 - OsclSingleton, [513](#)
- ~OsclSocketI
 - OsclSocketI, [517](#)
- ~OsclSocketIBase
 - OsclSocketIBase, [522](#)
- ~OsclSocketMethod
 - OsclSocketMethod, [527](#)
- ~OsclSocketObserver
 - OsclSocketObserver, [529](#)
- ~OsclSocketRequestAO
 - OsclSocketRequestAO, [532](#)
- ~OsclSocketServ
 - OsclSocketServ, [535](#)
- ~OsclSocketServIBase
 - OsclSocketServIBase, [540](#)
- ~OsclTCPSocket
 - OsclTCPSocket, [545](#)
- ~OsclTCPSocketI
 - OsclTCPSocketI, [551](#)
- ~OsclTLS
 - OsclTLS, [570](#)
- ~OsclTLSEx
 - OsclTLSEx, [572](#)
- ~OsclThread
 - OsclThread, [553](#)
- ~OsclThreadLock
 - OsclThreadLock, [557](#)
- ~OsclTimer
 - OsclTimer, [561](#)
- ~OsclTimerObject
 - OsclTimerObject, [565](#)
- ~OsclTimerObserver

OsclTimerObserver, 568
 ~OsclUDPSocket
 OsclUDPSocket, 579
 ~OsclUDPSocketI
 OsclUDPSocketI, 585
 ~Oscl_File
 Oscl_File, 176
 ~Oscl_FileFind
 Oscl_FileFind, 182
 ~Oscl_FileServer
 Oscl_FileServer, 185
 ~Oscl_Linked_List
 Oscl_Linked_List, 197
 ~Oscl_Linked_List_Base
 Oscl_Linked_List_Base, 202
 ~Oscl_MTLinked_List
 Oscl_MTLinked_List, 214
 ~Oscl_Queue
 Oscl_Queue, 225
 ~Oscl_Queue_Base
 Oscl_Queue_Base, 227
 ~Oscl_Rb_Tree
 Oscl_Rb_Tree, 232
 ~Oscl_TAlloc
 Oscl_TAlloc, 270
 ~Oscl_Tag
 Oscl_Tag, 253
 ~Oscl_TagTree
 Oscl_TagTree, 258
 ~Oscl_Vector
 Oscl_Vector, 274
 ~Oscl_Vector_Base
 Oscl_Vector_Base, 279
 ~PVActiveBase
 PVActiveBase, 589
 ~PVLogger
 PVLogger, 594
 ~PVLoggerAppender
 PVLoggerAppender, 599
 ~PVLoggerFilter
 PVLoggerFilter, 601
 ~PVLoggerLayout
 PVLoggerLayout, 602
 ~PVLoggerRegistry
 PVLoggerRegistry, 604
 ~PVSchedulerStopper
 PVSchedulerStopper, 607
 ~PVThreadContext
 PVThreadContext, 610
 ~SendToParam
 SendToParam, 616
 ~OsclBasicAllocator
 _OsclBasicAllocator, 105
 ~_OsclHeapBase

 _OsclHeapBase, 107
 _OSCL_Abort
 osclbase, 33
 _OSCL_CLEANUP_BASE_CLASS
 osclmemory, 47
 _OSCL_TRAP_NEW
 osclmemory, 47
 _OsclBasicAllocator, 104
 _OsclBasicAllocator
 ~_OsclBasicAllocator, 105
 allocate, 105
 deallocate, 105
 _OsclHeapBase, 106
 _OsclHeapBase, 107
 _OsclHeapBase
 ~_OsclHeapBase, 107
 _OsclHeapBase, 107
 PVCleanupStack, 107
 _OsclInteger64Transport
 oscl_int64_utils.h, 679
 _Ownership
 OSCLMemAutoPtr, 418
 _PVLOGGER_LOGBIN
 pvlogger.h, 821
 _PVLOGGER_LOGBIN_V
 pvlogger.h, 821
 _PVLOGGER_LOGMSG
 pvlogger.h, 821
 _PVLOGGER_LOGMSG_V
 pvlogger.h, 821
 _PV_TRAP
 oscl_error_imp_fatalerror.h, 660
 oscl_error_imp_jumps.h, 661
 osclerror, 84
 _PV_TRAP_NO_TLS
 oscl_error_imp_fatalerror.h, 660
 oscl_error_imp_jumps.h, 661
 osclerror, 84
 _Ptr
 OsclExclusiveArrayPtr, 370
 OsclExclusivePtr, 373
 OsclExclusivePtrA, 376
 OsclSingleton, 514
 OsclTLS, 571
 OsclTLSEx, 573
 _STRLIT
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
 _STRLIT_CHAR
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
 _STRLIT_WCHAR
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816

__TFS__
 osclconfig.h, 777
 __Validate_BasicTimeDateStruct__
 osclconfig_time_check.h, 808
 __Validate_BasicTimeStruct__
 osclconfig_time_check.h, 808
 __int16_check__
 osclconfig, 22
 __int32_check__
 osclconfig, 22
 __int8_check__
 osclconfig, 22
 __uint16_check__
 osclconfig, 22
 __uint32_check__
 osclconfig, 22
 __uint8_check__
 osclconfig, 22
 __verify_TOsclConditionObject_defined__
 osclconfig_proc_check.h, 801
 __verify_TOsclMutexObject_defined__
 osclconfig_proc_check.h, 801
 __verify_TOsclSemaphoreObject_defined__
 osclconfig_proc_check.h, 801
 __verify_TOsclThreadFuncArg_defined__
 osclconfig_proc_check.h, 801
 __verify_TOsclThreadFuncRet_defined__
 osclconfig_proc_check.h, 801
 __verify_TOsclThreadId_defined__
 osclconfig_proc_check.h, 801
 __verify_TOsclThreadObject_defined__
 osclconfig_proc_check.h, 801
 _oscl_audit_malloc
 osclmemory, 56
 _oscl_audit_free
 osclmemory, 56
 _oscl_audit_malloc
 osclmemory, 56
 _oscl_audit_new
 osclmemory, 56
 _oscl_audit_realloc
 osclmemory, 57
 _oscl_malloc
 osclmemory, 57
 _oscl_default_audit_malloc
 osclmemory, 57
 _oscl_default_audit_new
 osclmemory, 57
 _oscl_default_audit_realloc
 osclmemory, 57
 _oscl_free
 osclmemory, 57
 _oscl_malloc
 osclmemory, 57
 _oscl_realloc
 osclmemory, 57
 a
 internalLeave, 134
 Abort
 OsclDNSMethod, 347
 OsclDNSRequestAO, 352
 OsclSocketMethod, 527
 OsclSocketRequestAO, 532
 AbortAll
 OsclDNSMethod, 347
 OsclSocketMethod, 527
 Accept
 OsclAcceptMethod, 296
 OsclAcceptRequest, 297
 OsclSocketI, 517
 OsclSocketIBase, 522
 OsclTCPSocket, 545
 OsclTCPSocketI, 551
 AcceptParam, 108
 AcceptParam, 108
 AcceptParam
 AcceptParam, 108
 iBlankSocket, 108
 AcceptRequest
 OsclAcceptMethod, 296
 Activate
 OsclDNSRequest, 350
 OsclSocketRequest, 530
 PVActiveBase, 589
 Add
 OsclSocketServRequestList, 541
 OsclTimerQ, 569
 add_element
 Oscl_Linked_List, 198
 Oscl_Linked_List_Base, 202
 Oscl_MTLLinked_List, 215
 add_ref
 CHHeapRep, 126
 add_to_front
 Oscl_Linked_List, 198
 Oscl_Linked_List_Base, 202
 Oscl_MTLLinked_List, 215
 addAllocNode
 MM_Audit_Imp, 149
 AddAppender
 PVLogger, 594
 AddFilter
 PVLogger, 594
 AddFragment
 BuffFragGroup, 117

AddLocalFragment
 MediaData, 138

addnewmempoolbuffer
 OsclMemPoolResizableAllocator, 430

addRef
 Oscl_DefAllocWithRefCounter, 168
 OsclMemPoolFixedChunkAllocator, 425
 OsclMemPoolResizableAllocator, 430
 OsclRefCounter, 473
 OsclRefCounterDA, 476
 OsclRefCounterMTDA, 480
 OsclRefCounterMTSA, 482
 OsclRefCounterSA, 484

address
 Oscl_TAlloc, 270

AddToExecTimerQ
 OsclExecSchedulerCommonBase, 383

AddToScheduler
 OsclActiveObject, 299
 OsclTimerObject, 565
 PVActiveBase, 589

After
 OsclTimerObject, 565

Alloc
 OsclIPSocketI, 399
 OsclSocketMethod, 527
 OsclSocketRequestAO, 532

ALLOC_AND_CONSTRUCT
 osclbase, 30

alloc_and_construct
 Oscl_TAlloc, 270

alloc_and_construct_fl
 Oscl_TAlloc, 270

ALLOC_NODE_FLAG
 osclmemory, 59

alloc_type
 PVLogger, 594
 PVLoggerRegistry, 604

ALLOCATE
 osclbase, 30

allocate
 _OsclBasicAllocator, 105
 MemAllocator, 141
 Oscl_Alloc, 165
 Oscl_DefAlloc, 167
 Oscl_Opaque_Type_Alloc, 218
 Oscl_Opaque_Type_Alloc_LL, 219
 Oscl_TAlloc, 270
 OsclErrorAllocator, 362
 OsclMemAllocator, 406
 OsclMemAllocDestructDealloc, 407
 OSCLMemAutoPtr, 417
 OsclMemBasicAllocator, 419
 OsclMemBasicAllocDestructDealloc, 420

OsclMemPoolFixedChunkAllocator, 425
 OsclMemPoolResizableAllocator, 431
 OsclReadyAlloc, 463

allocate_fl
 Oscl_Alloc, 165
 Oscl_DefAlloc, 167
 Oscl_TAlloc, 270
 OsclMemAllocator, 406
 OsclMemAllocDestructDealloc, 407
 OsclReadyAlloc, 463

allocateblock
 OsclMemPoolResizableAllocator, 431

allocator, 109

allocNum
 MM_AllocInfo, 145
 MM_AllocQueryInfo, 147

AllPassFilter, 110
 AllPassFilter, 111

AllPassFilter
 ~AllPassFilter, 111
 AllPassFilter, 111
 filter_status_type, 110
 FilterOpaqueMessge, 111
 FilterString, 111
 log_level_type, 110
 message_id_type, 110

ALREADY_SUSPENDED_ERROR
 OsclProcStatus, 456

Append
 OsclPtr, 458

append
 CFastRep, 124
 CHeapRep, 126
 CStackRep, 128

APPEND_MEDIA_AT_END
 osclutil, 80

append_rep
 CHeapRep, 126
 OSCL_String, 249
 OSCL_wString, 293

AppendBuffers
 PVLoggerAppender, 599

AppendNext
 BufFragGroup, 117

AppendString
 PVLoggerAppender, 599

assign
 CHeapRep, 126

assign_vector
 Oscl_Vector_Base, 279

asyncfilereadcancel_test
 Oscl_File, 180

asyncfilereadwrite_test
 Oscl_File, 180

Attach
 OsclBinStream, 325
 audit_type
 OsclMemGlobalAuditObject, 421
 available_localbuf
 MediaData, 139

 back
 Oscl_Queue, 225
 Oscl_Vector, 275
 BAD_THREADID_ADDR_ERROR
 OsclProcStatus, 456
 base_link_type
 Oscl_Rb_Tree_Base, 234
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 Oscl_Rb_Tree_Node_Base, 242
 begin
 Oscl_Map, 208
 Oscl_Rb_Tree, 232
 Oscl_TagTree, 258
 Oscl_Vector, 275
 BeginScheduling
 OsclExecSchedulerCommonBase, 383
 BeginStats
 OsclExecSchedulerCommonBase, 383
 BFG_SUCCESS
 BufFragStatusClass, 119
 big_endian_to_host
 osclbase, 33
 Bind
 osclbase, 33
 OsclBindMethod, 310
 OsclBindRequest, 311
 OsclIPSocketI, 399
 OsclSocketI, 517
 OsclSocketIBase, 522
 OsclTCPSocket, 545
 OsclUDPSocket, 580
 bind
 BufferState, 115
 BindAsync
 OsclSocketIBase, 522
 OsclTCPSocket, 545
 OsclTCPSocketI, 551
 OsclUDPSocket, 580
 OsclUDPSocketI, 585
 BindParam, 112
 BindParam, 112
 BindParam
 BindParam, 112
 iAddr, 112
 BindRequest
 OsclBindMethod, 310

 black
 Oscl_Rb_Tree_Node_Base, 242
 BlockingLoopL
 OsclExecSchedulerCommonBase, 383
 bSetFailure
 MM_AllocInfo, 145
 Buffer
 OsclAsyncFileBuffer, 308
 buffer
 CFastRep, 124
 CHeapRep, 126
 CStackRep, 128
 buffer_states
 BufFragGroup, 118
 BufferFragment, 113
 BufferFreeFuncPtr
 osclutil, 66
 BufferMgr, 114
 BufferMgr
 ~BufferMgr, 114
 BufferReleased, 114
 BufferReleased
 BufferMgr, 114
 BufferState, 115
 BufferState, 115
 BufferState
 bind, 115
 BufferState, 115
 decrement_refcnt, 115
 get_buf_mgr, 115
 get_free_function, 115
 get_ptr, 115
 get_refcount, 115
 increment_refcnt, 115
 reset, 115
 BufFragGroup, 116
 BufFragGroup, 117
 BufFragGroup
 ~BufFragGroup, 117
 AddFragment, 117
 AppendNext, 117
 buffer_states, 118
 BuffFragGroup, 117
 Clear, 117
 fragments, 118
 GetLength, 117
 GetMaxFrags, 118
 GetNext, 118
 GetNumFrags, 118
 length, 118
 next, 118
 num_frags, 118
 BufFragStatusClass, 119
 BFG_SUCCESS, 119

EMPTY_FRAGMENT, 119
 FIXED_FRAG_LOC_FULL, 119
 INTERNAL_ERROR, 119
 INVALID_ID, 119
 NOT_ENOUGH_SPACE, 119
 NULL_INPUT, 119
 TOO_MANY_FRAGS, 119
BufFragStatusClass
 status_t, 119
bufsize
 Oscl_Queue_Base, 229
 Oscl_Vector_Base, 281
BYTES_IN_UUID_ARRAY
 oscl_uuid.h, 774

c
 OsclPriorityQueue, 454
c_bool
 osclbase, 32
c_str
 StrPtrLen, 624
 WStrPtrLen, 634
Callback
 OsclReadyQ, 466
callback_timer_type
 OsclTimer, 561
CallbackTimer, 120
 CallbackTimer, 120
CallbackTimer
 ~CallbackTimer, 120
 CallbackTimer, 120
 Run, 120
CallbackTimer< Alloc >
 OsclTimer, 562
CallbackTimerObserver, 122
CallbackTimerObserver
 ~CallbackTimerObserver, 122
 TimerBaseElapsed, 122
CallRunExec
 OsclExecSchedulerCommonBase, 383
Cancel
 OsclActiveObject, 299
 OsclTimer, 561
 OsclTimerObject, 565
 PVActiveBase, 589
CancelAccept
 OsclSocketIBase, 523
 OsclTCPSocket, 545
 OsclTCPSocketI, 551
CancelBind
 OsclSocketIBase, 523
 OsclTCPSocket, 546
 OsclTCPSocketI, 551
 OsclUDPSocket, 580

OsclUDPSocketI, 585
CancelConnect
 OsclSocketIBase, 523
 OsclTCPSocket, 546
 OsclTCPSocketI, 551
CancelFreeChunkAvailableCallback
 OsclMemPoolFixedChunkAllocator, 425
 OsclMemPoolResizableAllocator, 431
CancelFreeMemoryAvailableCallback
 OsclMemPoolResizableAllocator, 431
CancelFxn
 OsclDNSIBase, 344
 OsclSocketIBase, 523
CancelGetHostByName
 OsclDNS, 339
 OsclDNSIBase, 344
CancelListen
 OsclSocketIBase, 523
 OsclTCPSocket, 546
 OsclTCPSocketI, 551
CancelMethod
 OsclDNSMethod, 347
 OsclSocketMethod, 527
CancelRecv
 OsclSocketIBase, 523
 OsclTCPSocket, 546
 OsclTCPSocketI, 551
CancelRecvFrom
 OsclSocketIBase, 523
 OsclUDPSocket, 580
 OsclUDPSocketI, 585
CancelRequest
 OsclDNSRequest, 350
 OsclSocketRequest, 530
CancelSend
 OsclSocketIBase, 523
 OsclTCPSocket, 546
 OsclTCPSocketI, 551
CancelSendTo
 OsclSocketIBase, 523
 OsclUDPSocket, 580
 OsclUDPSocketI, 585
CancelShutdown
 OsclSocketIBase, 523
 OsclTCPSocket, 546
 OsclTCPSocketI, 551
capacity
 Oscl_Queue_Base, 228
 Oscl_Vector_Base, 279
CFastRep, 123
 CFastRep, 124
CFastRep
 append, 124
 buffer, 124

CFastRep, 124
 maxsize, 124
 set_r, 124
 set_w, 124
 size, 124
 writable, 124
chartype
 OSCL_FastString, 171
 OSCL_HeapString, 189
 OSCL_HeapStringA, 191
 OSCL_StackString, 246
 OSCL_String, 249
 OSCL_wFastString, 282
 OSCL_wHeapString, 285
 OSCL_wHeapStringA, 288
 OSCL_wStackString, 291
 OSCL_wString, 293
CHearRep, 125
 CHearRep, 126
CHearRep
 add_ref, 126
 append, 126
 append_rep, 126
 assign, 126
 buffer, 126
 CHearRep, 126
 maxsize, 126
 refcount, 126
 remove_ref, 126
 set, 126
 set_rep, 126
 size, 126
check_fence
 MM_AllocBlockFence, 142
check_list
 Oscl_Linked_List, 198
 Oscl_Linked_List_Base, 202
checkSum
 StrCSumPtrLen, 621
CheckSumType
 StrCSumPtrLen, 621
children
 Oscl_TagTree::Node, 268
children_type
 Oscl_TagTree, 258
 Oscl_TagTree::Node, 268
CleanInUse
 OsclAsyncFileBuffer, 308
Cleanup
 OsclErrorTrap, 364
 OsclInit, 396
 OsclMem, 405
 OsclScheduler, 497
 PVLogger, 595
CleanupExecQ
 OsclExecSchedulerCommonBase, 383
CleanupParam
 OsclSocketRequestAO, 532
CleanupStatQ
 OsclExecSchedulerCommonBase, 383
Clear
 BufFragGroup, 117
 MediaData, 138
 OsclTimer, 561
clear
 Oscl_Map, 208
 Oscl_Queue, 225
 Oscl_Queue_Base, 228
 Oscl_Rb_Tree, 232
 Oscl_TagTree, 259
 Oscl_Vector, 275
Close
 Oscl_File, 176
 Oscl_FileFind, 182
 Oscl_FileServer, 185
 OsclAsyncFile, 305
 OsclDNSI, 341
 OsclDNSIBase, 344
 OsclFileCache, 390
 OsclIPSocketI, 399
 OsclMutex, 440
 OsclNativeFile, 444
 OsclRegistryAccessClient, 485
 OsclRegistryClient, 490
 OsclRegistryClientImpl, 493
 OsclRegistryServTlsImpl, 496
 OsclSemaphore, 502
 OsclSocketI, 517
 OsclSocketIBase, 523
 OsclSocketServ, 535
 OsclSocketServI, 537
 OsclSocketServIBase, 540
 OsclSocketServRequestList, 541
 OsclTCPSocket, 546
 OsclTCPSocketI, 551
 OsclUDPSocket, 580
 OsclUDPSocketI, 585
CloseSession
 OsclComponentRegistry, 332
color
 Oscl_Rb_Tree_Node_Base, 243
color_type
 Oscl_Rb_Tree_Node_Base, 242
comp
 Oscl_Map::value_compare, 212
 OsclPriorityQueue, 454
compare
 OsclCompareLess, 330

OsclReadyCompare, 464
 OsclTimerCompare, 563
 compare_data
 Oscl_Opaque_Type_Alloc_LL, 219
 compare_EQ
 Oscl_Opaque_Type_Compare, 221
 OsclPriorityQueue, 452
 compare_LT
 Oscl_Opaque_Type_Compare, 221
 OsclPriorityQueue, 452
 CompareId
 OsclThread, 553
 Complete
 OsclDNSRequest, 350
 OsclSocketRequest, 530
 COMPUTE_MEM_ALIGN_SIZE
 osclmemory, 48
 Connect
 Oscl_FileServer, 185
 OsclConnectMethod, 336
 OsclConnectRequest, 337
 OsclRegistryAccessClient, 485
 OsclRegistryClient, 490
 OsclRegistryClientImpl, 493
 OsclRegistryServTlsImpl, 496
 OsclSocketI, 517
 OsclSocketIBase, 523
 OsclSocketServ, 535
 OsclSocketServI, 537
 OsclSocketServIBase, 540
 OsclTCPSocket, 547
 OsclTCPSocketI, 551
 ConnectParam, 127
 ConnectParam, 127
 ConnectParam
 ConnectParam, 127
 iAddr, 127
 ConnectRequest
 OsclConnectMethod, 336
 const_iterator
 Oscl_Map, 207
 Oscl_Rb_Tree, 232
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_TagTree::const_iterator, 262
 Oscl_Vector, 274
 const_pointer
 Oscl_Rb_Tree, 232
 Oscl_TAlloc, 270
 const_reference
 Oscl_Map, 207
 Oscl_Queue, 225
 Oscl_Rb_Tree, 232
 Oscl_TAlloc, 270
 Oscl_Vector, 274
 OsclPriorityQueue, 452
 Construct
 OsclReadyQ, 466
 OsclTimerQ, 569
 construct
 Oscl_Linked_List_Base, 202
 Oscl_Opaque_Type_Alloc, 218
 Oscl_Opaque_Type_Alloc_LL, 219
 Oscl_Queue_Base, 228
 Oscl_TAlloc, 270
 Oscl_Vector_Base, 279
 OsclPriorityQueueBase, 455
 ConstructL
 OsclDNSMethod, 347
 OsclDNSRequestAO, 352
 OsclExecSchedulerCommonBase, 383
 OsclIPSocketI, 399
 OsclSocketMethod, 527
 OsclSocketRequestAO, 532
 ConstructStatQ
 OsclExecSchedulerCommonBase, 383
 container_type
 OsclPriorityQueue, 452
 count
 Oscl_Map, 208
 Oscl_Rb_Tree, 232
 Oscl_TagTree, 259
 CPVInterfaceProxy
 OsclErrorTrapImp, 366
 Create
 GetHostByNameParam, 131
 OsclMutex, 440
 OsclSemaphore, 502
 OsclThread, 554
 CreateMemPool
 OsclMemPoolAllocator, 423
 createmempool
 OsclMemPoolFixedChunkAllocator, 425
 CreatePVLogger
 PVLoggerRegistry, 605
 createStatsNode
 MM_Audit_Imp, 149
 CStackRep, 128
 CStackRep, 128
 CStackRep
 append, 128
 buffer, 128
 CStackRep, 128
 maxsize, 128
 set, 128
 size, 128
 CTIME_BUFFER_SIZE
 osclbase, 43
 CtimeStrBuf

osclbase, 32
 Current
 OsclExecScheduler, 377

 data
 LinkedListElement, 135
 data1
 OsclUuid, 587
 data2
 OsclUuid, 587
 data3
 OsclUuid, 587
 data4
 OsclUuid, 587
 deallocate
 _OsclBasicAllocator, 105
 MemAllocator, 141
 Oscl_Dalloc, 166
 Oscl_DefAlloc, 167
 Oscl_Opaque_Type_Alloc, 218
 Oscl_Opaque_Type_Alloc_LL, 219
 Oscl_TAlloc, 270
 OsclErrorAllocator, 362
 OsclMemAllocator, 406
 OsclMemAllocDestructDealloc, 407
 OSCLMemAutoPtr, 417
 OsclMemBasicAllocator, 419
 OsclMemBasicAllocDestructDealloc, 420
 OsclMemPoolFixedChunkAllocator, 426
 OsclMemPoolResizableAllocator, 431
 OsclReadyAlloc, 463
 deallocateblock
 OsclMemPoolResizableAllocator, 431
 decrement_refcnt
 BufferState, 115
 DEFAULT_MM_AUDIT_MODE
 osclmemory, 49
 DEFAULT_POSTFILL_PATTERN
 osclmemory, 49
 DEFAULT_PREFILL_PATTERN
 osclmemory, 49
 Delete
 Oscl_DefAllocWithRefCounter, 168
 OsclAsyncFile, 305
 OsclBuf, 329
 Depth
 OsclReadyQ, 466
 depth
 Oscl_TagTree::Node, 268
 dequeue_element
 Oscl_Linked_List, 198
 Oscl_MTLLinked_List, 215
 Des
 OsclBuf, 329

 DesC
 OsclBuf, 329
 Destroy
 DNSRequestParam, 129
 GetHostByNameParam, 131
 PVActiveBase, 589
 destroy
 Oscl_Linked_List_Base, 202
 Oscl_Opaque_Type_Alloc, 218
 Oscl_Opaque_Type_Alloc_LL, 219
 Oscl_Queue_Base, 228
 Oscl_TAlloc, 270
 Oscl_Vector, 275
 Oscl_Vector_Base, 279
 destroyallmempoolbuffers
 OsclMemPoolResizableAllocator, 431
 DestroyMemPool
 OsclMemPoolAllocator, 423
 destroymempool
 OsclMemPoolFixedChunkAllocator, 426
 destruct_and_dealloc
 Oscl_TAlloc, 270
 OsclDestructDealloc, 338
 OsclMemAllocDestructDealloc, 407
 OsclMemBasicAllocDestructDealloc, 420
 difference_type
 Oscl_Rb_Tree, 232
 DIR_TYPE
 Oscl_FileFind, 181
 DisableAppenderInheritance
 PVLogger, 595
 DiscardAcceptedSocket
 OsclAcceptMethod, 296
 DNSRequestParam, 129
 DNSRequestParam, 129
 OsclDNSI, 342
 OsclDNSRequestAO, 353
 DNSRequestParam
 ~DNSRequestParam, 129
 Destroy, 129
 DNSRequestParam, 129
 iDNSRequest, 130
 iFxn, 130
 InThread, 129
 iRefCount, 130
 RemoveRef, 130
 DoCancel
 OsclActiveObject, 300
 OsclDNSRequestAO, 352
 OsclSocketRequestAO, 532
 OsclTimerObject, 565
 PVActiveBase, 589

 E_BUFFER_TOO_SMALL

Oscl_FileFind, 182
E_INVALID_ARG
 Oscl_FileFind, 181
E_INVALID_STATE
 Oscl_FileFind, 181
E_NO_MATCH
 Oscl_FileFind, 182
E_NOT_IMPLEMENTED
 Oscl_FileFind, 182
E_OK
 Oscl_FileFind, 181
E_OTHER
 Oscl_FileFind, 182
E_PATH_NOT_FOUND
 Oscl_FileFind, 181
E_PATH_TOO_LONG
 Oscl_FileFind, 181
element_type
 Oscl_FileFind, 181
elems
 Oscl_Queue_Base, 229
 Oscl_Vector_Base, 281
empty
 Oscl_Map, 208
 Oscl_Queue_Base, 228
 Oscl_Rb_Tree, 232
 Oscl_TagTree, 259
 Oscl_Vector_Base, 279
 OsclPriorityQueue, 453
EMPTY_FRAGMENT
 BufFragStatusClass, 119
EnableKill
 OsclThread, 554
enablenullpointerreturn
 OsclMemPoolFixedChunkAllocator, 426
 OsclMemPoolResizableAllocator, 431
End
 OsclFileStats, 392
end
 Oscl_Map, 208
 Oscl_Rb_Tree, 232
 Oscl_TagTree, 259
 Oscl_Vector, 275
EndOfFile
 Oscl_File, 176
 OsclAsyncFile, 305
 OsclFileCache, 390
 OsclNativeFile, 444
EndScheduling
 OsclExecSchedulerCommonBase, 383
EndStats
 OsclExecSchedulerCommonBase, 383
EnterThreadContext
 PVThreadContext, 610

eof
 OsclBinStream, 325
EOF_STATE
 OsclBinStream, 325
EOsclFileOp_Close
 osclio, 94
EOsclFileOp_EndOfFile
 osclio, 94
EOsclFileOp_Flush
 osclio, 94
EOsclFileOp_Last
 osclio, 94
EOsclFileOp_NativeClose
 osclio, 94
EOsclFileOp_NativeEndOfFile
 osclio, 94
EOsclFileOp_NativeFlush
 osclio, 94
EOsclFileOp_NativeOpen
 osclio, 94
EOsclFileOp_NativeRead
 osclio, 94
EOsclFileOp_NativeSeek
 osclio, 94
EOsclFileOp_NativeSize
 osclio, 94
EOsclFileOp_NativeTell
 osclio, 94
EOsclFileOp_NativeWrite
 osclio, 94
EOsclFileOp_Open
 osclio, 94
EOsclFileOp_Read
 osclio, 94
EOsclFileOp_Seek
 osclio, 94
EOsclFileOp_Size
 osclio, 94
EOsclFileOp_Tell
 osclio, 94
EOsclFileOp_Write
 osclio, 94
eOsclProcError
 OsclProcStatus, 456
EOsclSocket_DataRecv
 oscl_socket_stats.h, 747
EOsclSocket_DataSent
 oscl_socket_stats.h, 747
EOsclSocket_Except
 oscl_socket_stats.h, 746
EOsclSocket_OS
 oscl_socket_stats.h, 746
EOsclSocket_Readable
 oscl_socket_stats.h, 746

EOscISocket_RequestAO_Canceled
 oscl_socket_stats.h, 746
 EOscISocket_RequestAO_Error
 oscl_socket_stats.h, 746
 EOscISocket_RequestAO_Success
 oscl_socket_stats.h, 746
 EOscISocket_RequestAO_Timeout
 oscl_socket_stats.h, 746
 EOscISocket_ServPoll
 oscl_socket_stats.h, 746
 EOscISocket_ServRequestCancelIssued
 oscl_socket_stats.h, 747
 EOscISocket_ServRequestComplete
 oscl_socket_stats.h, 747
 EOscISocket_ServRequestIssued
 oscl_socket_stats.h, 746
 EOscISocket_Writable
 oscl_socket_stats.h, 746
 EOscISocketServ_LastEvent
 oscl_socket_stats.h, 746
 EOscISocketServ_LoopsockError
 oscl_socket_stats.h, 747
 EOscISocketServ_LoopsockOk
 oscl_socket_stats.h, 747
 EOscISocketServ_SelectActivity
 oscl_socket_stats.h, 746
 EOscISocketServ_SelectNoActivity
 oscl_socket_stats.h, 746
 EOscISocketServ_SelectRescheduleAsap
 oscl_socket_stats.h, 746
 EOscISocketServ_SelectReschedulePoll
 oscl_socket_stats.h, 746
 EOtherExecStats_Last
 OsclExecSchedulerCommonBase, 382
 EOtherExecStats_NativeOS
 OsclExecSchedulerCommonBase, 382
 EOtherExecStats_QueueTime
 OsclExecSchedulerCommonBase, 382
 EOtherExecStats_ReleaseTime
 OsclExecSchedulerCommonBase, 382
 EOtherExecStats_WaitTime
 OsclExecSchedulerCommonBase, 382
 EPriorityHigh
 OsclActiveObject, 299
 EPriorityHighest
 OsclActiveObject, 299
 EPriorityIdle
 OsclActiveObject, 299
 EPriorityLow
 OsclActiveObject, 299
 EPriorityNominal
 OsclActiveObject, 299
 EPVDNSCancel
 osclio, 95
 EPVDNSFailure
 osclio, 95
 EPVDNSGetHostByName
 osclio, 95
 EPVDNSPending
 osclio, 95
 EPVDNSSuccess
 osclio, 95
 EPVDNSTimeout
 osclio, 95
 EPVSocket_Last
 oscl_socket_types.h, 751
 EPVSocketAccept
 oscl_socket_types.h, 751
 EPVSocketBind
 oscl_socket_types.h, 751
 EPVSocketBothShutdown
 oscl_socket_types.h, 751
 EPVSocketCancel
 oscl_socket_types.h, 750
 EPVSocketConnect
 oscl_socket_types.h, 751
 EPVSocketFailure
 oscl_socket_types.h, 750
 EPVSocketListen
 oscl_socket_types.h, 751
 EPVSocketPending
 oscl_socket_types.h, 750
 EPVSocketRecv
 oscl_socket_types.h, 751
 EPVSocketRecvFrom
 oscl_socket_types.h, 751
 EPVSocketRecvShutdown
 oscl_socket_types.h, 751
 EPVSocketSend
 oscl_socket_types.h, 751
 EPVSocketSendShutdown
 oscl_socket_types.h, 751
 EPVSocketSendTo
 oscl_socket_types.h, 751
 EPVSocketShutdown
 oscl_socket_types.h, 751
 EPVSocketSuccess
 oscl_socket_types.h, 750
 EPVSocketTimeout
 oscl_socket_types.h, 750
 EPVThreadContext_InThread
 osclproc, 102
 EPVThreadContext_NonOsclThread
 osclproc, 102
 EPVThreadContext_OsclThread
 osclproc, 102
 EPVThreadContext_Undetermined
 osclproc, 102

equal_range
 Oscl_Map, 208
 Oscl_Rb_Tree, 232
 erase
 Oscl_Map, 209
 Oscl_Rb_Tree, 232
 Oscl_TagTree, 259
 Oscl_Vector, 275
 Oscl_Vector_Base, 279, 280
 Error
 OsclExecSchedulerCommonBase, 383
 error_type
 Oscl_FileFind, 181
 ESocketServ_Connected
 OsclSocketServIBase, 539
 ESocketServ_Error
 OsclSocketServIBase, 540
 ESocketServ_Idle
 OsclSocketServIBase, 539
 ESymbianAccessMode_Rfile
 Oscl_File, 175
 ESymbianAccessMode_RfileBuf
 Oscl_File, 175
 EXCEED_MAX_COUNT_VARIABLE_-
 ERROR
 OsclProcStatus, 457
 EXCEED_MAX_SEM_COUNT_ERROR
 OsclProcStatus, 457
 Exit
 OsclThread, 554
 ExitThreadContext
 PVThreadContext, 610
 extract_string
 osclutil, 66

 fail
 OsclBinStream, 326
 FAIL_STATE
 OsclBinStream, 325
 FENCE_PATTERN
 osclmemory, 49
 FILE_TYPE
 Oscl_FileFind, 181
 fileName
 MM_AllocQueryInfo, 147
 FileSize
 OsclFileCache, 390
 fill_fence
 MM_AllocBlockFence, 142
 filter_status_type
 AllPassFilter, 110
 PVLogger, 594
 PVLoggerFilter, 600
 FilterOpaqueMessge

AllPassFilter, 111
 PVLoggerFilter, 601
 FilterString
 AllPassFilter, 111
 PVLoggerFilter, 601
 Find
 OsclComponentRegistryData, 333
 find
 Oscl_Map, 209
 Oscl_Rb_Tree, 232
 Oscl_TagTree, 259
 find_heap
 OsclPriorityQueue, 453
 OsclPriorityQueueBase, 455
 FindExact
 OsclComponentRegistry, 332
 FindFirst
 Oscl_FileFind, 182
 findfreeblock
 OsclMemPoolResizableAllocator, 432
 FindHierarchical
 OsclComponentRegistry, 332
 FindNext
 Oscl_FileFind, 183
 FindPVBase
 OsclExecSchedulerCommonBase, 383
 first
 Oscl_Pair, 223
 firstFragPtr
 OsclBinStream, 327
 FIXED_FRAG_LOC_FULL
 BuffFragStatusClass, 119
 Flush
 Oscl_File, 176
 OsclAsyncFile, 305
 OsclFileCache, 390
 OsclNativeFile, 444
 FormatOpaqueMessage
 PVLoggerLayout, 602
 FormatString
 PVLoggerLayout, 602
 fragments
 BuffFragGroup, 118
 fragsLeft
 OsclBinStream, 327
 freeblockavailable
 OsclMemPoolResizableAllocatorObserver,
 438
 freebytes
 oscl_fsstat, 187
 freechunkavailable
 OsclMemPoolFixedChunkAllocator-
 Observer, 428
 freememoryavailable

OsclMemPoolResizableAllocatorMemory-
 Observer, 437
front
 Oscl_Queue, 226
 Oscl_Vector, 276
Fxn
 OsclSocketRequest, 530
get
 OsclBinIStream, 312
 OsclExclusiveArrayPtr, 369
 OsclExclusivePtr, 372
 OsclExclusivePtrA, 375
 OSCLMemAutoPtr, 417
get_buf_mgr
 BufferState, 115
get_count
 OsclSharedPtr, 509
get_cstr
 OSCL_FastString, 171
 OSCL_HeapStringA, 192
 OSCL_String, 249
 OSCL_wFastString, 283
 OSCL_wHeapStringA, 288
 OSCL_wString, 293
 osclutil, 66, 67
get_data
 Oscl_Opaque_Type_Alloc_LL, 220
get_element
 Oscl_Linked_List, 198
 Oscl_Linked_List_Base, 202
 Oscl_MTLINKED_List, 215
get_first
 Oscl_Linked_List, 198
 Oscl_Linked_List_Base, 203
get_free_function
 BufferState, 115
get_index
 Oscl_Linked_List, 199
 Oscl_Linked_List_Base, 203
 Oscl_MTLINKED_List, 215
get_int64_lower32
 Oscl_Int64_Utils, 195
get_int64_middle32
 Oscl_Int64_Utils, 195
get_int64_upper32
 Oscl_Int64_Utils, 195
get_local_time
 TimeValue, 627
get_lower32
 NTPTime, 163
get_maxsize
 OSCL_FastString, 171
 OSCL_HeapStringA, 192
 OSCL_String, 249
 OSCL_wFastString, 283
 OSCL_wHeapStringA, 288
 OSCL_wString, 293
 osclutil, 67, 68
get_num_elements
 Oscl_Linked_List, 199
get_next
 Oscl_Linked_List, 199
 Oscl_Linked_List_Base, 203
 Oscl_Opaque_Type_Alloc_LL, 220
get_ptr
 BufferState, 115
get_pv8601_str_time
 TimeValue, 627
get_refcount
 BufferState, 115
get_registry
 TLSStorageOps, 631
get_rfc822_gmtime_str
 TimeValue, 627
get_sec
 TimeValue, 628
get_size
 OSCL_FastString, 172
 OSCL_HeapStringA, 192
 OSCL_String, 249
 OSCL_wFastString, 283
 OSCL_wHeapStringA, 288
 OSCL_wString, 293
 osclutil, 67, 68
get_str
 OSCL_FastString, 172
 OSCL_HeapStringA, 193
 OSCL_String, 250
 OSCL_wFastString, 283
 OSCL_wHeapStringA, 288
 OSCL_wString, 293
 osclutil, 68
get_str_ctime
 TimeValue, 628
get_timeval_ptr
 TimeValue, 628
get_uint64_lower32
 Oscl_Int64_Utils, 195
get_uint64_middle32
 Oscl_Int64_Utils, 195
get_uint64_upper32
 Oscl_Int64_Utils, 195
get_upper32
 NTPTime, 163
get_usec

TimeValue, 628
 get_value
 NTPTime, 163
 GetAcceptedSocket
 OsclAcceptMethod, 296
 GetAcceptedSocketL
 OsclTCPSocket, 547
 OsclTCPSocketI, 551
 getAllocatedSize
 OsclMemPoolResizableAllocator, 432
 getAuditRoot
 MM_Audit_Imp, 149
 GetAvailableBufferSize
 MediaData, 138
 getAvailableSize
 OsclMemPoolResizableAllocator, 432
 getBufferSize
 OsclMemPoolResizableAllocator, 432
 GetBufferState
 osclutil, 68
 getCapacity
 OsclRefCounterMemFrag, 478
 getCheckSum
 StrCSumPtrLen, 621
 getCount
 Oscl_DefAllocWithRefCounter, 168
 OsclRefCounter, 473
 OsclRefCounterDA, 476
 OsclRefCounterMemFrag, 478
 OsclRefCounterMTDA, 480
 OsclRefCounterMTSA, 482
 OsclRefCounterSA, 484
 GetElementType
 Oscl_FileFind, 183
 GetError
 Oscl_File, 176
 OsclNativeFile, 444
 GetErrorTrapImp
 OsclErrorTrap, 364
 GetFactories
 OsclRegistryAccessClient, 485
 OsclRegistryClientImpl, 493
 OsclRegistryServTlsImpl, 496
 GetFactory
 OsclRegistryAccessClient, 485
 OsclRegistryClientImpl, 493
 OsclRegistryServTlsImpl, 496
 GetFragment
 osclutil, 69
 getGlobalMemAuditObject
 OsclMemGlobalAuditObject, 421
 getHead
 OsclDoubleListBase, 357
 GetHostName

OsclDNS, 340
 OsclDNSI, 341
 OsclDNSIBase, 344
 OsclGetHostByNameMethod, 394
 GetHostByNameParam, 131
 GetHostByNameParam
 ~GetHostByNameParam, 131
 Create, 131
 Destroy, 131
 iAddr, 131
 iName, 131
 GetHostByNameSuccess
 OsclDNSI, 341
 OsclDNSIBase, 344
 GetId
 OsclExecSchedulerCommonBase, 383
 OsclThread, 554
 getInstance
 OsclSingletonRegistry, 515
 OsclTLSRegistry, 574
 OsclTLSRegistryEx, 575
 getLargestContiguousFreeBlockSize
 OsclMemPoolResizableAllocator, 432
 GetLastError
 Oscl_FileFind, 183
 getLeaveCode
 OsclException, 367
 GetLength
 BufFragGroup, 117
 GetLocalBufsize
 MediaData, 139
 GetLocalFragment
 MediaData, 139
 GetLock
 OsclMemAudit, 410
 getLoggerObject
 PVLogger, 595
 GetLogLevel
 PVLogger, 595
 GetMaxFrags
 BufFragGroup, 118
 GetMediaFragment
 MediaData, 139
 GetMediaSize
 MediaData, 139
 getMemFrag
 OsclRefCounterMemFrag, 478
 getMemFragPtr
 OsclRefCounterMemFrag, 478
 getMemFragSize
 OsclRefCounterMemFrag, 478
 getMemPoolBufferAllocatedSize
 OsclMemPoolResizableAllocator, 432
 getMemPoolBufferSize

OsclMemPoolResizableAllocator, 432
 GetName
 OsclExecSchedulerCommonBase, 383
 GetNext
 BufFragGroup, 118
 GetNumAppenders
 PVLogger, 595
 GetNumFrags
 BufFragGroup, 118
 GetNumMediaFrags
 MediaData, 139
 getOffset
 OsclDoubleListBase, 357
 GetParent
 PVLogger, 596
 GetPriority
 OsclThread, 555
 GetPVLoggerObject
 PVLoggerRegistry, 605
 GetPVLoggerRegistry
 PVLoggerRegistry, 605
 GetReadAsyncNumElements
 OsclNativeFile, 444
 GetRecvData
 OsclIPSocketI, 399
 OsclRecvFromMethod, 467
 OsclRecvFromRequest, 469
 OsclRecvMethod, 471
 OsclRecvRequest, 472
 OsclTCPSocket, 547
 OsclTCPSocketI, 551
 OsclUDPSocket, 581
 OsclUDPSocketI, 585
 GetRefCounter
 OsclSharedPtr, 509
 getRefCounter
 OsclRefCounterMemFrag, 478
 GetRep
 OsclSharedPtr, 509
 GetScheduler
 OsclExecSchedulerCommonBase, 383
 GetSendData
 OsclIPSocketI, 399
 OsclSendMethod, 504
 OsclSendRequest, 505
 OsclSendToMethod, 506
 OsclSendToRequest, 507
 OsclTCPSocket, 547
 OsclTCPSocketI, 551
 OsclUDPSocket, 581
 OsclUDPSocketI, 585
 GetShutdown
 OsclSocketIBase, 523
 getSize
 MM_Audit_Imp, 149
 GetSocketError
 OsclDNSRequestAO, 352
 OsclSocketRequestAO, 532
 getTagActualSize
 MM_Audit_Imp, 149
 GetTimestamp
 MediaData, 139
 good
 OsclBinStream, 326
 GOOD_STATE
 OsclBinStream, 325
 Handle
 Oscl_File, 177
 OsclFileHandle, 391
 HandleDNSEvent
 OsclDNSObserver, 349
 HandleSocketEvent
 OsclSocketObserver, 529
 HasAsyncBind
 OsclSocketIBase, 523
 HasAsyncListen
 OsclSocketIBase, 523
 HasAsyncRead
 OsclNativeFile, 444
 hash
 OSCL_String, 250
 OSCL_wString, 293
 HasThisOffset
 OsclAsyncFileBuffer, 308
 HaveRoomInCurrentBlock
 OsclBinStream, 326
 Head
 OsclDoubleList, 355
 OsclPriorityList, 450
 head
 Oscl_Linked_List_Base, 204
 HeapBase, 132
 HeapBase, 133
 HeapBase
 ~HeapBase, 133
 HeapBase, 133
 host_to_big_endian
 osclbase, 33
 host_to_little_endian
 osclbase, 34
 iActive
 OsclDNSRequest, 350
 iAddedNum
 PVActiveBase, 591
 iAddr
 BindParam, 112

ConnectParam, 127
 GetHostByNameParam, 131
 RecvFromParam, 612
 SendToParam, 616
iAddress
 OsclIPSocketI, 400
iAlloc
 OsclDNSIBase, 344
 OsclDNSMethod, 348
 OsclExecSchedulerCommonBase, 387
 OsclIPSocketI, 400
 OsclSocketIBase, 525
 OsclSocketServIBase, 540
iAllocatedSz
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 436
iAOPriority
 TReadyQueLink, 632
iAsyncReadBufferSize
 OsclNativeFileParams, 446
iBlankSocket
 AcceptParam, 108
iBlockBuffer
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 435
iBlockInfoAlignedSize
 OsclMemPoolResizableAllocator, 434
iBlockingMode
 OsclExecSchedulerCommonBase, 387
iBlockPostFence
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 435
iBlockPreFence
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 435
iBlockSize
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 435
iBuffer
 OsclBuf, 329
iBufferInfoAlignedSize
 OsclMemPoolResizableAllocator, 434
iBufferPostFence
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 436
iBufferPreFence
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 436
iBufferSize
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 436
iBufRecv
 RecvFromParam, 612
 RecvParam, 614
iBufSend
 SendParam, 615
 SendToParam, 616
iBusy
 PVActiveBase, 591
iCancel
 OsclSocketServRequestQElem, 543
iCBase
 OsclTrapStackItem, 578
iCheckFreeMemoryAvailable
 OsclMemPoolResizableAllocator, 434
iCheckNextAvailable
 OsclMemPoolResizableAllocator, 434
iCheckNextAvailableFreeChunk
 OsclMemPoolFixedChunkAllocator, 427
iChunkSize
 OsclMemPoolFixedChunkAllocator, 427
iChunkSizeMemAligned
 OsclMemPoolFixedChunkAllocator, 427
iComponentId
 OsclComponentRegistryElement, 334
iComponentIdCounter
 OsclComponentRegistry, 332
iContainer
 OsclSocketMethod, 528
 OsclSocketRequestAO, 534
Id
 OsclAsyncFileBuffer, 308
 OsclSocketRequestAO, 533
 PVThreadContext, 610
iData
 OsclComponentRegistry, 332
iDebugLogger
 OsclExecSchedulerCommonBase, 387
iDefAlloc
 OsclExecSchedulerCommonBase, 387
iDelta
 OsclExecSchedulerCommonBase, 387
iDNSFxn
 OsclDNSMethod, 348
iDNSI
 OsclDNSRequestAO, 353
iDNSMethod
 OsclDNSRequestAO, 353
iDNSObserver
 OsclDNSMethod, 348
iDNSRequest
 DNSRequestParam, 130
iDNSRequestAO
 OsclDNSMethod, 348
 OsclDNSRequest, 350
iDNSRequestParam
 OsclDNSRequest, 350
iDoStop

OsclExecSchedulerCommonBase, 387
iDoSuspend
 OsclExecSchedulerCommonBase, 387
iEnableNullPtrReturn
 OsclMemPoolFixedChunkAllocator, 427
 OsclMemPoolResizableAllocator, 434
iEndAddr
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 436
iErrAlloc
 OsclSelect, 501
iErrorTrapImp
 OsclExecSchedulerCommonBase, 387
iExecTimerQ
 OsclExecSchedulerCommonBase, 387
iExpectedNumBlocksPerBuffer
 OsclMemPoolResizableAllocator, 434
iFactory
 OsclComponentRegistryElement, 334
 OsclRegistryAccessElement, 489
iFlags
 RecvFromParam, 612
 RecvParam, 614
 SendParam, 615
 SendToParam, 616
iFreeMemChunkList
 OsclMemPoolFixedChunkAllocator, 427
iFreeMemContextData
 OsclMemPoolResizableAllocator, 434
iFreeMemPoolObserver
 OsclMemPoolResizableAllocator, 434
ifront
 Oscl_Queue_Base, 229
iFxN
 DNSRequestParam, 130
 SocketRequestParam, 619
iGrandTotalTicks
 OsclExecSchedulerCommonBase, 387
iHead
 OsclDoubleListBase, 357
 OsclDoubleRunner, 358
iHeapCheck
 OsclSelect, 501
iHigh
 OsclInteger64Transport, 397
iHow
 ShutdownParam, 617
iId
 OsclComponentRegistryElement, 334
 OsclDNSMethod, 348
 OsclIPSocketI, 400
iIsIn
 TReadyQueLink, 632
iJumpData
 OsclErrorTrapImp, 366
iLeave
 OsclErrorTrapImp, 366
iLen
 PVSockBufRecv, 608
 PVSockBufSend, 609
iLength
 OsclBuf, 329
iLogger
 OsclDNSMethod, 348
 OsclDNSRequestAO, 353
 OsclExecSchedulerCommonBase, 387
 OsclIPSocketI, 400
 OsclSocketServIBase, 540
iLogPerfIndentStr
 OsclExecSchedulerCommonBase, 387
iLogPerfIndentStrLen
 OsclExecSchedulerCommonBase, 387
iLogPerfTotal
 OsclExecSchedulerCommonBase, 387
iLow
 OsclInteger64Transport, 397
iMaxLen
 PVSockBufRecv, 608
iMaxLength
 OsclBuf, 329
iMaxNewMemPoolBufferSz
 OsclMemPoolResizableAllocator, 434
iMemPool
 OsclMemPoolFixedChunkAllocator, 427
iMemPoolAllocator
 OsclMemPoolFixedChunkAllocator, 427
iMemPoolBufferAllocator
 OsclMemPoolResizableAllocator, 434
iMemPoolBufferList
 OsclMemPoolResizableAllocator, 434
iMemPoolBufferNumLimit
 OsclMemPoolResizableAllocator, 434
iMemPoolBufferSize
 OsclMemPoolResizableAllocator, 434
iMimeType
 OsclRegistryAccessElement, 489
iMultiMaxLen
 RecvFromParam, 612
iMutex
 OsclComponentRegistry, 332
iName
 GetHostByNameParam, 131
 OsclExecSchedulerCommonBase, 387
 PVActiveBase, 591
iNativeAccessMode
 OsclNativeFileParams, 446
iNativeBufferSize
 OsclNativeFileParams, 446

iNativeMode
 OsclExecSchedulerCommonBase, 387

IncLogPerf
 OsclExecSchedulerCommonBase, 384

increment_refcnt
 BufferState, 115

iNext
 OsclDoubleLink, 354
 OsclDoubleRunner, 358
 OsclTrapStackItem, 578

iNextAvailableContextData
 OsclMemPoolFixedChunkAllocator, 427
 OsclMemPoolResizableAllocator, 434

iNextFreeBlock
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 435
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 436

Init
 OsclErrorTrap, 364
 OsclInit, 396
 OsclMem, 405
 OsclScheduler, 497
 PVLogger, 596

InitExecQ
 OsclExecSchedulerCommonBase, 384

Insert
 OsclDoubleListBase, 357
 OsclPriorityList, 450

insert
 Oscl_Map, 209
 Oscl_TagTree, 260
 Oscl_Vector, 276
 Oscl_Vector_Base, 280

insert_unique
 Oscl_Rb_Tree, 232

InsertAfter
 OsclDoubleLink, 354

InsertBefore
 OsclDoubleLink, 354

InsertHead
 OsclDoubleList, 355
 OsclDoubleListBase, 357

InsertTail
 OsclDoubleList, 355
 OsclDoubleListBase, 357

InstallScheduler
 OsclExecSchedulerCommonBase, 384

INT64
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816

int64
 osclbase, 32

INT64_HILO

osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816

INTERNAL_ERROR
 BuffFragStatusClass, 119

internalLeave, 134
 osclerror, 84

internalLeave
 a, 134

InThread
 DNSRequestParam, 129

iNumAOAdded
 OsclExecSchedulerCommonBase, 387

iNumChunk
 OsclMemPoolFixedChunkAllocator, 427

iNumOfRun
 OsclAsyncFile, 306

iNumOfRunErr
 OsclAsyncFile, 306

iNumOutstanding
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 436

iNumSessions
 OsclComponentRegistry, 332

INVALID_ACCESS_ERROR
 OsclProcStatus, 457

INVALID_ARGUMENT_ERROR
 OsclProcStatus, 457

INVALID_FUNCTION_ERROR
 OsclProcStatus, 457

INVALID_HANDLE_ERROR
 OsclProcStatus, 457

INVALID_ID
 BuffFragStatusClass, 119

INVALID_OPERATION_ERROR
 OsclProcStatus, 457

INVALID_PARAM_ERROR
 OsclProcStatus, 456

INVALID_POINTER_ERROR
 OsclProcStatus, 457

INVALID_PRIORITY_ERROR
 OsclProcStatus, 456

INVALID_THREAD_ERROR
 OsclProcStatus, 456

INVALID_THREAD_ID_ERROR
 OsclProcStatus, 456

INVALID_TYPE
 Oscl_FileFind, 181

iObserver
 OsclIPSocketI, 400
 OsclMemPoolFixedChunkAllocator, 427
 OsclMemPoolResizableAllocator, 434

iOffset
 OsclDoubleListBase, 357
 OsclDoubleRunner, 358

iOpCount
 OsclFileStatsItem, 393

iOsclBase
 OsclSelect, 501

iOsclErrorTrap
 OsclSelect, 501

iOsclLogger
 OsclSelect, 501

iOsclMemory
 OsclSelect, 501

iOsclScheduler
 OsclSelect, 501

iOtherExecStats
 OsclExecSchedulerCommonBase, 387

iOutputFile
 OsclSelect, 501

iPacketLen
 RecvFromParam, 612

iPacketSource
 RecvFromParam, 612

ipAddr
 OsclNetworkAddress, 447

iParam
 OsclFileStatsItem, 393
 OsclSocketRequest, 530
 OsclSocketRequestAO, 534

iParam2
 OsclFileStatsItem, 393

iParamSize
 OsclSocketRequestAO, 534

iParentBuffer
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 435

iPrev
 OsclDoubleLink, 354

iPrevFreeBlock
 OsclMemPoolResizableAllocator::Mem-
 PoolBlockInfo, 435

iPriority
 OsclPriorityLink, 449

iPtr
 PVSockBufRecv, 608
 PVSockBufSend, 609

iPVActiveStats
 PVActiveBase, 591

iPVRreadyQLink
 PVActiveBase, 591

iPVStatQ
 OsclExecSchedulerCommonBase, 387

iPVStats
 OsclExecSchedulerCommonBase, 387

iQSize
 ListenParam, 136

iReadyQ

OsclExecSchedulerCommonBase, 387

irear
 Oscl_Queue_Base, 229

iRefCount
 DNSRequestParam, 130
 OsclMemPoolFixedChunkAllocator, 427
 OsclMemPoolResizableAllocator, 434

iRequestedAvailableFreeMemSize
 OsclMemPoolResizableAllocator, 434

iRequestedNextAvailableSize
 OsclMemPoolResizableAllocator, 434

iResumeSem
 OsclExecSchedulerCommonBase, 387

is_writable
 OSCL_String, 250
 OSCL_wString, 294

is_zero
 TimeValue, 628

IsActive
 PVLogger, 596

IsAdded
 PVActiveBase, 589

isAllocNodePtr
 MM_AllocBlockHdr, 143

IsBusy
 OsclActiveObject, 300
 OsclTimerObject, 566

iSchedulerAlloc
 OsclSelect, 501

iSchedulerName
 OsclSelect, 501

iSchedulerReserve
 OsclSelect, 501

isCIEquivalentTo
 StrCSumPtrLen, 621
 StrPtrLen, 624
 WStrPtrLen, 634

isCIPrefixOf
 StrPtrLen, 624

iSelect
 OsclSocketServRequestQElem, 543

IsEmpty
 OsclDoubleListBase, 357

iSeqNum
 TReadyQueLink, 632

iServerError
 OsclSocketServIBase, 540

iServState
 OsclSocketServIBase, 540

IsHead
 OsclDoubleList, 355
 OsclPriorityList, 450

IsIn
 OsclReadyQ, 466

OsclTimerQ, 569
 IsInAnyQ
 PVActiveBase, 590
 IsInstalled
 OsclExecSchedulerCommonBase, 384
 IsInUse
 OsclAsyncFileBuffer, 308
 isLetter
 StrPtrLen, 624
 IsLocalData
 MediaData, 139
 iSocket
 OsclIPSocketI, 400
 iSocketError
 OsclIDNSRequestAO, 353
 OsclSocketRequestAO, 534
 iSocketFxn
 OsclSocketMethod, 528
 iSocketI
 OsclSocketRequest, 530
 iSocketRequest
 OsclSocketServRequestQElem, 543
 iSocketRequestAO
 OsclSocketMethod, 528
 OsclSocketRequest, 530
 iSocketServ
 OsclDNSIBase, 344
 OsclIPSocketI, 400
 OsclSocketIBase, 525
 IsOpen
 OsclSocketIBase, 523
 IsReady
 OsclDNSIBase, 344
 IsSameThreadContext
 PVThreadContext, 610
 IsServConnected
 OsclSocketServIBase, 540
 IsServerThread
 OsclSocketServI, 538
 isSetFailure
 MM_Audit_Imp, 150
 IsStarted
 OsclExecSchedulerCommonBase, 384
 IsTail
 OsclDoubleList, 355
 OsclPriorityList, 450
 iStartAddr
 OsclMemPoolResizableAllocator::Mem-
 PoolBufferInfo, 436
 iStartTick
 OsclFileStatsItem, 393
 iStatus
 PVActiveBase, 591
 iStopper
 OsclExecSchedulerCommonBase, 387
 iStopperCrit
 OsclExecSchedulerCommonBase, 387
 iSuspended
 OsclExecSchedulerCommonBase, 387
 IsValid
 OsclAsyncFileBuffer, 308
 iTAny
 OsclTrapStackItem, 578
 iterator
 Oscl_Linked_List_Base, 204
 Oscl_Map, 207
 Oscl_Rb_Tree, 232
 Oscl_Rb_Tree_Iterator, 239
 Oscl_TagTree::iterator, 265
 Oscl_Vector, 274
 OsclPriorityQueue, 452
 iThreadContext
 OsclExecSchedulerCommonBase, 387
 PVActiveBase, 591
 iTime
 OsclExecSchedulerCommonBase, 387
 iTimeCompareThreshold
 OsclExecSchedulerCommonBase, 387
 iTimeQueuedTicks
 TReadyQueLink, 632
 iTimeToRunTicks
 TReadyQueLink, 632
 iTotalPercent
 OsclExecSchedulerCommonBase, 387
 iTotalTicks
 OsclFileStatsItem, 393
 iTotalTicksTemp
 OsclExecSchedulerCommonBase, 387
 iTrapOperation
 OsclTrapStackItem, 578
 iTrapStack
 OsclErrorTrapImp, 366
 iVec
 OsclComponentRegistryData, 333
 iXferLen
 SendParam, 615
 SendToParam, 616
 Join
 OsclIPSocketI, 399
 OsclSocketI, 517
 OsclSocketIBase, 523
 OsclUDPSocket, 581
 Jump
 OsclJump, 401
 key_comp
 Oscl_Map, 210

key_compare
 Oscl_Map, 207
 key_type
 Oscl_Map, 207
 Oscl_Rb_Tree, 232

 Leave
 OsclError, 360
 LeaveIfError
 OsclError, 360
 LeaveIfNull
 OsclError, 360
 Left
 OsclPtrC, 461
 left
 Oscl_Rb_Tree_Node_Base, 243
 len
 OsclMemoryFragment, 422
 StrPtrLen, 624
 WStrPtrLen, 634
 Length
 OsclAsyncFileBuffer, 308
 OsclBuf, 329
 OsclPtr, 458
 OsclPtrC, 461
 length
 BufFragGroup, 118
 OsclBinStream, 327
 StrPtrLen, 624
 WStrPtrLen, 634
 lineNo
 MM_AllocInfo, 145
 MM_AllocQueryInfo, 147
 link_type
 Oscl_Rb_Tree, 232
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 Oscl_Rb_Tree_Node, 241
 LinkedListElement, 135
 LinkedListElement, 135
 LinkedListElement
 data, 135
 LinkedListElement, 135
 next, 135
 Listen
 OsclListenMethod, 402
 OsclListenRequest, 403
 OsclSocketI, 517
 OsclSocketIBase, 523
 OsclTCPSocket, 548
 OsclTCPSocketI, 551
 ListenAsync
 OsclSocketIBase, 523
 OsclTCPSocket, 548

 OsclTCPSocketI, 552
 ListenParam, 136
 ListenParam, 136
 ListenParam
 iQSize, 136
 ListenParam, 136
 ListenRequest
 OsclListenMethod, 402
 little_endian_to_host
 osclbase, 34
 localbuf
 MediaData, 139
 Lock
 OsclLockBase, 404
 OsclMutex, 441
 OsclNullLock, 448
 OsclThreadLock, 557
 lockAndGetInstance
 OsclSingletonRegistry, 515
 Log
 OsclFileStats, 392
 log_level_type
 AllPassFilter, 110
 PVLogger, 594
 PVLoggerFilter, 600
 PVLoggerRegistry, 604
 LogAll
 OsclFileStats, 392
 Logger
 OsclSocketI, 517
 LogMsgBuffers
 PVLogger, 596
 LogMsgBuffersV
 PVLogger, 596
 LogMsgString
 PVLogger, 597
 LogMsgStringV
 PVLogger, 597
 LoopbackSocket
 OsclSocketServI, 538
 lower_bound
 Oscl_Map, 210
 Oscl_Rb_Tree, 232

 MakeAddr
 OsclSocketI, 518
 makeValidTag
 MM_Audit_Imp, 150
 map_type
 Oscl_TagTree, 258
 mapit
 Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
 mapiter

Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
Match
 OsclComponentRegistryElement, 334
max_size
 Oscl_Map, 210
 Oscl_Rb_Tree, 232
MAX_THRDS_REACHED_ERROR
 OsclProcStatus, 456
maximum
 Oscl_Rb_Tree_Node_Base, 243
MaxLen
 OsclNameString, 442
maxsize
 CFastRep, 124
 CHeapRep, 126
 CStackRep, 128
mbchar
 osclbase, 32
MediaData, 137
 MediaData, 138
MediaData
 ~MediaData, 138
 AddLocalFragment, 138
 available_localbuf, 139
 Clear, 138
 GetAvailableBufferSize, 138
 GetLocalBufsize, 139
 GetLocalFragment, 139
 GetMediaFragment, 139
 GetMediaSize, 139
 GetNumMediaFrags, 139
 GetTimestamp, 139
 IsLocalData, 139
 localbuf, 139
 MediaData, 138
 num_reserved_fragments, 139
 SetTimestamp, 139
 timestamp, 139
MediaStatusClass, 140
MediaTimestamp
 osclutil, 66
MEM_ALIGN_SIZE
 osclmemory, 49
MemAllocator, 141
MemAllocator
 ~MemAllocator, 141
 allocate, 141
 deallocate, 141
 pointer, 141
memoryPoolBufferMgmtOverhead
 OsclMemPoolResizableAllocator, 432
message_id_type
 AllPassFilter, 110

 PVLogger, 594
 PVLoggerAppender, 599
 PVLoggerFilter, 600
 PVLoggerLayout, 602
MethodDone
 OsclDNSMethod, 347
 OsclSocketMethod, 527
MICROSECONDS
 osclbase, 33
MILLISECONDS
 osclbase, 33
MIN_FENCE_SIZE
 osclmemory, 49
minimum
 Oscl_Rb_Tree_Node_Base, 243
MM_AddTag
 MM_Audit_Imp, 150
 OsclMemAudit, 410
MM_ALLOC_MAX_QUERY_FILENAME_LEN
 osclmemory, 49
MM_ALLOC_MAX_QUERY_TAG_LEN
 osclmemory, 49
MM_allocate
 MM_Audit_Imp, 150
 OsclMemAudit, 410
MM_AllocBlockFence, 142
 MM_AllocBlockFence, 142
MM_AllocBlockFence
 check_fence, 142
 fill_fence, 142
 MM_AllocBlockFence, 142
 pad, 142
MM_AllocBlockHdr, 143
 MM_AllocBlockHdr, 143
MM_AllocBlockHdr
 isAllocNodePtr, 143
 MM_AllocBlockHdr, 143
 pad, 143
 pNode, 143
 pRootNode, 143
 setAllocNodeFlag, 143
 size, 143
MM_AllocInfo, 144
 MM_AllocInfo, 145
MM_AllocInfo
 ~MM_AllocInfo, 145
 allocNum, 145
 bSetFailure, 145
 lineNo, 145
 MM_AllocInfo, 145
 operator delete, 145
 operator new, 145
 pFileName, 145

pMemBlock, 145
 pStatsNode, 145
 size, 145
MM_AllocNode, 146
 MM_AllocNode, 146
MM_AllocNode
 ~MM_AllocNode, 146
 MM_AllocNode, 146
 operator delete, 146
 operator new, 146
 pAllocInfo, 146
 pNext, 146
 pPrev, 146
MM_AllocNodeAutoPtr
 osclmemory, 56
MM_AllocQueryInfo, 147
MM_AllocQueryInfo
 allocNum, 147
 fileName, 147
 lineNo, 147
 pMemBlock, 147
 size, 147
 tag, 147
MM_AUDIT_ALLOC_NODE_ENABLE_-FLAG
 osclmemory, 49
MM_AUDIT_ALLOC_NODE_SUPPORT
 osclmemory, 49
MM_AUDIT_FAILURE_SIMULATION_-SUPPORT
 osclmemory, 49
MM_AUDIT_FENCE_SUPPORT
 osclmemory, 49
MM_AUDIT_FILL_SUPPORT
 osclmemory, 49
MM_Audit_Imp, 148
 ~MM_Audit_Imp, 149
 addAllocNode, 149
 createStatsNode, 149
 getAuditRoot, 149
 getSize, 149
 getTagActualSize, 149
 isSetFailure, 150
 makeValidTag, 150
MM_AddTag, 150
MM_allocate, 150
MM_Audit_Imp, 149
MM_CreateAllocNodeInfo, 150
MM_deallocate, 150
MM_GetAllocNo, 150
MM_GetAllocNodeInfo, 151
MM_GetExistingTag, 151
MM_GetMode, 151
MM_GetNumAllocNodes, 151

 MM_GetOverheadStats, 151
 MM_GetPostfillPattern, 151
 MM_GetPrefillPattern, 151
 MM_GetRootNode, 152
 MM_GetStats, 152
 MM_GetStatsInDepth, 152
 MM_GetTagName, 152
 MM_GetTreeNodes, 152
 MM_ReleaseAllocNodeInfo, 152
 MM_SetFailurePoint, 153
 MM_SetMode, 153
 MM_SetPostfillPattern, 153
 MM_SetPrefillPattern, 153
 MM_SetTagLevel, 153
 MM_UnsetFailurePoint, 153
 MM_Validate, 153
 pruneSubtree, 154
 removeALLAllocNodes, 154
 removeAllocNode, 154
 retrieveParentTag, 154
 retrieveParentTagLength, 154
 updateStatsNode, 154
 updateStatsNodeInFailure, 154
 validate, 154
 validate_all_heap, 154
MM_AUDIT_INCLUDE_ALL_HEAP_-VALIDATION
 osclmemory, 49
MM_AUDIT_POSTFILL_FLAG
 osclmemory, 49
MM_AUDIT_PREFILL_FLAG
 osclmemory, 49
MM_AUDIT_SUPPRESS_FILENAME_FLAG
 osclmemory, 49
MM_AUDIT_VALIDATE_ALL_HEAP_FLAG
 osclmemory, 49
MM_AUDIT_VALIDATE_BLOCK
 osclmemory, 49
MM_AUDIT_VALIDATE_ON_FREE_FLAG
 osclmemory, 49
MM_AuditOverheadStats, 156
MM_AuditOverheadStats
 per_allocation_overhead, 156
 stats_overhead, 156
MM_CreateAllocNodeInfo
 MM_Audit_Imp, 150
 OsclMemAudit, 410
MM_deallocate
 MM_Audit_Imp, 150
 OsclMemAudit, 410
MM_FailInsertParam, 157
 MM_FailInsertParam, 157
MM_FailInsertParam
 MM_FailInsertParam, 157

nAllocNum, 157
 operator delete, 157
 operator new, 157
 reset, 157
 xsubi, 157
MM_GetAllocNo
 MM_Audit_Imp, 150
 OsclMemAudit, 410
MM_GetAllocNodeInfo
 MM_Audit_Imp, 151
 OsclMemAudit, 410
MM_GetExistingTag
 MM_Audit_Imp, 151
 OsclMemAudit, 411
MM_GetMode
 MM_Audit_Imp, 151
 OsclMemAudit, 411
MM_GetNumAllocNodes
 MM_Audit_Imp, 151
 OsclMemAudit, 411
MM_GetOverheadStats
 MM_Audit_Imp, 151
 OsclMemAudit, 411
MM_GetPostfillPattern
 MM_Audit_Imp, 151
 OsclMemAudit, 411
MM_GetPrefillPattern
 MM_Audit_Imp, 151
 OsclMemAudit, 411
MM_GetRefCount
 OsclMemAudit, 411
MM_GetRootNode
 MM_Audit_Imp, 152
 OsclMemAudit, 412
MM_GetStats
 MM_Audit_Imp, 152
 OsclMemAudit, 412
MM_GetStatsInDepth
 MM_Audit_Imp, 152
 OsclMemAudit, 412
MM_GetTagName
 MM_Audit_Imp, 152
 OsclMemAudit, 412
MM_GetTreeNodes
 MM_Audit_Imp, 152
 OsclMemAudit, 412
MM_ReleaseAllocNodeInfo
 MM_Audit_Imp, 152
 OsclMemAudit, 412
MM_SetFailurePoint
 MM_Audit_Imp, 153
 OsclMemAudit, 412
MM_SetMode
 MM_Audit_Imp, 153

OsclMemAudit, 413
MM_SetPostfillPattern
 MM_Audit_Imp, 153
 OsclMemAudit, 413
MM_SetPrefillPattern
 MM_Audit_Imp, 153
 OsclMemAudit, 413
MM_SetTagLevel
 MM_Audit_Imp, 153
 OsclMemAudit, 413
MM_Stats_CB, 158
 MM_Stats_CB, 158
 num_child_nodes, 158
 operator delete, 158
 operator new, 158
 pStats, 158
 tag, 158
MM_Stats_t, 159
 MM_Stats_t, 160
 numAllocFails, 160
 numAllocs, 160
 numBytes, 160
 operator delete, 160
 operator new, 160
 peakNumAllocs, 160
 peakNumBytes, 160
 reset, 160
 totalNumAllocs, 160
 totalNumBytes, 160
 update, 160
MM_StatsNodeTagTreeType
 osclmemory, 56
MM_UnsetFailurePoint
 MM_Audit_Imp, 153
 OsclMemAudit, 413
MM_Validate
 MM_Audit_Imp, 153
 OsclMemAudit, 413
MMAuditCharAutoPtr
 osclmemory, 56
MMAuditUint8AutoPtr
 osclmemory, 56
Mode
 OsclNativeFile, 444
mode
 oscl_stat_buf, 247
MODE_APPEND
 Oscl_File, 175
MODE_BINARY
 Oscl_File, 175
MODE_READ
 Oscl_File, 175
MODE_READ_PLUS
 Oscl_File, 175

MODE_READWRITE
 Oscl_File, 175
 MODE_TEXT
 Oscl_File, 175
 mode_type
 Oscl_File, 175
 move_to_end
 Oscl_Linked_List, 199
 Oscl_Linked_List_Base, 203
 Oscl_MTLinkedList, 215
 move_to_front
 Oscl_Linked_List, 199
 Oscl_Linked_List_Base, 203
 Oscl_MTLinkedList, 216
 MSEC_PER_SEC
 osclbase, 43
 MSEC_TO_MICROSEC
 oscl_socket_method.h, 735
 MsecToTicks
 OsclTickCount, 558
 MUTEX_LOCKED_ERROR
 OsclProcStatus, 457

 nAllocNum
 MM_FailInsertParam, 157
 New
 Oscl_DefAllocWithRefCounter, 169
 NewL
 OsclAcceptMethod, 296
 OsclAsyncFile, 305
 OsclAsyncFileBuffer, 308
 OsclBindMethod, 310
 OsclBuf, 329
 OsclConnectMethod, 336
 OsclDNS, 340
 OsclIDNSI, 342
 OsclGetHostByNameMethod, 394
 OsclListenMethod, 402
 OsclRecvFromMethod, 467
 OsclRecvMethod, 471
 OsclSendMethod, 504
 OsclSendToMethod, 506
 OsclShutdownMethod, 511
 OsclSocketI, 518
 OsclSocketServ, 536
 OsclSocketServI, 538
 OsclTCPSocket, 548
 OsclTCPSocketI, 552
 OsclUDPSocket, 581
 OsclUDPSocketI, 585

 NewRequest
 OsclIDNSRequestAO, 352
 OsclSocketRequestAO, 533
 next

 BuffFragGroup, 118
 LinkedListElement, 135
 nextFragPtr
 OsclBinStream, 327
 NO_PERMISSION_ERROR
 OsclProcStatus, 456
 Node
 Oscl_TagTree::Node, 268
 node
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 node_ptr
 Oscl_TagTree, 258
 node_type
 Oscl_TagTree, 258
 NOT_ENOUGH_MEMORY_ERROR
 OsclProcStatus, 456
 NOT_ENOUGH_RESOURCES_ERROR
 OsclProcStatus, 456
 NOT_ENOUGH_SPACE
 BuffFragStatusClass, 119
 NOT_IMPLEMENTED
 OsclProcStatus, 457
 NOT_SUSPENDED_ERROR
 OsclProcStatus, 456
 notifyfreeblockavailable
 OsclMemPoolResizableAllocator, 432
 notifyfreechunkavailable
 OsclMemPoolFixedChunkAllocator, 426
 notifyfreememoryavailable
 OsclMemPoolResizableAllocator, 432
 NTPTime, 161
 get_lower32, 163
 get_middle32, 163
 get_upper32, 163
 get_value, 163
 NTPTime, 162, 163
 operator+=, 163
 operator-, 163
 operator=, 163, 164
 set_from_system_time, 164
 set_to_current_time, 164
 TimeValue, 630
 to_system_time, 164
 NULL
 osclbase, 30
 NULL_INPUT
 BuffFragStatusClass, 119
 NULL_TERM_CHAR
 osclbase, 30
 num_child_nodes
 MM_Stats_CB, 158
 num_elements
 Oscl_Linked_List_Base, 204

num_fragments
 BufFragGroup, 118
 num_reserved_fragments
 MediaData, 139
 numAllocFails
 MM_Stats_t, 160
 numAllocs
 MM_Stats_t, 160
 numBytes
 MM_Stats_t, 160
 numelems
 Oscl_Queue_Base, 229
 Oscl_Vector_Base, 281
 numFrags
 OsclBinStream, 327

 octet
 osclbase, 32
 Offset
 OsclAsyncFileBuffer, 308
 Open
 Oscl_File, 177
 OsclAsyncFile, 305, 306
 OsclDNSI, 342
 OsclDNSIBase, 344
 OsclFileCache, 390
 OsclNativeFile, 444
 OsclSocketI, 518
 OsclSocketIBase, 524
 OsclSocketServRequestList, 541
 OpenSession
 OsclComponentRegistry, 332
 operator *
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
 OsclExclusiveArrayPtr, 369
 OsclExclusivePtr, 372
 OsclExclusivePtrA, 375
 OSCLMemAutoPtr, 417
 OsclSharedPtr, 509
 OsclSingleton, 513
 OsclTLS, 570
 OsclTLSEx, 572
 operator *=
 TimeValue, 629
 operator delete
 MM_AllocInfo, 145
 MM_AllocNode, 146
 MM_FailInsertParam, 157
 MM_Stats_CB, 158
 MM_Stats_t, 160
 oscl_mem.h, 689

 OsclErrorAllocator, 363
 osclmemory, 57
 OsclMemStatsNode, 439
 operator delete[]
 osclmemory, 57
 operator new
 MM_AllocInfo, 145
 MM_AllocNode, 146
 MM_FailInsertParam, 157
 MM_Stats_CB, 158
 MM_Stats_t, 160
 oscl_mem.h, 689
 osclconfig_global_placement_new.h, 784
 OsclErrorAllocator, 363
 osclmemory, 57
 OsclMemStatsNode, 439
 operator new[]
 osclmemory, 57
 operator T *
 OsclDoubleRunner, 358
 operator TheClass *
 OsclSharedPtr, 510
 operator!=
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 OSCL_String, 250
 Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
 OSCL_wString, 294
 OsclAOStatus, 303
 OsclUuid, 587
 StrCSumPtrLen, 621
 StrPtrLen, 624
 TimeValue, 630
 WStrPtrLen, 634
 operator()
 Oscl_Less, 196
 Oscl_Map::value_compare, 212
 Oscl_Select1st, 244
 Oscl_Tag_Base, 256
 operator++
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
 OsclDoubleRunner, 358
 operator+=
 NTPTime, 163
 OSCL_String, 250
 OSCL_wString, 294
 TimeValue, 629
 operator-
 NTPTime, 163
 osclbase, 34

operator–
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
 OsclDoubleRunner, 358
 operator-=
 TimeValue, 629
 operator->
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
 OsclExclusiveArrayPtr, 369
 OsclExclusivePtr, 372
 OsclExclusivePtrA, 375
 OSCLMemAutoPtr, 417
 OsclSharedPtr, 510
 OsclSingleton, 513
 OscITLS, 570
 OscITLSEx, 572
 operator<
 OSCL_String, 250
 Oscl_Tag, 253
 OSCL_wString, 294
 OsclAOStatus, 303
 TimeValue, 630
 operator<<
 OsclBinOStreamBigEndian, 321
 OsclBinOStreamLittleEndian, 323
 operator<=
 OSCL_String, 250
 OSCL_wString, 294
 OsclAOStatus, 303
 TimeValue, 630
 operator=–
 NTPTime, 163, 164
 OSCL_FastString, 172
 OSCL_HeapStringA, 193
 Oscl_Map, 210
 Oscl_Rb_Tree, 232
 OSCL_String, 250, 251
 Oscl_TagTree, 260
 Oscl_Vector, 276
 OSCL_wFastString, 283
 OSCL_wHeapStringA, 288, 289
 OSCL_wString, 294
 OsclAOStatus, 303
 OsclComponentRegistryElement, 334
 OsclExclusiveArrayPtr, 369
 OsclExclusivePtr, 372
 OsclExclusivePtrA, 375
 OSCLMemAutoPtr, 417
 OsclRefCounterMemFrag, 478
 OsclSharedPtr, 510
 osclutil, 69, 70
 OsclUuid, 587
 StrCSumPtrLen, 621
 StrPtrLen, 624
 TimeValue, 629
 WStrPtrLen, 634
 operator==
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 OSCL_String, 251
 Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
 OSCL_wString, 294
 OsclAOStatus, 303
 osclbase, 34
 OsclNetworkAddress, 447
 OsclUuid, 587
 StrCSumPtrLen, 621
 StrPtrLen, 624
 TimeValue, 630
 WStrPtrLen, 634
 operator>
 OSCL_String, 251
 OSCL_wString, 294
 OsclAOStatus, 303
 TimeValue, 630
 operator>=
 OSCL_String, 251
 OSCL_wString, 294
 OsclAOStatus, 303
 TimeValue, 630
 operator>>
 OsclBinIStreamBigEndian, 315
 OsclBinIStreamLittleEndian, 318
 operator[]
 Oscl_Map, 210
 OSCL_String, 251
 Oscl_TagTree, 260
 Oscl_Vector, 276
 OSCL_wString, 294
 OSCL Base, 23
 OSCL config, 19
 OSCL Error, 81
 OSCL Init, 103
 OSCL IO, 91
 OSCL Memory, 44
 OSCL Proc, 99
 OSCL Util, 60
 OSCL_ABS
 osclbase, 30
 oscl_abs
 osclutil, 70
 OSCL_AF_INET

osclconfig_io.h, 787
Oscl_Alloc, 165
 allocate, 165
 allocate_fl, 165
OSCL_ALLOC_DELETE
 osclmemory, 49
OSCL_ALLOC_NEW
 osclmemory, 50
oscl_aostatus.h, 635
OSCL_ARRAY_DELETE
 osclmemory, 50
OSCL_ARRAY_NEW
 osclmemory, 50
OSCL_ASCII_CASE_MAGIC_BIT
 osclutil, 80
oscl_asin
 osclutil, 70
OSCL_ASSERT
 osclbase, 30
OSCL_Assert
 osclbase, 34
oscl_assert.h, 636
OSCL_ASSERT_ALWAYS
 osclconfig, 20
oscl_atan
 osclutil, 70
OSCL_AUDIT_ARRAY_NEW
 osclmemory, 50
OSCL_AUDIT_CALLOC
 osclmemory, 51
OSCL_AUDIT_MALLOC
 osclmemory, 51
OSCL_AUDIT_NEW
 osclmemory, 51
OSCL_AUDIT_REALLOC
 osclmemory, 52
OSCL_BAD_ALLOC_EXCEPTION_CODE
 osclerror, 84
oscl_base.h, 637
oscl_base_alloc.h, 638
oscl_base_macros.h, 639
OSCL_BEGIN_PACKED
 osclbase, 30
 osclconfig.h, 777
oscl_bin_stream.h, 640
OSCL_BYPASS_MEMMGT
 osclconfig_memory.h, 797
oscl_byte_order.h, 641
OSCL_BYTE_ORDER_BIG_ENDIAN
 osclconfig, 20
OSCL_BYTE_ORDER_LITTLE_ENDIAN
 osclconfig, 20
OSCL_CALLOC
 osclmemory, 52

oscl_calloc
 osclmemory, 52
OSCL_CATCH
 osclerror, 84
OSCL_CATCH_ANY
 osclerror, 85
OSCL_CHAR_IS_SIGNED
 osclconfig_limits_typedefs.h, 796
OSCL_CHAR_IS_UNSIGNED
 osclconfig_limits_typedefs.h, 796
oscl_chdir
 osclio, 95
oscl_Clstrcmp
 osclbase, 34, 35
oscl_Clstrcmp
 osclbase, 35
OSCL_CLEANUP_BASE_CLASS
 osclmemory, 52
OSCL_CLOCK_HAS_DRIFT_CORRECTION
 osclconfig_util.h, 817
OSCL_COND_EXPORT_REF
 osclbase, 30
OSCL_COND_IMPORT_REF
 osclbase, 30
OSCL_CONST_CAST
 osclbase, 30
oscl_cos
 osclutil, 70
Oscl_Dealloc, 166
 deallocate, 166
Oscl_DefAlloc, 167
Oscl_DefAlloc
 allocate, 167
 allocate_fl, 167
 deallocate, 167
oscl_defalloc.h, 642
Oscl_DefAllocWithRefCounter, 168
Oscl_DefAllocWithRefCounter
 addRef, 168
 Delete, 168
 getCount, 168
 New, 169
 removeRef, 169
OSCL_DEFAULT_FREE
 osclmemory, 53
OSCL_DEFAULT_MALLOC
 osclmemory, 53
OSCL_DELETE
 osclmemory, 53
Oscl_DeleteFile
 Oscl_FileServer, 185, 186
OSCL_DISABLE_INLINES
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816

**OSCL_DISABLE_WARNING_RETURN_-
TYPE_NOT_UDT**
 osclbase, 30
 osclmemory, 53

**OSCL_DISABLE_WARNING_TRUNCATE_-
DEBUG_MESSAGE**
 oscl_map.h, 683
 oscl_mem.h, 689
 oscl_mem_audit.h, 692
 oscl_mem_audit_internals.h, 693
 oscl_mem_auto_ptr.h, 694
 oscl_tagtree.h, 761
 oscl_tree.h, 770
 osclbase, 30
 osclmemory, 53
 oscldll.h, 643

OSCL_DLL_ENTRY_POINT
 osclbase, 30

OSCL_DLL_ENTRY_POINT_DEFAULT
 osclbase, 31

oscl_dns.h, 644
 oscl_dns_gethostbyname.h, 645
 oscl_dns_imp.h, 646
 oscl_dns_imp_base.h, 647
 oscl_dns_imp_pv.h, 648
 oscl_dns_method.h, 649
 oscl_dns_param.h, 650
 TDNSRequestParamAllocator, 650
 oscl_dns_request.h, 651
 oscl_dns_tuneables.h, 652
 PV_DNS_IS_THREAD, 652
 PV_DNS_SERVER, 652

oscl_double_list.h, 653

OSCL_DYNAMIC_CAST
 osclbase, 31

OSCL_END_PACKED
 osclbase, 31
 oscldconfig.h, 777

OSCL_ERR_NONE
 osclerror, 85

oscl_errno.h, 654

oscl_error.h, 655

oscl_error_allocator.h, 656

oscl_error_codes.h, 657

oscl_error_imp.h, 658

oscl_error_imp_cppexceptions.h, 659

oscl_error_imp_fatalerror.h, 660
 _PV_TRAP, 660
 _PV_TRAP_NO_TLS, 660
 PSError_DoLeave, 660

oscl_error_imp_jumps.h, 661
 _PV_TRAP, 661
 _PV_TRAP_NO_TLS, 661
 PSError_DoLeave, 662

oscl_error_trapcleanup.h, 663

oscl_exception.h, 664

OSCL_EXCEPTSET_FLAG
 oscl_socket_serv_imp_pv.h, 743

oscl_exclusive_ptr.h, 665

oscl_exp
 osclutil, 71

OSCL_FastString, 170
 OSCL_FastString, 171

OSCL_FastString
 ~OSCL_FastString, 171
 chartype, 171
 get_cstr, 171
 get_maxsize, 171
 get_size, 172
 get_str, 172
 operator=, 172
 OSCL_FastString, 171
 OSCL_String, 172
 set, 172
 set_length, 172

Oscl_File
 ESymbianAccessMode_Rfile, 175
 ESymbianAccessMode_RfileBuf, 175
 MODE_APPEND, 175
 MODE_BINARY, 175
 MODE_READ, 175
 MODE_READ_PLUS, 175
 MODE_READWRITE, 175
 MODE_TEXT, 175
 SEEKCUR, 175
 SEEKEND, 175
 SEEKSET, 175

Oscl_File, 174
 ~Oscl_File, 176
 asyncfilereadcancel_test, 180
 asyncfilereadwrite_test, 180
 Close, 176
 EndOfFile, 176
 Flush, 176
 GetError, 176
 Handle, 177
 mode_type, 175
 Open, 177
 Oscl_File, 175, 176
 Oscl_FileServer, 186
 OsclFileCache, 180
 OsclFileHandle, 391
 Read, 177
 Seek, 178
 seek_type, 175
 SetAsyncReadBufferSize, 178
 SetFileHandle, 178
 SetLoggingEnable, 179

SetNativeAccessMode, 179
 SetNativeBufferSize, 179
 SetPVCacheSize, 179
 SetSummaryStatsLoggingEnable, 179
 Size, 179
 Tell, 180
 TSymbianAccessMode, 175
 Write, 180
 oscl_file_async_read.h, 666
OSCL_FILE_BUFFER_MAX_SIZE
 osclconfig_io.h, 787
 oscl_file_cache.h, 667
OSCL_FILE_CHAR_PATH_DELIMITER
 osclio, 93
 oscl_file_dir_utils.h, 668
 oscl_file_find.h, 670
 oscl_file_handle.h, 671
 oscl_file_io.h, 672
 oscl_file_native.h, 673
 oscl_file_server.h, 674
 oscl_file_stats.h, 675
OSCL_FILE_STATS_LOGGER_NODE
 osclio, 93
 oscl_file_types.h, 676
OSCL_FILE_WCHAR_PATH_DELIMITER
 osclio, 93
Oscl_FileFind
 DIR_TYPE, 181
 E_BUFFER_TOO_SMALL, 182
 E_INVALID_ARG, 181
 E_INVALID_STATE, 181
 E_NO_MATCH, 182
 E_NOT_IMPLEMENTED, 182
 E_OK, 181
 E_OTHER, 182
 E_PATH_NOT_FOUND, 181
 E_PATH_TOO_LONG, 181
 FILE_TYPE, 181
 INVALID_TYPE, 181
Oscl_FileFind, 181
 Oscl_FileFind, 182
Oscl_FileFind
 ~Oscl_FileFind, 182
 Close, 182
 element_type, 181
 error_type, 181
 FindFirst, 182
 FindNext, 183
 GetElementType, 183
 GetLastError, 183
 Oscl_FileFind, 182
OSCL_FILEMGMT_E_ALREADY_EXISTS
 osclio, 93
OSCL_FILEMGMT_E_NO_MATCH
 osclio, 94
OSCL_FILEMGMT_E_NOT_EMPTY
 osclio, 94
OSCL_FILEMGMT_E_NOT_IMPLEMENTED
 osclio, 94
OSCL_FILEMGMT_E_OK
 osclio, 93
OSCL_FILEMGMT_E_PATH_NOT_FOUND
 osclio, 93
OSCL_FILEMGMT_E_PATH_TOO_LONG
 osclio, 93
OSCL_FILEMGMT_E_PERMISSION_DENIED
 osclio, 94
OSCL_FILEMGMT_E_SYS_SPECIFIC
 osclio, 94
OSCL_FILEMGMT_E_UNKNOWN
 osclio, 94
OSCL_FILEMGMT_ERR_TYPE
 osclio, 93
OSCL_FILEMGMT_MODE_DIR
 osclio, 94
OSCL_FILEMGMT_MODES
 osclio, 94
OSCL_FILEMGMT_PERMS
 osclio, 94
OSCL_FILEMGMT_PERMS_EXECUTE
 osclio, 94
OSCL_FILEMGMT_PERMS_READ
 osclio, 94
OSCL_FILEMGMT_PERMS_WRITE
 osclio, 94
Oscl_FileServer, 185
 Oscl_FileServer, 185
Oscl_FileServer
 ~Oscl_FileServer, 185
 Close, 185
 Connect, 185
 Oscl_DeleteFile, 185, 186
 Oscl_File, 186
 Oscl_FileServer, 185
 OsclNativeFile, 186
OSCL_FIRST_CATCH
 osclerror, 85
OSCL_FIRST_CATCH_ANY
 osclerror, 85
oscl_floor
 osclutil, 71
OSCL_FREE
 osclmemory, 53
oscl_free
 osclmemory, 53
OSCL_FSSTAT

osclio, 93
oscl_fsstat, 187
 freebytes, 187
 totalbytes, 187
OSCL_FUNCTION_PTR
 osclconfig_compiler_warnings.h, 780
oscl_getcwd
 osclio, 95
OSCL_GetLastError
 osclerror, 89
OSCL_HAS_ANSI_FILE_IO_SUPPORT
 osclconfig_io.h, 787
OSCL_HAS_ANSI_MATH_SUPPORT
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
OSCL_HAS_ANSI_MEMORY_FUNCS
 osclconfig_ansi_memory.h, 778
OSCL_HAS_ANSI_STDIO_SUPPORT
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
OSCL_HAS_ANSI_STDLIB_SUPPORT
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
OSCL_HAS_ANSI_STRING_SUPPORT
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
**OSCL_HAS_ANSI_WIDE_STRING_-
 SUPPORT**
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
OSCL_HAS_BASIC_LOCK
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
OSCL_HAS_BERKELEY_SOCKETS
 osclconfig, 20
 osclconfig_io.h, 787
OSCL_HAS_ERRNO_H
 osclconfig_error.h, 781
OSCL_HAS_EXCEPTIONS
 osclconfig_error.h, 781
OSCL_HAS_GLOB
 osclconfig_io.h, 787
OSCL_HAS_GLOBAL_NEW_DELETE
 osclconfig_memory.h, 797
 osclmemory, 53
**OSCL_HAS_GLOBAL_VARIABLE_-
 SUPPORT**
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
OSCL_HAS_HEAP_BASE_SUPPORT
 osclconfig_memory.h, 797
OSCL_HAS_MSWIN_PARTIAL_SUPPORT
 osclconfig, 21
OSCL_HAS_MSWIN_SUPPORT

 osclconfig, 21
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
**OSCL_HAS_NATIVE_FILE_CACHE_-
 ENABLE**
 osclconfig_io.h, 787
OSCL_HAS_NATIVE_INT64_TYPE
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
OSCL_HAS_NATIVE_UINT64_TYPE
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
OSCL_HAS_NJ_FILE_IO_SUPPORT
 osclconfig.h, 777
OSCL_HAS_NJ_SUPPORT
 osclconfig.h, 777
**OSCL_HAS_NON_PREEMPTIVE_-
 THREAD_SUPPORT**
 osclconfig_proc_unix_common.h, 804
 osclconfig_proc_unix_nj.h, 806
OSCL_HAS_PTHREAD_SUPPORT
 osclconfig, 21
 osclconfig_proc_unix_common.h, 804
 osclconfig_proc_unix_nj.h, 806
**OSCL_HAS_PV_C_OS_API_MEMORY_-
 FUNCS**
 osclconfig, 21
OSCL_HAS_PV_C_OS_SUPPORT
 osclconfig, 21
OSCL_HAS_PV_C_OS_TIME_FUNCS
 osclconfig, 21
OSCL_HAS_PV_FILE_CACHE
 osclconfig_io.h, 787
**OSCL_HAS_RUNTIME_LIB_LOADING_-
 SUPPORT**
 osclconfig_lib.h, 794
OSCL_HAS_SAVAJE_IO_SUPPORT
 osclconfig, 21
OSCL_HAS_SAVAJE_SUPPORT
 osclconfig, 21
OSCL_HAS_SEM_TIMEDWAIT_SUPPORT
 osclconfig, 21
 osclconfig_proc_unix_common.h, 804
 osclconfig_proc_unix_nj.h, 806
OSCL_HAS_SETJMP_H
 osclconfig_error.h, 781
OSCL_HAS_SINGLETON_SUPPORT
 osclbase, 31
OSCL_HAS_SOCKET_SUPPORT
 osclconfig_io.h, 787
**OSCL_HAS_SYMBIAN_COMPATIBLE_IO_-
 FUNCTION**
 osclconfig, 21
 osclconfig_io.h, 787

OSCL_HAS_SYMBIAN_DNS_SERVER
 osclconfig, 21
 osclconfig_io.h, 787

OSCL_HAS_SYMBIAN_ERRORTRAP
 osclconfig, 21
 osclconfig_error.h, 781

OSCL_HAS_SYMBIAN_MATH
 osclconfig, 21
 osclconfig_util.h, 817

OSCL_HAS_SYMBIAN_MEMORY_FUNCS
 osclconfig, 21
 osclconfig_memory.h, 797

OSCL_HAS_SYMBIAN_SCHEDULER
 osclconfig, 21
 osclconfig_proc_unix_common.h, 804
 osclconfig_proc_unix_nj.h, 806

OSCL_HAS_SYMBIAN_SOCKET_SERVER
 osclconfig, 21
 osclconfig_io.h, 787

OSCL_HAS_SYMBIAN_SUPPORT
 osclconfig, 21
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816

OSCL_HAS_SYMBIAN_TIMERS
 osclconfig, 21
 osclconfig_util.h, 817

OSCL_HAS_THREAD_SUPPORT
 osclconfig_proc_unix_common.h, 804
 osclconfig_proc_unix_nj.h, 806

OSCL_HAS_TLS_SUPPORT
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816

OSCL_HAS_UNICODE_SUPPORT
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816

OSCL_HAS_UNIX_SUPPORT
 osclconfig, 21
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816

OSCL_HAS_UNIX_TIME_FUNCS
 osclconfig, 21
 osclconfig_time.h, 807

oscl_heapbase.h, 677

OSCL_HeapString, 188
 osclutil, 71

OSCL_HeapString
 chartype, 189
 OSCL_String, 189

OSCL_HeapStringA, 190
 OSCL_HeapStringA, 191, 192

OSCL_HeapStringA
 ~OSCL_HeapStringA, 192
 chartype, 191
 get_cstr, 192

get_maxsize, 192
 get_size, 192
 get_str, 193
 operator=, 193
 OSCL_HeapStringA, 191, 192
 OSCL_String, 193
 set, 193

oscl_init.h, 678

OSCL_INLINE
 osclbase, 31

Oscl_Int64_Utils, 194
 get_int64_lower32, 195
 get_int64_middle32, 195
 get_int64_upper32, 195
 get_uint64_lower32, 195
 get_uint64_middle32, 195
 get_uint64_upper32, 195
 set_int64, 195
 set_uint64, 195

oscl_int64_utils.h, 679
 _OsclInteger64Transport, 679

OSCL_INTEGERS_WORD_ALIGNED
 osclconfig, 21

OSCL_IO_EXTENSION_MAXLEN
 osclio, 93

OSCL_IO_FILENAME_MAXLEN
 osclio, 93

oscl_ip_socket.h, 680

OSCL_IPPROTO_TCP
 osclconfig_io.h, 787

OSCL_IPPROTO_UDP
 osclconfig_io.h, 787

oscl_isdigit
 osclutil, 66

OSCL_IsErrnoSupported
 osclerror, 89

oscl_isLetter
 osclbase, 35

OSCL_JUMP_MAX_JUMP_MARKS
 osclerror, 85

OSCL_LAST_CATCH
 osclerror, 85

OSCL_LEAVE
 osclerror, 86

Oscl_Less, 196
 operator(), 196

OSCL_LIB_READ_DEBUG_LIBS
 osclconfig_lib.h, 794

Oscl_Linked_List, 197
 ~Oscl_Linked_List, 197
 add_element, 198
 add_to_front, 198
 check_list, 198
 dequeue_element, 198

get_element, 198
 get_first, 198
 get_index, 199
 get_next, 199
 get_num_elements, 199
 move_to_end, 199
 move_to_front, 199
 Oscl_Linked_List, 197
 remove_element, 200
 oscl_linked_list.h, 681
 Oscl_Linked_List_Base, 201
 ~Oscl_Linked_List_Base, 202
 add_element, 202
 add_to_front, 202
 check_list, 202
 construct, 202
 destroy, 202
 get_element, 202
 get_first, 203
 get_index, 203
 get_next, 203
 head, 204
 iterator, 204
 move_to_end, 203
 move_to_front, 203
 num_elements, 204
 remove_element, 204
 sizeof_T, 204
 tail, 204
 oscl_lock_base.h, 682
 oscl_log
 osclutil, 72
 oscl_log10
 osclutil, 72
 OSCL_MALLOC
 osclmemory, 54
 oscl_malloc
 osclmemory, 54
 Oscl_Map, 205
 begin, 208
 clear, 208
 const_iterator, 207
 const_reference, 207
 count, 208
 empty, 208
 end, 208
 equal_range, 208
 erase, 209
 find, 209
 insert, 209
 iterator, 207
 key_comp, 210
 key_compare, 207
 key_type, 207
 lower_bound, 210
 max_size, 210
 operator=, 210
 operator[], 210
 Oscl_Map, 207
 pair_citerator_citerator, 207
 pair_iterator_bool, 207
 pair_iterator_iterator, 207
 pointer, 207
 reference, 207
 self, 207
 size, 210
 size_type, 207
 upper_bound, 210, 211
 value_comp, 211
 value_type, 207
 oscl_map.h, 683
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 683
 Oscl_Map::value_compare, 212
 comp, 212
 operator(), 212
 Oscl_Map< Key, T, Alloc, Compare >, 212
 value_compare, 212
 Oscl_Map< Key, T, Alloc, Compare >
 Oscl_Map::value_compare, 212
 oscl_math.h, 684
 OSCL_MAX
 osclbase, 31
 OSCL_MAX_TRAP_LEVELS
 osclerror, 86
 oscl_media_data.h, 685
 oscl_media_status.h, 686
 oscl_mem.h, 687
 operator delete, 689
 operator new, 689
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 689
 oscl_mem_align.h, 690
 oscl_mem_aligned_size
 osclmemory, 57
 OsclMemPoolAllocator, 423
 oscl_mem_audit.h, 691
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 692
 oscl_mem_audit_internals.h, 693
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 693
 oscl_mem_auto_ptr.h, 694

OSCL_DISABLE_WARNING_-_TRUNCATE_DEBUG_MESSAGE,
694
oscl_mem_basic_functions.h, *695*
oscl_mem_inst.h, *696*
oscl_mem_mempool.h, *697*
oscl_memcmp
 osclmemory, *58*
oscl_memcpy
 osclmemory, *58*
OSCL_MEMFRAG_PTR_BEFORE_LEN
 osclconfig_unix_common.h, *812*
 osclconfig_unix_nj.h, *816*
oscl_memmove
 osclmemory, *58*
oscl_memmove32
 osclmemory, *58*
oscl_mempool_allocator.h, *698*
oscl_memset
 osclmemory, *59*
oscl_memsize_t
 osclconfig_ansi_memory.h, *778*
OSCL_MIN
 osclbase, *31*
oscl_mkdir
 osclio, *96*
Oscl_MTLinked_List, *214*
 ~Oscl_MTLinked_List, *214*
 add_element, *215*
 add_to_front, *215*
 dequeue_element, *215*
 get_element, *215*
 get_index, *215*
 move_to_end, *215*
 move_to_front, *216*
 Oscl_MTLinked_List, *214*
 remove_element, *216*
 the_list, *216*
oscl_mutex.h, *699*
 OsclNoYieldMutex, *699*
oscl_namestring.h, *700*
OSCL_NATIVE_INT64_TYPE
 osclconfig_unix_common.h, *812*
 osclconfig_unix_nj.h, *816*
OSCL_NATIVE_UINT64_TYPE
 osclconfig.h, *777*
 osclconfig_unix_common.h, *812*
 osclconfig_unix_nj.h, *816*
OSCL_NATIVE_WCHAR_TYPE
 osclconfig_unix_common.h, *812*
 osclconfig_unix_nj.h, *816*
OSCL_NEW
 osclmemory, *54*
oscl_opaque_type.h, *701*
Oscl_Opaque_Type_Alloc, *218*
 allocate, *218*
 construct, *218*
 deallocate, *218*
 destroy, *218*
Oscl_Opaque_Type_Alloc_LL, *219*
 allocate, *219*
 compare_data, *219*
 construct, *219*
 deallocate, *219*
 destroy, *219*
 get_data, *220*
 get_next, *220*
 set_next, *220*
Oscl_Opaque_Type_Compare, *221*
 compare_EQ, *221*
 compare_LT, *221*
 swap, *221*
OSCL_PACKED_STRUCT_BEGIN
 osclconfig.h, *777*
OSCL_PACKED_STRUCT_END
 osclconfig.h, *777*
OSCL_PACKED_VAR
 osclbase, *31*
 osclconfig.h, *777*
Oscl_Pair, *223*
 first, *223*
 Oscl_Pair, *223*
 second, *223*
OSCL_PERF_SUMMARY_LOGGING
 osclproc, *101*
OSCL_PLACEMENT_NEW
 osclmemory, *54*
oscl_pow
 osclutil, *72*
oscl_priqueue.h, *702*
oscl_priqueue_test
 OsclPriorityQueue, *454*
oscl_procstatus.h, *703*
Oscl_Queue, *224*
 ~Oscl_Queue, *225*
 back, *225*
 clear, *225*
 const_reference, *225*
 front, *226*
 Oscl_Queue, *225*
 pointer, *225*
 pop, *226*
 push, *226*
 reference, *225*
 size_type, *225*
 value_type, *225*
oscl_queue.h, *704*
Oscl_Queue_Base, *227*

~Oscl_Queue_Base, 227
 bufsize, 229
 capacity, 228
 clear, 228
 construct, 228
 destroy, 228
 elems, 229
 empty, 228
 ifront, 229
 irear, 229
 numelems, 229
 pop, 228
 push, 228
 reserve, 228
 size, 228
 sizeof_T, 229
 oscl_rand.h, 705
 OSCL_RAND_MAX
 osclconfig_util.h, 817
 Oscl_Rb_Tree, 230
 ~Oscl_Rb_Tree, 232
 begin, 232
 clear, 232
 const_iterator, 232
 const_pointer, 232
 const_reference, 232
 count, 232
 difference_type, 232
 empty, 232
 end, 232
 equal_range, 232
 erase, 232
 find, 232
 insert_unique, 232
 iterator, 232
 key_type, 232
 link_type, 232
 lower_bound, 232
 max_size, 232
 operator=, 232
 Oscl_Rb_Tree, 232
 pointer, 232
 reference, 232
 size, 232
 size_type, 232
 upper_bound, 232
 value_type, 232
 Oscl_Rb_Tree_Base, 234
 base_link_type, 234
 rebalance, 234
 rebalance_for_erase, 234
 rotate_left, 234
 rotate_right, 234
 Oscl_Rb_Tree_Const_Iterator, 235
 base_link_type, 236
 const_iterator, 236
 link_type, 236
 node, 236
 operator *, 236
 operator!=, 236
 operator++, 236
 operator-, 236
 operator->, 236
 operator==, 236
 Oscl_Rb_Tree_Const_Iterator, 236
 pointer, 236
 reference, 236
 self, 236
 value_type, 236
 Oscl_Rb_Tree_Iterator, 238
 base_link_type, 239
 iterator, 239
 link_type, 239
 node, 239
 operator *, 239
 operator!=, 239
 operator++, 239
 operator-, 239
 operator->, 239
 operator==, 239
 Oscl_Rb_Tree_Iterator, 239
 pointer, 239
 reference, 239
 self, 239
 value_type, 239
 Oscl_Rb_Tree_Node, 241
 link_type, 241
 value, 241
 value_type, 241
 Oscl_Rb_Tree_Node_Base
 black, 242
 red, 242
 Oscl_Rb_Tree_Node_Base, 242
 base_link_type, 242
 color, 243
 color_type, 242
 left, 243
 maximum, 243
 minimum, 243
 parent, 243
 RedBl, 242
 right, 243
 OSCL_READSET_FLAG
 oscl_socket_serv_imp_pv.h, 743
 OSCL_REALLOC
 osclmemory, 54
 oscl_realloc
 osclmemory, 54

oscl_refcounter.h, 706
 oscl_refcounter_memfrag.h, 707
 oscl_registry_access_client.h, 708
 oscl_registry_client.h, 709
 oscl_registry_client_impl.h, 710
 oscl_registry_serv_impl.h, 711
 oscl_registry_serv_impl_global.h, 712
 oscl_registry_serv_impl_tls.h, 713
 oscl_registry_types.h, 714
OSCL_REINTERPRET_CAST
 osclbase, 31
OSCL_RELEASE_BUILD
 osclconfig.h, 777
 oscl_rename
 osclio, 96
OSCL_REQUEST_ERR_CANCEL
 osclproc, 102
OSCL_REQUEST_ERR_GENERAL
 osclproc, 102
OSCL_REQUEST_ERR_NONE
 osclproc, 102
OSCL_REQUEST_PENDING
 osclproc, 102
 oscl_rmdir
 osclio, 97
 oscl_scheduler.h, 715
 oscl_scheduler_ao.h, 716
 oscl_scheduler_aobase.h, 717
 oscl_scheduler_readyq.h, 718
 oscl_scheduler_threadcontext.h, 719
 oscl_scheduler_tuneables.h, 720
 oscl_scheduler_types.h, 721
OSCL_SD_BOTH
 osclconfig_io.h, 787
OSCL_SD_RECEIVE
 osclconfig_io.h, 787
OSCL_SD_SEND
 osclconfig_io.h, 787
 Oscl_Select1st, 244
 operator(), 244
 oscl_semaphore.h, 722
 OSCL_SetLastError
 osclerror, 89
 oscl_shared_ptr.h, 723
 oscl_sin
 osclutil, 72
 oscl_singleton.h, 724
 OSCL_SINGLETON_ID_CPM_PLUGIN,
 725
 OSCL_SINGLETON_ID_LAST, 725
 OSCL_SINGLETON_ID_OMX, 725
 OSCL_SINGLETON_ID_-
 OMXMASTERCORE, 725
OSCL_SINGLETON_ID_OSCLMEM,
 725
OSCL_SINGLETON_ID_-
 OSCLREGISTRY, 725
OSCL_SINGLETON_ID_-
 PAYLOADPARSER, 725
OSCL_SINGLETON_ID_-
 PVERRORTRAP, 725
OSCL_SINGLETON_ID_PVLOGGER,
 725
OSCL_SINGLETON_ID_-
 PVMFRECOGNIZER, 725
OSCL_SINGLETON_ID_-
 PVSCHEDULER, 725
OSCL_SINGLETON_ID_-
 SDPMEDIAPARSER, 725
OSCL_SINGLETON_ID_TEST, 725
OSCL_SINGLETON_ID_TICKCOUNT,
 725
OSCL_SINGLETON_ID_CPM_PLUGIN
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_LAST
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_OMX
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_-
 OMXMASTERCORE
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_OSCLMEM
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_OSCLREGISTRY
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_PAYLOADPARSER
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_PVERRORTRAP
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_PVLOGGER
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_-
 PVMFRECOGNIZER
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_PVSCHEDULER
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_-
 SDPMEDIAPARSER
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_TEST
 oscl_singleton.h, 725
OSCL_SINGLETON_ID_TICKCOUNT
 oscl_singleton.h, 725
 oscl_snprintf
 osclutil, 72
 oscl_snprintf.h, 726
OSCL_SOCKETDATAGRAM

osclconfig_io.h, [787](#)
OSCL_SOCKET_STREAM
 osclconfig_io.h, [787](#)
 oscl_socket.h, [727](#)
 oscl_socket_accept.h, [728](#)
 oscl_socket_bind.h, [729](#)
 oscl_socket_connect.h, [730](#)
 oscl_socket_imp.h, [731](#)
 oscl_socket_imp_base.h, [732](#)
 oscl_socket_imp_pv.h, [733](#)
 PVSOCK_ERR_BAD_PARAM, [733](#)
 PVSOCK_ERR_NOT_IMPLEMENTED, [733](#)
 PVSOCK_ERR_SERV_NOT_CONNECTED, [733](#)
 PVSOCK_ERR SOCK_NO_SERV, [733](#)
 PVSOCK_ERR SOCK_NOT_CONNECTED, [733](#)
 PVSOCK_ERR SOCK_NOT_OPEN, [733](#)
 oscl_socket_listen.h, [734](#)
 OSCL_SOCKET_LISTEN_H_INCLUDEDd, [734](#)
 OSCL_SOCKET_LISTEN_H_INCLUDEDd
 oscl_socket_listen.h, [734](#)
 oscl_socket_method.h, [735](#)
 MSEC_TO_MICROSEC, [735](#)
 oscl_socket_recv.h, [736](#)
 oscl_socket_recv_from.h, [737](#)
 oscl_socket_request.h, [738](#)
 oscl_socket_send.h, [739](#)
 oscl_socket_send_to.h, [740](#)
 oscl_socket_serv_imp.h, [741](#)
 oscl_socket_serv_imp_base.h, [742](#)
 oscl_socket_serv_imp_pv.h, [743](#)
 OSCL_EXCEPTSET_FLAG, [743](#)
 OSCL_READSET_FLAG, [743](#)
 OSCL_WRITESET_FLAG, [743](#)
 oscl_socket_serv_imp_reqlist.h, [744](#)
 oscl_socket_shutdown.h, [745](#)
 oscl_socket_stats.h
 EOsclSocket_DataRecv, [747](#)
 EOsclSocket_DataSent, [747](#)
 EOsclSocket_Except, [746](#)
 EOsclSocket_OS, [746](#)
 EOsclSocket_Readable, [746](#)
 EOsclSocket_RequestAO_Canceled, [746](#)
 EOsclSocket_RequestAO_Error, [746](#)
 EOsclSocket_RequestAO_Success, [746](#)
 EOsclSocket_RequestAO_Timeout, [746](#)
 EOsclSocket_ServPoll, [746](#)
 EOsclSocket_ServRequestCancelIssued, [747](#)
 EOsclSocket_ServRequestComplete, [747](#)
 EOsclSocket_ServRequestIssued, [746](#)
 EOsclSocket_Writable, [746](#)
 EOsclSocketServ_LastEvent, [746](#)
 EOsclSocketServ_LoopsckError, [747](#)
 EOsclSocketServ_LoopsckOk, [747](#)
 EOsclSocketServ_SelectActivity, [746](#)
 EOsclSocketServ_SelectNoActivity, [746](#)
 EOsclSocketServ_SelectRescheduleAsap, [746](#)
 EOsclSocketServ_SelectReschedulePoll, [746](#)
 oscl_socket_stats.h, [746](#)
 TOsclSocketStatEvent, [746](#)
 TOsclSocketStatEvent, [746](#)
 oscl_socket_tuneables.h, [748](#)
 PV_OSCL_SOCKET_1MB_RECV_BUF, [748](#)
 PV_OSCL_SOCKET_SERVER_LOGGER_OUTPUT, [748](#)
 PV_OSCL_SOCKET_STATS_LOGGING, [748](#)
 PV_SOCKET_REQUEST_AO_PRIORITY, [748](#)
 PV_SOCKET_SERVER, [748](#)
 PV_SOCKET_SERVER_AO_INTERVAL_MSEC, [749](#)
 PV_SOCKET_SERVER_AO_PRIORITY, [749](#)
 PV_SOCKET_SERVER_IS_THREAD, [749](#)
 PV_SOCKET_SERVER_SELECT, [749](#)
 PV_SOCKET_SERVER_SELECT_LOOPBACK_SOCKET, [749](#)
 PV_SOCKET_SERVER_SELECT_TIMEOUT_MSEC, [749](#)
 PV_SOCKET_SERVER_THREAD_PRIORITY, [749](#)
 PV_SOCKET_SERVI_STATS, [749](#)
 oscl_socket_types.h
 EPVSocket_Last, [751](#)
 EPVSocketAccept, [751](#)
 EPVSocketBind, [751](#)
 EPVSocketBothShutdown, [751](#)
 EPVSocketCancel, [750](#)
 EPVSocketConnect, [751](#)
 EPVSocketFailure, [750](#)
 EPVSocketListen, [751](#)
 EPVSocketPending, [750](#)
 EPVSocketRecv, [751](#)
 EPVSocketRecvFrom, [751](#)
 EPVSocketRecvShutdown, [751](#)
 EPVSocketSend, [751](#)
 EPVSocketSendShutdown, [751](#)
 EPVSocketSendTo, [751](#)
 EPVSocketShutdown, [751](#)

EPVSocketSuccess, 750
 EPVSocketTimeout, 750
oscl_socket_types.h, 750
 PVNETWORKADDRESS_LEN, 750
 TPVSocketEvent, 750
 TPVSocketFxn, 750
 TPVSocketShutdown, 751
oscl_sqrt
 osclutil, 72
OSCL_StackString, 245
 osclutil, 72, 73
OSCL_StackString
 chartype, 246
 OSCL_String, 246
oscl_stat
 oscilio, 97
OSCL_STAT_BUF
 oscilio, 93
oscl_stat_buf, 247
 mode, 247
 perms, 247
oscl_stats
 oscilio, 98
OSCL_STATIC_CAST
 oscibase, 31
oscl_stdstring.h, 752
oscl_str_escape_xml
 osclutil, 73
oscl_str_is_valid_utf8
 osclutil, 74
oscl_str_need_escape_xml
 osclutil, 74
oscl_str_ptr_len.h, 753
oscl_str_truncate_utf8
 osclutil, 74
oscl_str_unescape_uri
 osclutil, 75
oscl_strcat
 oscibase, 36
oscl_strchr
 oscibase, 36, 37
oscl_strcmp
 oscibase, 37
OSCL_StrError
 oscilerror, 89
OSCL_String, 248
 ~OSCL_String, 249
 append_rep, 249
 chartype, 249
 get_cstr, 249
 get_maxsize, 249
 get_size, 249
 get_str, 250
 hash, 250
 is_writable, 250
 operator!=, 250
 operator+=, 250
 operator<, 250
 operator<=, 250
 operator=, 250, 251
 operator==, 251
 operator>, 251
 operator>=, 251
 operator[], 251
 OSCL_FastString, 172
 OSCL_HeapString, 189
 OSCL_HeapStringA, 193
 OSCL_StackString, 246
 OSCL_String, 249
 read, 251
 set_len, 251
 set_rep, 251, 252
 write, 252
oscl_string.h, 754
oscl_string_containers.h, 755
oscl_string_rep.h, 756
oscl_string_uri.h, 757
oscl_string_utf8.h, 758
oscl_string_utils.h, 759
oscl_string_xml.h, 760
oscl_strlen
 oscibase, 37
oscl_strncat
 oscibase, 38
oscl_strncmp
 oscibase, 38, 39
oscl_strncpy
 oscibase, 39
oscl strrchr
 oscibase, 40
oscl_strset
 oscibase, 40
oscl_strstr
 oscibase, 40, 41
Oscl_Tag, 253
 ~Oscl_Tag, 253
 operator<, 253
Oscl_Tag, 253
 tag, 253
 tagAllocator, 253
Oscl_Tag_Base, 255
 operator(), 256
 size_type, 256
 tag_ancestor, 256
 tag_base_type, 256
 tag_base_unit, 256
 tag_cmp, 256
 tag_copy, 256

tag_depth, 256
 tag_len, 256
Oscl_TagTree, 257
 Oscl_TagTree, 258
Oscl_TagTree
 ~Oscl_TagTree, 258
 begin, 258
 children_type, 258
 clear, 259
 count, 259
 empty, 259
 end, 259
 erase, 259
 find, 259
 insert, 260
 map_type, 258
 node_ptr, 258
 node_type, 258
 operator=, 260
 operator[], 260
 Oscl_TagTree, 258
 pair_iterator_bool, 258
 size, 260
 size_type, 258
 tag_base_type, 258
 tag_type, 258
 value_type, 258
oscl_tagtree.h, 761
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 761
Oscl_TagTree::const_iterator, 261
Oscl_TagTree::const_iterator
 const_iterator, 262
 mapit, 262
 mapiter, 262
 operator *, 262
 operator!=, 262
 operator++, 262
 operator-, 262
 operator->, 262
 operator==, 262
 pointer, 262
 reference, 262
 self, 262
Oscl_TagTree::iterator, 264
Oscl_TagTree::iterator
 iterator, 265
 mapit, 265
 mapiter, 265
 operator *, 265
 operator!=, 265
 operator++, 265
 operator-, 265
 operator->, 265
 operator==, 265
 pointer, 265
 reference, 265
 self, 265
 operator->, 265
 operator==, 265
 pointer, 265
 reference, 265
 self, 265
Oscl_TagTree::Node, 267
Oscl_TagTree::Node
 children, 268
 children_type, 268
 depth, 268
 Node, 268
 parent, 268
 sort_children, 268
 tag, 268
 value, 268
Oscl_TAlloc, 269
 ~Oscl_TAlloc, 270
 address, 270
 alloc_and_construct, 270
 alloc_and_construct_fl, 270
 allocate, 270
 allocate_fl, 270
 const_pointer, 270
 const_reference, 270
 construct, 270
 deallocate, 270
 destroy, 270
 destruct_and_dealloc, 270
 pointer, 270
 reference, 270
 size_type, 270
 value_type, 270
Oscl_TAlloc::rebind, 272
 other, 272
oscl_tan
 osclutil, 76
OSCL_TCHAR
 osclbase, 32
oscl_tcp_socket.h, 762
OSCL_TEMPLATED_DESTRUCTOR_CALL
 osclbase, 31
 osclconfig.h, 777
oscl_thread.h
 Start_on_creation, 763
 Suspend_on_creation, 763
 ThreadPriorityAboveNormal, 764
 ThreadPriorityBelowNormal, 763
 ThreadPriorityHighest, 764
 ThreadPriorityLow, 763
 ThreadPriorityLowest, 763
 ThreadPriorityNormal, 763
 ThreadPriorityTimeCritical, 764
oscl_thread.h, 763
OsclThread_State, 763

OsclThreadPriority, [763](#)
 TOsclThreadFuncPtr, [763](#)
OSCL_THREAD_DECL
 osclconfig_proc_unix_common.h, [804](#)
 osclconfig_proc_unix_nj.h, [806](#)
 oscl_tickcount.h, [765](#)
 oscl_time.h, [766](#)
 oscl_timer.h, [768](#)
 oscl_tls.h, [769](#)
OSCL_TLS_BASE_SLOTS
 osclbase, [31](#)
OSCL_TLS_EXTERNAL_SLOTS
 osclbase, [31](#)
OSCL_TLS_GET_FUNC
 osclconfig_unix_common.h, [812](#)
 osclconfig_unix_nj.h, [816](#)
OSCL_TLS_ID_BASE_LAST
 osclbase, [43](#)
OSCL_TLS_ID_ERRORHOOK
 osclbase, [43](#)
OSCL_TLS_ID_MAGICNUM
 osclbase, [43](#)
OSCL_TLS_ID_OSCLREGISTRY
 osclbase, [43](#)
OSCL_TLS_ID_PAYLOADPARSER
 osclbase, [43](#)
OSCL_TLS_ID_PVERRORTRAP
 osclbase, [43](#)
OSCL_TLS_ID_PVLOGGER
 osclbase, [43](#)
OSCL_TLS_ID_PVMFRECOGNIZER
 osclbase, [43](#)
OSCL_TLS_ID_PVSCHEDULER
 osclbase, [43](#)
OSCL_TLS_ID_SDPMEDIAPARSER
 osclbase, [43](#)
OSCL_TLS_ID_SQLITE3
 osclbase, [43](#)
OSCL_TLS_ID_TEST
 osclbase, [43](#)
OSCL_TLS_ID_WMDRM
 osclbase, [43](#)
OSCL_TLS_IS_KEYED
 osclconfig_unix_common.h, [812](#)
 osclconfig_unix_nj.h, [816](#)
OSCL_TLS_KEY_CREATE_FUNC
 osclconfig_unix_common.h, [812](#)
 osclconfig_unix_nj.h, [816](#)
OSCL_TLS_KEY_DELETE_FUNC
 osclconfig_unix_common.h, [812](#)
 osclconfig_unix_nj.h, [816](#)
OSCL_TLS_MAX_SLOTS
 osclbase, [31](#)
OSCL_TLS_STORE_FUNC
 osclconfig_unix_common.h, [812](#)
 osclconfig_unix_nj.h, [816](#)
oscl_tolower
 osclbase, [41](#)
OSCL_TRAP_ALLOC_NEW
 osclmemory, [54](#)
OSCL_TRAP_AUDIT_NEW
 osclmemory, [55](#)
OSCL_TRAP_NEW
 osclmemory, [55](#)
OSCL_TRAPSTACK_POP
 osclerror, [86](#)
OSCL_TRAPSTACK_POPDEALLOC
 osclerror, [86](#)
OSCL_TRAPSTACK_PUSH
 osclerror, [86](#)
oscl_tree.h, [770](#)
OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
[770](#)
OSCL_TRY
 osclerror, [86](#)
OSCL_TRY_NO_TLS
 osclerror, [86](#)
OSCL_TStrPtrLen
 osclutil, [66](#)
oscl_types.h, [771](#)
oscl_udp_socket.h, [772](#)
oscl_UnicodeToUTF8
 osclutil, [76](#)
OSCL_UNSIGNED_CONST
 osclbase, [31](#)
 osclconfig.h, [777](#)
OSCL_UNUSED_ARG
 osclbase, [31](#)
OSCL_UNUSED_RETURN
 osclbase, [32](#)
oscl_utf8conv.h, [773](#)
oscl_UTF8ToUnicode
 osclutil, [76](#)
oscl_uuid.h, [774](#)
 BYTES_IN_UUID_ARRAY, [774](#)
 OsclUid32, [774](#)
 PV_CHAR_CLOSE_BRACKET, [774](#)
 PV_CHAR_COMMA, [774](#)
Oscl_Vector, [273](#)
 ~Oscl_Vector, [274](#)
 back, [275](#)
 begin, [275](#)
 clear, [275](#)
 const_iterator, [274](#)
 const_reference, [274](#)
 destroy, [275](#)
 end, [275](#)

erase, 275
 front, 276
 insert, 276
 iterator, 274
 operator=, 276
 operator[], 276
 Oscl_Vector, 274
 pointer, 274
 pop_back, 276
 push_back, 277
 push_front, 277
 reference, 274
 value_type, 274
 oscl_vector.h, 775
 Oscl_Vector_Base, 278
 ~Oscl_Vector_Base, 279
 assign_vector, 279
 bufsize, 281
 capacity, 279
 construct, 279
 destroy, 279
 elems, 281
 empty, 279
 erase, 279, 280
 insert, 280
 numelems, 281
 OsclPriorityQueueBase, 281
 pop_back, 280
 push_back, 280
 push_front, 280
 reserve, 281
 size, 281
 sizeof_T, 281
 oscl_vsnprintf
 osclutil, 77, 79
 oscl_wchar
 osclbase, 32
 OSCL_wFastString, 282
 OSCL_wFastString, 283
 OSCL_wFastString
 ~OSCL_wFastString, 283
 chartype, 282
 get_cstr, 283
 get_maxsize, 283
 get_size, 283
 get_str, 283
 operator=, 283
 OSCL_wFastString, 283
 OSCL_wString, 283
 set, 283
 set_length, 283
 OSCL_wHeapString, 285
 osclutil, 79
 OSCL_wHeapString
 chartype, 285
 get_cstr, 285
 get_maxsize, 285
 get_size, 285
 get_str, 285
 operator=, 285
 OSCL_wHeapStringA, 285
 OSCL_wString, 285
 set, 285
 set_length, 285
 OSCL_wString
 chartype, 285
 OSCL_wString, 286
 OSCL_wHeapStringA, 287
 OSCL_wHeapStringA, 288
 OSCL_wHeapStringA
 ~OSCL_wHeapStringA, 288
 chartype, 288
 get_cstr, 288
 get_maxsize, 288
 get_size, 288
 get_str, 288
 operator=, 288, 289
 OSCL_wHeapStringA, 288
 OSCL_wString, 289
 set, 289
 OSCL_WRITESET_FLAG
 oscl_socket_serv_imp_pv.h, 743
 OSCL_wStackString, 290
 osclutil, 79
 OSCL_wStackString
 chartype, 291
 OSCL_wString, 291
 OSCL_wString, 292
 OSCL_wFastString, 283
 OSCL_wHeapString, 286
 OSCL_wHeapStringA, 289
 OSCL_wStackString, 291
 OSCL_wString, 293
 OSCL_wString
 ~OSCL_wString, 293
 append_rep, 293
 chartype, 293
 get_cstr, 293
 get_maxsize, 293
 get_size, 293
 get_str, 293
 hash, 293
 is_writable, 294
 operator!=, 294
 operator+=, 294
 operator<, 294
 operator<=, 294
 operator=, 294
 operator==, 294
 operator>, 294
 operator>=, 294
 operator[], 294
 OSCL_wString, 293
 read, 294
 set_len, 295
 set_rep, 295
 write, 295
 OSCL_ZEROIZE
 osclproc, 101

OsclAccept
osclconfig_io.h, 787
OsclAcceptMethod, 296
OsclAcceptMethod
~OsclAcceptMethod, 296
Accept, 296
AcceptRequest, 296
DiscardAcceptedSocket, 296
GetAcceptedSocket, 296
NewL, 296
OsclAcceptRequest, 297
OsclAcceptRequest, 297
OsclSocketI, 519
OsclAcceptRequest
Accept, 297
OsclAcceptRequest, 297
OsclActiveObject, 298
EPriorityHigh, 299
EPriorityHighest, 299
EPriorityIdle, 299
EPriorityLow, 299
EPriorityNominal, 299
OsclActiveObject, 299
OsclExecSchedulerCommonBase, 385
PVActiveBase, 591
PVActiveStats, 592
PVThreadContext, 611
OsclActiveObject
~OsclActiveObject, 299
AddToScheduler, 299
Cancel, 299
DoCancel, 300
IsBusy, 300
OsclActiveObject, 299
OsclActivePriority, 299
PendComplete, 300
PendForExec, 300
Priority, 300
RemoveFromScheduler, 300
RunError, 300
RunIfNotReady, 301
SetBusy, 301
SetStatus, 301
Status, 301
StatusRef, 301
OsclActivePriority
OsclActiveObject, 299
OsclAllocDestructDealloc, 302
OsclAllocDestructDealloc
~OsclAllocDestructDealloc, 302
OsclAny
osclbase, 32
OsclAOStatus, 303
OsclAOStatus, 303

OsclAOStatus
operator!=, 303
operator<, 303
operator<=, 303
operator=, 303
operator==, 303
operator>, 303
operator>=, 303
OsclAOStatus, 303
Value, 303
OsclAsyncFile, 304
OsclAsyncFile
~OsclAsyncFile, 305
Close, 305
Delete, 305
EndOfFile, 305
Flush, 305
iNumOfRun, 306
iNumOfRunErr, 306
NewL, 305
Open, 305, 306
Read, 306
Seek, 306
Size, 306
Tell, 306
Write, 306
OsclAsyncFileBuffer, 307
OsclAsyncFileBuffer
~OsclAsyncFileBuffer, 308
Buffer, 308
CleanInUse, 308
HasThisOffset, 308
Id, 308
IsInUse, 308
IsValid, 308
Length, 308
NewL, 308
Offset, 308
SetInUse, 308
SetOffset, 308
StartAsyncRead, 308
UpdateData, 308
OsclAuditCB, 309
OsclAuditCB, 309
OsclAuditCB
OsclAuditCB, 309
pAudit, 309
pStatsNode, 309
OsclBase
OsclSingletonRegistry, 515
OsclTLSRegistry, 574
osclbase
_OSCL_Abort, 33
ALLOC_AND_CONSTRUCT, 30

ALLOCATE, 30
 big_endian_to_host, 33
 Bind, 33
 c_bool, 32
 CTIME_BUFFER_SIZE, 43
 CtimeStrBuf, 32
 host_to_big_endian, 33
 host_to_little_endian, 34
 int64, 32
 little_endian_to_host, 34
 mbchar, 32
 MICROSECONDS, 33
 MILLISECONDS, 33
 MSEC_PER_SEC, 43
 NULL, 30
 NULL_TERM_CHAR, 30
 octet, 32
 operator-, 34
 operator==, 34
 OSCL_ABS, 30
 OSCL_ASSERT, 30
 OSCL Assert, 34
 OSCL_BEGIN_PACKED, 30
 oscl_CIstrcmp, 34, 35
 oscl_CIstrncmp, 35
 OSCL_COND_EXPORT_REF, 30
 OSCL_COND_IMPORT_REF, 30
 OSCL_CONST_CAST, 30
 OSCL_DISABLE_WARNING_-
 RETURN_TYPE_NOT_UDT, 30
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 30
 OSCL_DLL_ENTRY_POINT, 30
 OSCL_DLL_ENTRY_POINT_DEFAULT,
 31
 OSCL_DYNAMIC_CAST, 31
 OSCL_END_PACKED, 31
 OSCL_HAS_SINGLETON_SUPPORT, 31
 OSCL_INLINE, 31
 oscl_isLetter, 35
 OSCL_MAX, 31
 OSCL_MIN, 31
 OSCL_PACKED_VAR, 31
 OSCL_REINTERPRET_CAST, 31
 OSCL_STATIC_CAST, 31
 oscl_streat, 36
 oscl_strchr, 36, 37
 oscl_strcmp, 37
 oscl_strlen, 37
 oscl_strncat, 38
 oscl_strncmp, 38, 39
 oscl_strncpy, 39
 oscl strrchr, 40
 oscl_strset, 40
 oscl strstr, 40, 41
 OSCL_TCHAR, 32
 OSCL_TEMPLATED_DESTRUCTOR_-
 CALL, 31
 OSCL_TLS_BASE_SLOTS, 31
 OSCL_TLS_EXTERNAL_SLOTS, 31
 OSCL_TLS_ID_BASE_LAST, 43
 OSCL_TLS_ID_ERRORHOOK, 43
 OSCL_TLS_ID_MAGICNUM, 43
 OSCL_TLS_ID_OSCLREGISTRY, 43
 OSCL_TLS_ID_PAYLOADPARSER, 43
 OSCL_TLS_ID_PVERRORTRAP, 43
 OSCL_TLS_ID_PVLOGGER, 43
 OSCL_TLS_ID_PVMFRECOGNIZER, 43
 OSCL_TLS_ID_PVSCHEDULER, 43
 OSCL_TLS_ID_SDPMEDIAPARSER, 43
 OSCL_TLS_ID_SQLITE3, 43
 OSCL_TLS_ID_TEST, 43
 OSCL_TLS_ID_WMDRM, 43
 OSCL_TLS_MAX_SLOTS, 31
 oscl_tolower, 41
 OSCL_UNSIGNED_CONST, 31
 OSCL_UNUSED_ARG, 31
 OSCL_UNUSED_RETURN, 32
 oscl_wchar, 32
 OsclAny, 32
 OsclFloat, 32
 PV8601TIME_BUFFER_SIZE, 43
 PV8601timeStrBuf, 32
 PV8601ToRFC822, 41
 PVMEM_INST_LEVEL, 32
 PVOsclBase_Cleanup, 42
 PVOsclBase_Init, 42
 RFC822ToPV8601, 42
 SECONDS, 33
 TimeUnits, 33
 TOsclITlsKey, 33
 uint, 33
 uint64, 33
 unix_ntp_offset, 43
 USEC_PER_SEC, 43
 OsclBasicDateStruct
 osclconfig_time.h, 807
 OsclBasicTimeStruct
 osclconfig_time.h, 807
 OsclBind
 osclconfig_io.h, 787
 OsclBindMethod, 310
 OsclBindMethod
 ~OsclBindMethod, 310
 Bind, 310
 BindRequest, 310
 NewL, 310

OsclBindRequest, 311
 OsclBindRequest, 311
 OsclBindRequest
 Bind, 311
 OsclBindRequest, 311
 OsclBinIStream, 312
 OsclBinIStream, 312
 OsclBinIStream
 ~OsclBinIStream, 312
 get, 312
 OsclBinIStream, 312
 Read_uint8, 312
 OsclBinIStreamBigEndian, 314
 OsclBinIStreamBigEndian, 315
 OsclBinIStreamBigEndian
 operator>>, 315
 OsclBinIStreamBigEndian, 315
 Read, 315
 Read_uint16, 315
 Read_uint32, 315
 OsclBinIStreamLittleEndian, 317
 OsclBinIStreamLittleEndian, 318
 OsclBinIStreamLittleEndian
 operator>>, 318
 OsclBinIStreamLittleEndian, 318
 Read_uint16, 318
 Read_uint32, 318
 OsclBinOStream, 319
 OsclBinOStream, 319
 OsclBinOStream
 ~OsclBinOStream, 319
 OsclBinOStream, 319
 write, 319
 OsclBinOStreamBigEndian, 320
 OsclBinOStreamBigEndian, 321
 OsclBinOStreamBigEndian
 operator<<, 321
 OsclBinOStreamBigEndian, 321
 WriteUnsignedLong, 321
 WriteUnsignedShort, 321
 OsclBinOStreamLittleEndian, 322
 OsclBinOStreamLittleEndian, 323
 OsclBinOStreamLittleEndian
 operator<<, 323
 OsclBinOStreamLittleEndian, 323
 WriteUnsignedLong, 323
 WriteUnsignedShort, 323
 OsclBinStream, 324
 EOF_STATE, 325
 FAIL_STATE, 325
 GOOD_STATE, 325
 OsclBinStream, 325
 OsclBinStream
 Attach, 325
 eof, 325
 fail, 326
 firstFragPtr, 327
 fragsLeft, 327
 good, 326
 HaveRoomInCurrentBlock, 326
 length, 327
 nextFragPtr, 327
 numFrags, 327
 OsclBinStream, 325
 pBasePosition, 327
 PositionInBlock, 326
 pPosition, 327
 ReserveSpace, 326
 Seek, 326
 seekFromCurrentPosition, 326
 specialFragBuffer, 327
 state, 327
 state_t, 325
 tellg, 326
 OsclBuf, 328
 OsclBuf, 329
 OsclBuf
 Delete, 329
 Des, 329
 DesC, 329
 iBuffer, 329
 iLength, 329
 iMaxLength, 329
 Length, 329
 NewL, 329
 OsclBuf, 329
 OsclCloseSocket
 osclconfig_io.h, 788
 OsclCoeActiveScheduler
 OsclExecSchedulerBase, 379
 OsclExecSchedulerCommonBase, 385
 PVThreadContext, 611
 OsclCoeActiveSchedulerBase
 PVThreadContext, 611
 OsclCompareLess, 330
 OsclCompareLess
 compare, 330
 OsclComponentFactory
 osclutil, 66
 OsclComponentRegistry, 331
 OsclComponentRegistry, 332
 OsclComponentRegistry
 ~OsclComponentRegistry, 332
 CloseSession, 332
 FindExact, 332
 FindHierarchical, 332
 iComponentIdCounter, 332
 iData, 332

iMutex, [332](#)
 iNumSessions, [332](#)
 OpenSession, [332](#)
 OsclComponentRegistry, [332](#)
 Register, [332](#)
 Unregister, [332](#)
 OsclComponentRegistryData, [333](#)
 OsclComponentRegistryData
 Find, [333](#)
 iVec, [333](#)
 OsclComponentRegistryElement, [334](#)
 OsclComponentRegistryElement, [334](#)
 OsclComponentRegistryElement
 ~OsclComponentRegistryElement, [334](#)
 iComponentId, [334](#)
 iFactory, [334](#)
 iId, [334](#)
 Match, [334](#)
 operator=, [334](#)
 OsclComponentRegistryElement, [334](#)
osclconfig
 __int16_check__, [22](#)
 __int32_check__, [22](#)
 __int8_check__, [22](#)
 __uint16_check__, [22](#)
 __uint32_check__, [22](#)
 __uint8_check__, [22](#)
 OSCL_ASSERT_ALWAYS, [20](#)
 OSCL_BYTE_ORDER_BIG_ENDIAN, [20](#)
 OSCL_BYTE_ORDER_LITTLE_ENDIAN, [20](#)
 OSCL_HAS_BERKELEY_SOCKETS, [20](#)
 OSCL_HAS_MSWIN_PARTIAL_SUPPORT, [21](#)
 OSCL_HAS_MSWIN_SUPPORT, [21](#)
 OSCL_HAS_PTHREAD_SUPPORT, [21](#)
 OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS, [21](#)
 OSCL_HAS_PV_C_OS_SUPPORT, [21](#)
 OSCL_HAS_PV_C_OS_TIME_FUNCS, [21](#)
 OSCL_HAS_SAVAJE_IO_SUPPORT, [21](#)
 OSCL_HAS_SAVAJE_SUPPORT, [21](#)
 OSCL_HAS_SEM_TIMEDWAIT_SUPPORT, [21](#)
 OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION, [21](#)
 OSCL_HAS_SYMBIAN_DNS_SERVER, [21](#)
 OSCL_HAS_SYMBIAN_ERRORTRAP, [21](#)
 OSCL_HAS_SYMBIAN_MATH, [21](#)
 OSCL_HAS_SYMBIAN_MEMORY_FUNCS, [21](#)
 OSCL_HAS_SYMBIAN_SCHEDULER, [21](#)
 OSCL_HAS_SYMBIAN_SOCKET_SERVER, [21](#)
 OSCL_HAS_SYMBIAN_SUPPORT, [21](#)
 OSCL_HAS_SYMBIAN_TIMERS, [21](#)
 OSCL_HAS_UNIX_SUPPORT, [21](#)
 OSCL_HAS_UNIX_TIME_FUNCS, [21](#)
 OSCL_INTEGERS_WORD_ALIGNED, [21](#)
osclconfig.h, [776](#)
 __TFS__, [777](#)
 OSCL_BEGIN_PACKED, [777](#)
 OSCL_END_PACKED, [777](#)
 OSCL_HAS_NJ_FILE_IO_SUPPORT, [777](#)
 OSCL_HAS_NJ_SUPPORT, [777](#)
 OSCL_NATIVE_UINT64_TYPE, [777](#)
 OSCL_PACKED_STRUCT_BEGIN, [777](#)
 OSCL_PACKED_STRUCT_END, [777](#)
 OSCL_PACKED_VAR, [777](#)
 OSCL_RELEASE_BUILD, [777](#)
 OSCL_TEMPLATED_DESTRUCTOR_CALL, [777](#)
 OSCL_UNSIGNED_CONST, [777](#)
osclconfig_ansi_memory.h, [778](#)
 OSCL_HAS_ANSI_MEMORY_FUNCS, [778](#)
 oscl_memsize_t, [778](#)
osclconfig_check.h, [779](#)
osclconfig_compiler_warnings.h, [780](#)
 OSCL_FUNCTION_PTR, [780](#)
osclconfig_error.h, [781](#)
 OSCL_HAS_ERRNO_H, [781](#)
 OSCL_HAS_EXCEPTIONS, [781](#)
 OSCL_HAS_SETJMP_H, [781](#)
 OSCL_HAS_SYMBIAN_ERRORTRAP, [781](#)
osclconfig_error_check.h, [782](#)
osclconfig_global_new_delete.h, [783](#)
osclconfig_global_placement_new.h, [784](#)
 operator new, [784](#)
osclconfig_io.h, [785](#)
 OSCL_AF_INET, [787](#)
 OSCL_FILE_BUFFER_MAX_SIZE, [787](#)
 OSCL_HAS_ANSI_FILE_IO_SUPPORT, [787](#)
 OSCL_HAS_BERKELEY_SOCKETS, [787](#)
 OSCL_HAS_GLOB, [787](#)
 OSCL_HAS_NATIVE_FILE_CACHE_ENABLE, [787](#)

OSCL_HAS_PV_FILE_CACHE, 787
 OSCL_HAS_SOCKET_SUPPORT, 787
 OSCL_HAS_SYMBIAN_-
 COMPATIBLE_IO_FUNCTION,
 787
 OSCL_HAS_SYMBIAN_DNS_SERVER,
 787
 OSCL_HAS_SYMBIAN_SOCKET_-
 SERVER, 787
 OSCL_IPPROTO_TCP, 787
 OSCL_IPPROTO_UDP, 787
 OSCL_SD_BOTH, 787
 OSCL_SD_RECEIVE, 787
 OSCL_SD_SEND, 787
 OSCL SOCK_DGRAM, 787
 OSCL SOCK_STREAM, 787
 OsclAccept, 787
 OsclBind, 787
 OsclCloseSocket, 788
 OsclConnect, 788
 OsclConnectComplete, 788
 OsclGetAsyncSockErr, 788
 OsclGetDottedAddr, 788
 OsclGethostbyname, 788
 OsclJoin, 789
 OsclListen, 789
 OsclMakeSockAddr, 789
 OsclRecv, 789
 OsclRecvFrom, 789
 OsclSend, 790
 OsclSendTo, 790
 OsclSetNonBlocking, 790
 OsclSetRecvBufferSize, 790
 OsclShutdown, 790
 OsclSocket, 790
 OsclSocketCleanup, 791
 OsclSocketSelect, 791
 OsclSocketStartup, 791
 OsclUnMakeSockAddr, 791
 OsclValidInetAddr, 791
 TOsclHostent, 791
 TOsclSockAddr, 791
 TOsclSockAddrLen, 791
 TOsclSocket, 791
 osclconfig_io_check.h, 792
 osclconfig_ix86.h, 793
 osclconfig_lib.h, 794
 OSCL HAS_RUNTIME_LIB_-
 LOADING_SUPPORT, 794
 OSCL_LIB_READ_DEBUG_LIBS, 794
 PV_DYNAMIC_LOADING_CONFIG_-
 FILE_PATH, 794
 PV_RUNTIME_LIB_FILENAME_-
 EXTENSION, 794
 osclconfig_lib_check.h, 795
 osclconfig_limits_typedefs.h, 796
 OSCL_CHAR_IS_SIGNED, 796
 OSCL_CHAR_IS_UNSIGNED, 796
 osclconfig_memory.h, 797
 OSCL_BYPASS_MEMMGT, 797
 OSCL HAS_GLOBAL_NEW_DELETE,
 797
 OSCL HAS_HEAP_BASE_SUPPORT,
 797
 OSCL HAS_SYMBIAN_MEMORY_-
 FUNCS, 797
 PVMEM INST_LEVEL, 797
 osclconfig_memory_check.h, 798
 osclconfig_no_os.h, 799
 osclconfig_proc.h, 800
 osclconfig_proc_check.h, 801
 __verify__TOsclConditionObject__-
 defined__, 801
 __verify__TOsclMutexObject__defined__,
 801
 __verify__TOsclSemaphoreObject__-
 defined__, 801
 __verify__TOsclThreadFuncArg__-
 defined__, 801
 __verify__TOsclThreadFuncRet__-
 defined__, 801
 __verify__TOsclThreadId__defined__, 801
 __verify__TOsclThreadObject__defined__-
 __, 801
 osclconfig_proc_unix_common.h, 803
 OSCL HAS_NON_PREEMPTIVE_-
 THREAD_SUPPORT, 804
 OSCL HAS_PTHREAD_SUPPORT, 804
 OSCL HAS_SEM_TIMEDWAIT_-
 SUPPORT, 804
 OSCL HAS_SYMBIAN_SCHEDULER,
 804
 OSCL HAS_THREAD_SUPPORT, 804
 OSCL THREAD DECL, 804
 TOsclConditionObject, 804
 TOsclMutexObject, 804
 TOsclSemaphoreObject, 804
 TOsclThreadFuncArg, 804
 TOsclThreadFuncRet, 804
 TOsclThreadId, 804
 TOsclThreadObject, 804
 osclconfig_proc_unix_nj.h, 805
 OSCL HAS_NON_PREEMPTIVE_-
 THREAD_SUPPORT, 806
 OSCL HAS_PTHREAD_SUPPORT, 806
 OSCL HAS_SEM_TIMEDWAIT_-
 SUPPORT, 806

OSCL_HAS_SYMBIAN_SCHEDULER,
 806
 OSCL_HAS_THREAD_SUPPORT, 806
 OSCL_THREAD_DECL, 806
 TOsclConditionObject, 806
 TOsclMutexObject, 806
 TOsclSemaphoreObject, 806
 TOsclThreadFuncArg, 806
 TOsclThreadFuncRet, 806
 TOsclThreadId, 806
 TOsclThreadObject, 806
 osclconfig_time.h, 807
 OSCL_HAS_UNIX_TIME_FUNCS, 807
 OsclBasicDateStruct, 807
 OsclBasicTimeStruct, 807
 osclconfig_time_check.h, 808
 __Validate_BasicTimeDateStruct__, 808
 __Validate_BasicTimeStruct__, 808
 osclconfig_unix_common.h, 809
 __STRLIT, 812
 __STRLIT_CHAR, 812
 __STRLIT_WCHAR, 812
 INT64, 812
 INT64_HILO, 812
 OSCL_DISABLE_INLINES, 812
 OSCL_HAS_ANSI_MATH_SUPPORT,
 812
 OSCL_HAS_ANSI_STDIO_SUPPORT,
 812
 OSCL_HAS_ANSI_STDLIB_SUPPORT,
 812
 OSCL_HAS_ANSI_STRING_SUPPORT,
 812
 OSCL_HAS_ANSI_WIDE_STRING_-
 SUPPORT, 812
 OSCL_HAS_BASIC_LOCK, 812
 OSCL_HAS_GLOBAL_VARIABLE_-
 SUPPORT, 812
 OSCL_HAS_MSWIN_SUPPORT, 812
 OSCL_HAS_NATIVE_INT64_TYPE, 812
 OSCL_HAS_NATIVE_UINT64_TYPE,
 812
 OSCL_HAS_SYMBIAN_SUPPORT, 812
 OSCL_HAS_TLS_SUPPORT, 812
 OSCL_HAS_UNICODE_SUPPORT, 812
 OSCL_HAS_UNIX_SUPPORT, 812
 OSCL_MEMFRAG_PTR_BEFORE_LEN,
 812
 OSCL_NATIVE_INT64_TYPE, 812
 OSCL_NATIVE_UINT64_TYPE, 812
 OSCL_NATIVE_WCHAR_TYPE, 812
 OSCL_TLS_GET_FUNC, 812
 OSCL_TLS_IS_KEYED, 812
 OSCL_TLS_KEY_CREATE_FUNC, 812
 OSCL_TLS_KEY_DELETE_FUNC, 812
 OSCL_TLS_STORE_FUNC, 812
 TOsclBasicLockObject, 812
 TOsclItIsKey, 812
 UINT64, 812
 UINT64_HILO, 812
 osclconfig_unix_nj.h, 813
 __STRLIT, 816
 __STRLIT_CHAR, 816
 __STRLIT_WCHAR, 816
 INT64, 816
 INT64_HILO, 816
 OSCL_DISABLE_INLINES, 816
 OSCL_HAS_ANSI_MATH_SUPPORT,
 816
 OSCL_HAS_ANSI_STDIO_SUPPORT,
 816
 OSCL_HAS_ANSI_STDLIB_SUPPORT,
 816
 OSCL_HAS_ANSI_STRING_SUPPORT,
 816
 OSCL_HAS_ANSI_WIDE_STRING_-
 SUPPORT, 816
 OSCL_HAS_BASIC_LOCK, 816
 OSCL_HAS_GLOBAL_VARIABLE_-
 SUPPORT, 816
 OSCL_HAS_MSWIN_SUPPORT, 816
 OSCL_HAS_NATIVE_INT64_TYPE, 816
 OSCL_HAS_NATIVE_UINT64_TYPE,
 816
 OSCL_HAS_SYMBIAN_SUPPORT, 816
 OSCL_HAS_TLS_SUPPORT, 816
 OSCL_HAS_UNICODE_SUPPORT, 816
 OSCL_HAS_UNIX_SUPPORT, 816
 OSCL_MEMFRAG_PTR_BEFORE_LEN,
 816
 OSCL_NATIVE_INT64_TYPE, 816
 OSCL_NATIVE_UINT64_TYPE, 816
 OSCL_NATIVE_WCHAR_TYPE, 816
 OSCL_TLS_GET_FUNC, 816
 OSCL_TLS_IS_KEYED, 816
 OSCL_TLS_KEY_CREATE_FUNC, 816
 OSCL_TLS_KEY_DELETE_FUNC, 816
 OSCL_TLS_STORE_FUNC, 816
 TOsclBasicLockObject, 816
 TOsclItIsKey, 816
 UINT64, 816
 UINT64_HILO, 816
 osclconfig_util.h, 817
 OSCL_CLOCK_HAS_DRIFT_-
 CORRECTION, 817
 OSCL_HAS_SYMBIAN_MATH, 817
 OSCL_HAS_SYMBIAN_TIMERS, 817
 OSCL RAND_MAX, 817

SLEEP_ONE_SEC, 817
 osclconfig_util_check.h, 818
OsclConnect
 osclconfig_io.h, 788
OsclConnectComplete
 osclconfig_io.h, 788
OsclConnectMethod, 336
OsclConnectMethod
 ~OsclConnectMethod, 336
 Connect, 336
 ConnectRequest, 336
 NewL, 336
OsclConnectRequest, 337
 OsclConnectRequest, 337
 OsclSocketI, 519
OsclConnectRequest
 Connect, 337
 OsclConnectRequest, 337
OsclDestructDealloc, 338
OsclDestructDealloc
 destruct_and_dealloc, 338
OsclDNS, 339
 OsclSocketServ, 536
OsclDNS
 ~OsclDNS, 339
 CancelGetHostByName, 339
 GetHostByName, 340
 NewL, 340
 OsclDNSRequestAO, 340
OsclDNSI, 341
 OsclDNSRequestAO, 353
 OsclSocketServI, 538
OsclDNSI
 ~OsclDNSI, 341
 Close, 341
 DNSRequestParam, 342
 GetHostByName, 341
 GetHostByNameSuccess, 341
 NewL, 342
 Open, 342
 OsclDNSRequest, 342
OsclDNSIBase, 343
 OsclDNSIBase, 344
OsclDNSIBase
 ~OsclDNSIBase, 344
 CancelFxn, 344
 CancelGetHostByName, 344
 Close, 344
 GetHostByName, 344
 GetHostByNameSuccess, 344
 iAlloc, 344
 iSocketServ, 344
 IsReady, 344
 Open, 344
 OsclDNSIBase, 344
 OsclDNSRequest, 344
 OsclDNSRequestAO, 352
OsclDNSRequestAO
 Abort, 352
 ConstructL, 352
 DNSRequestParam, 353
 DoCancel, 352
 GetSocketError, 352
 iDNSI, 353
 iDNSMethod, 353
 iLogger, 353
 iSocketError, 353
 NewRequest, 352
 OsclDNSI, 353

OsclDNSMethod, 353
 OsclDNSRequest, 353
 OsclDNSRequestAO, 352
 RequestDone, 352
 Run, 352
 Serv, 352
 Success, 353
OsclDoubleLink, 354
 OsclDoubleLink, 354
OsclDoubleLink
 iNext, 354
 InsertAfter, 354
 InsertBefore, 354
 iPrev, 354
 OsclDoubleLink, 354
 Remove, 354
OsclDoubleList, 355
 OsclDoubleList, 355
OsclDoubleList
 Head, 355
 InsertHead, 355
 InsertTail, 355
 IsHead, 355
 IsTail, 355
 OsclDoubleList, 355
 Tail, 355
OsclDoubleListBase, 356
 OsclDoubleListBase, 357
OsclDoubleListBase
 getHead, 357
 getOffset, 357
 iHead, 357
 Insert, 357
 InsertHead, 357
 InsertTail, 357
 iOffset, 357
 IsEmpty, 357
 OsclDoubleListBase, 357
 Reset, 357
 SetOffset, 357
OsclDoubleRunner, 358
 OsclDoubleRunner, 358
OsclDoubleRunner
 iHead, 358
 iNext, 358
 iOffset, 358
 operator T *, 358
 operator++, 358
 operator-, 358
 OsclDoubleRunner, 358
 Set, 358
 SetToHead, 358
 SetToTail, 358
OsclErrAlreadyExists
 osclerror, 88
OsclErrAlreadyInstalled
 osclerror, 88
OsclErrArgument
 osclerror, 88
OsclErrBadHandle
 osclerror, 88
OsclErrBusy
 osclerror, 88
OsclErrCancelled
 osclerror, 88
OsclErrCorrupt
 osclerror, 88
OsclErrGeneral
 osclerror, 88
OsclErrInvalidState
 osclerror, 88
OsclErrNoHandler
 osclerror, 88
OsclErrNoMemory
 osclerror, 88
OsclErrNone
 osclerror, 88
OsclErrNoResources
 osclerror, 88
OsclErrNotInstalled
 osclerror, 88
OsclErrNotReady
 osclerror, 88
OsclErrNotSupported
 osclerror, 88
OsclError, 360
 OsclErrorTrapImp, 366
 OsclExecSchedulerCommonBase, 385
 OsclTrapStack, 577
OsclError
 Leave, 360
 LeaveIfError, 360
 LeaveIfNull, 360
 Pop, 360
 PopDealloc, 360, 361
 PushL, 361
osclerror
 _PV_TRAP, 84
 _PV_TRAP_NO_TLS, 84
 internalLeave, 84
 OSCL_BAD_ALLOC_EXCEPTION_-
 CODE, 84
 OSCL_CATCH, 84
 OSCL_CATCH_ANY, 85
 OSCL_ERR_NONE, 85
 OSCL_FIRST_CATCH, 85
 OSCL_FIRST_CATCH_ANY, 85
 OSCL_GetLastError, 89

OSCL_IsErrnoSupported, 89
 OSCL_JUMP_MAX_JUMP_MARKS, 85
 OSCL_LAST_CATCH, 85
 OSCL_LEAVE, 86
 OSCL_MAX_TRAP_LEVELS, 86
 OSCL_SetLastError, 89
 OSCL_StrError, 89
 OSCL_TRAPSTACK_POP, 86
 OSCL_TRAPSTACK_POPDEALLOC, 86
 OSCL_TRAPSTACK_PUSH, 86
 OSCL_TRY, 86
 OSCL_TRY_NO_TLS, 86
 OsclErrAlreadyExists, 88
 OsclErrAlreadyInstalled, 88
 OsclErrArgument, 88
 OsclErrBadHandle, 88
 OsclErrBusy, 88
 OsclErrCancelled, 88
 OsclErrCorrupt, 88
 OsclErrGeneral, 88
 OsclErrInvalidState, 88
 OsclErrNoHandler, 88
 OsclErrNoMemory, 88
 OsclErrNone, 88
 OsclErrNoResources, 88
 OsclErrNotInstalled, 88
 OsclErrNotReady, 88
 OsclErrNotSupported, 88
 OsclErrOverflow, 88
 OsclErrSystemCallFailed, 88
 OsclErrThreadContextIncorrect, 88
 OsclErrTimeout, 88
 OsclErrUnderflow, 88
 OsclFailure, 88
 OsclLeaveCode, 89
 OsclPending, 88
 OsclReturnCode, 89
 OsclSuccess, 88
 OsclTrapOperation, 89
 PVError_DoLeave, 88
 PVERROR_IMP_JUMPS, 88
 PVERRORTRAP_REGISTRY, 88
 PVERRORTRAP_REGISTRY_ID, 89
 OsclErrorAllocator, 362
 OsclErrorAllocator, 362
 OsclErrorAllocator
 allocate, 362
 deallocate, 362
 operator delete, 363
 operator new, 363
 OsclErrorAllocator, 362
 OsclErrorTrap, 364
 OsclErrorTrapImp, 366
 OsclTrapStack, 577
 OsclErrorTrap
 Cleanup, 364
 GetErrorTrapImp, 364
 Init, 364
 OsclErrorTrapImp, 365
 OsclJump, 401
 OsclTrapStack, 577
 OsclErrorTrapImp
 CPVInterfaceProxy, 366
 iJumpData, 366
 iLeave, 366
 iTrapStack, 366
 OsclError, 366
 OsclErrorTrap, 366
 OsclExecScheduler, 366
 OsclExecSchedulerCommonBase, 366
 OsclJump, 366
 OsclJumpMark, 366
 OsclScheduler, 366
 OsclTrapStack, 366
 Trap, 365
 TrapNoTls, 365
 UnTrap, 365
 OsclErrOverflow
 osclerror, 88
 OsclErrSystemCallFailed
 osclerror, 88
 OsclErrThreadContextIncorrect
 osclerror, 88
 OsclErrTimeout
 osclerror, 88
 OsclErrUnderflow
 osclerror, 88
 OsclException, 367
 OsclException, 367
 OsclException
 getLeaveCode, 367
 OsclException, 367
 OsclExclusiveArrayPtr, 368
 OsclExclusiveArrayPtr, 369
 OsclExclusiveArrayPtr
 ~OsclExclusiveArrayPtr, 369
 _Ptr, 370
 get, 369
 operator *, 369
 operator->, 369
 operator=, 369
 OsclExclusiveArrayPtr, 369
 release, 370
 set, 370
 OsclExclusivePtr, 371
 OsclExclusivePtr, 372
 OsclExclusivePtr
 ~OsclExclusivePtr, 372

_Ptr, 373
 get, 372
 operator *, 372
 operator->, 372
 operator=, 372
 OsclExclusivePtr, 372
 release, 373
 set, 373
OsclExclusivePtrA, 374
 OsclExclusivePtrA, 375
OsclExclusivePtrA
 ~OsclExclusivePtrA, 375
 _Ptr, 376
 get, 375
 operator *, 375
 operator->, 375
 operator=, 375
 OsclExclusivePtrA, 375
 release, 376
 set, 376
OsclExecScheduler, 377
 OsclErrorTrapImp, 366
 OsclExecSchedulerBase, 379
 OsclExecSchedulerCommonBase, 385
 PVActiveBase, 591
 PVActiveStats, 592
 PVThreadContext, 611
OsclExecScheduler
 Current, 377
 OsclScheduler, 378
 RegisterForCallback, 377
 RunSchedulerNonBlocking, 377
OsclExecSchedulerBase, 379
 PVThreadContext, 611
OsclExecSchedulerBase
 OsclCoeActiveScheduler, 379
 OsclExecScheduler, 379
 PVActiveBase, 379
OsclExecSchedulerCommonBase, 380
 EOtherExecStats_Last, 382
 EOtherExecStats_NativeOS, 382
 EOtherExecStats_QueueTime, 382
 EOtherExecStats_ReleaseTime, 382
 EOtherExecStats_WaitTime, 382
 OsclErrorTrapImp, 366
 OsclExecSchedulerCommonBase, 383
 PVActiveStats, 592
 PVThreadContext, 611
OsclExecSchedulerCommonBase
 ~OsclExecSchedulerCommonBase, 383
 AddToExecTimerQ, 383
 BeginScheduling, 383
 BeginStats, 383
 BlockingLoopL, 383
 CallRunExec, 383
 CleanupExecQ, 383
 CleanupStatQ, 383
 ConstructL, 383
 ConstructStatQ, 383
 EndScheduling, 383
 EndStats, 383
 Error, 383
 FindPVBase, 383
 GetId, 383
 GetName, 383
 GetScheduler, 383
 iAlloc, 387
 iBlockingMode, 387
 iDebugLogger, 387
 iDefAlloc, 387
 iDelta, 387
 iDoStop, 387
 iDoSuspend, 387
 iErrorTrapImp, 387
 iExecTimerQ, 387
 iGrandTotalTicks, 387
 iLogger, 387
 iLogPerfIndentStr, 387
 iLogPerfIndentStrLen, 387
 iLogPerfTotal, 387
 iName, 387
 iNativeMode, 387
 IncLogPerf, 384
 InitExecQ, 384
 InstallScheduler, 384
 iNumAOAdded, 387
 iOtherExecStats, 387
 iPVStatQ, 387
 iPVStats, 387
 iReadyQ, 387
 iResumeSem, 387
 IsInstalled, 384
 Started, 384
 iStopper, 387
 iStopperCrit, 387
 iSuspended, 387
 iThreadContext, 387
 iTime, 387
 iTimeCompareThreshold, 387
 iTTotalPercent, 387
 iTTotalTicksTemp, 387
 OsclActiveObject, 385
 OsclCoeActiveScheduler, 385
 OsclError, 385
 OsclExecScheduler, 385
 OsclExecSchedulerCommonBase, 383
 OsclReadyQ, 385
 OsclScheduler, 385

OsclTimerCompare, 385
 OsclTimerObject, 387
 PendComplete, 384
 PVActiveBase, 387
 PVActiveStats, 387
 PVSchedulerStopper, 387
 PVThreadContext, 387
 RequestCanceled, 384
 ResetLogPerf, 384
 ResumeScheduler, 384
 SetScheduler, 384
 ShowStats, 384
 ShowSummaryStats, 384
 StartNativeScheduler, 384
 StartScheduler, 384
 StopScheduler, 384
 SuspendScheduler, 385
 TOtherExecStats, 382
 UninstallScheduler, 385
 UpdateTimers, 385
 UpdateTimersMsec, 385
 WaitForReadyAO, 385

OsclFailure
 osclerror, 88

OsclFileCache, 389
 Oscl_File, 180
 OsclFileCache, 390

OsclFileCache
 ~OsclFileCache, 390
 Close, 390
 EndOfFile, 390
 FileSize, 390
 Flush, 390
 Open, 390
 OsclFileCache, 390
 Read, 390
 Seek, 390
 Tell, 390
 Write, 390

OsclFileHandle, 391
 OsclFileHandle, 391

OsclFileHandle
 Handle, 391
 Oscl_File, 391
 OsclFileHandle, 391

OsclFileStats, 392
 OsclFileStats, 392

OsclFileStats
 End, 392
 Log, 392
 LogAll, 392
 OsclFileStats, 392
 Start, 392

OsclFileStatsItem, 393

 OsclFileStatsItem
 iOpCount, 393
 iParam, 393
 iParam2, 393
 iStartTick, 393
 iTotalsTicks, 393

OsclFloat
 osclbase, 32

OsclGetAsyncSockErr
 osclconfig_io.h, 788

OsclGetDottedAddr
 osclconfig_io.h, 788

OsclGetHostbyname
 osclconfig_io.h, 788

OsclGetHostByNameMethod, 394
 OsclGetHostByNameRequest, 395

OsclGetHostByNameMethod
 ~OsclGetHostByNameMethod, 394
 GetHostName, 394
 NewL, 394

OsclGetHostByNameRequest, 395
 OsclDNSIBase, 344

OsclGetHostByNameRequest
 OsclGetHostByNameMethod, 395

OsclInit, 396

OsclInit
 Cleanup, 396
 Init, 396

OsclInteger64Transport, 397

OsclInteger64Transport
 iHigh, 397
 iLow, 397

osclio
 EOscIFileOp_Close, 94
 EOscIFileOp_EndOfFile, 94
 EOscIFileOp_Flush, 94
 EOscIFileOp_Last, 94
 EOscIFileOp_NativeClose, 94
 EOscIFileOp_NativeEndOfFile, 94
 EOscIFileOp_NativeFlush, 94
 EOscIFileOp_NativeOpen, 94
 EOscIFileOp_NativeRead, 94
 EOscIFileOp_NativeSeek, 94
 EOscIFileOp_NativeSize, 94
 EOscIFileOp_NativeTell, 94
 EOscIFileOp_NativeWrite, 94
 EOscIFileOp_Open, 94
 EOscIFileOp_Read, 94
 EOscIFileOp_Seek, 94
 EOscIFileOp_Size, 94
 EOscIFileOp_Tell, 94
 EOscIFileOp_Write, 94
 EPVDNSCancel, 95
 EPVDNSFailure, 95

EPVDNSGetHostByName, 95
 EPVDNSPending, 95
 EPVDNSSuccess, 95
 EPVDNSTimeout, 95
 oscl_chdir, 95
 OSCL_FILE_CHAR_PATH_-
 DELIMITER, 93
 OSCL_FILE_STATS_LOGGER_NODE,
 93
 OSCL_FILE_WCHAR_PATH_-
 DELIMITER, 93
 OSCL_FILEMGMT_E_ALREADY_-
 EXISTS, 93
 OSCL_FILEMGMT_E_NO_MATCH, 94
 OSCL_FILEMGMT_E_NOT_EMPTY, 94
 OSCL_FILEMGMT_E_NOT_-
 IMPLEMENTED, 94
 OSCL_FILEMGMT_E_OK, 93
 OSCL_FILEMGMT_E_PATH_NOT_-
 FOUND, 93
 OSCL_FILEMGMT_E_PATH_TOO_-
 LONG, 93
 OSCL_FILEMGMT_E_PERMISSION_-
 DENIED, 94
 OSCL_FILEMGMT_E_SYS_SPECIFIC,
 94
 OSCL_FILEMGMT_E_UNKNOWN, 94
 OSCL_FILEMGMT_ERR_TYPE, 93
 OSCL_FILEMGMT_MODE_DIR, 94
 OSCL_FILEMGMT_MODES, 94
 OSCL_FILEMGMT_PERMS, 94
 OSCL_FILEMGMT_PERMS_EXECUTE,
 94
 OSCL_FILEMGMT_PERMS_READ, 94
 OSCL_FILEMGMT_PERMS_WRITE, 94
 OSCL_FSSTAT, 93
 oscl_getcwd, 95
 OSCL_IO_EXTENSION_MAXLEN, 93
 OSCL_IO_FILENAME_MAXLEN, 93
 oscl_mkdir, 96
 oscl_rename, 96
 oscl_rmdir, 97
 oscl_stat, 97
 OSCL_STAT_BUF, 93
 oscl_statfs, 98
 TOsclFileHandle, 93
 TOsclFileOp, 94
 TPVDNSEvent, 94
 TPVDNSFxn, 95
 OsclIPSocketI, 398
 OsclIPSocketI, 399
 OsclIPSocketI
 ~OsclIPSocketI, 399
 Alloc, 399
 Bind, 399
 Close, 399
 ConstructL, 399
 GetRecvData, 399
 GetSendData, 399
 iAddress, 400
 iAlloc, 400
 iId, 400
 iLogger, 400
 iObserver, 400
 iSocket, 400
 iSocketServ, 400
 Join, 399
 OsclIPSocketI, 399
 OsclSocketMethod, 400
 OsclSocketRequestAO, 400
 SetRecvBufferSize, 400
 SocketServ, 400
 OsclJoin
 osclconfig_io.h, 789
 OsclJump, 401
 OsclErrorTrapImp, 366
 OsclJump
 ~OsclJump, 401
 Jump, 401
 OsclErrorTrapImp, 401
 StaticJump, 401
 Top, 401
 OsclJumpMark
 OsclErrorTrapImp, 366
 OsclLeaveCode
 osclerror, 89
 OsclListen
 osclconfig_io.h, 789
 OsclListenMethod, 402
 OsclListenMethod
 ~OsclListenMethod, 402
 Listen, 402
 ListenRequest, 402
 NewL, 402
 OsclListenRequest, 403
 OsclListenRequest, 403
 OsclListenRequest
 Listen, 403
 OsclListenRequest, 403
 OsclLockBase, 404
 OsclLockBase
 ~OsclLockBase, 404
 Lock, 404
 Unlock, 404
 OsclMakeSockAddr
 osclconfig_io.h, 789
 OsclMem, 405
 OsclMemGlobalAuditObject, 421

OsclMem
 Cleanup, 405
 Init, 405
 OsclMemAllocator, 406
 OsclMemAllocator
 allocate, 406
 allocate_fl, 406
 deallocate, 406
 OsclMemAllocDestructDealloc, 407
 OsclMemAllocDestructDealloc
 allocate, 407
 allocate_fl, 407
 deallocate, 407
 destruct_and_dealloc, 407
 OsclMemAudit, 409
 OsclMemAudit, 409
 OsclMemAudit
 ~OsclMemAudit, 409
 GetLock, 410
 MM_AddTag, 410
 MM_allocate, 410
 MM_CreateAllocNodeInfo, 410
 MM_deallocate, 410
 MM_GetAllocNo, 410
 MM_GetAllocNodeInfo, 410
 MM_GetExistingTag, 411
 MM_GetMode, 411
 MM_GetNumAllocNodes, 411
 MM_GetOverheadStats, 411
 MM_GetPostfillPattern, 411
 MM_GetPrefillPattern, 411
 MM_GetRefCount, 411
 MM_GetRootNode, 412
 MM_GetStats, 412
 MM_GetStatsInDepth, 412
 MM_GetTagName, 412
 MM_GetTreeNodes, 412
 MM_ReleaseAllocNodeInfo, 412
 MM_SetFailurePoint, 412
 MM_SetMode, 413
 MM_SetPostfillPattern, 413
 MM_SetPrefillPattern, 413
 MM_SetTagLevel, 413
 MM_UnsetFailurePoint, 413
 MM_Validate, 413
 OsclMemAudit, 409
 OsclMemGlobalAuditObject, 414
 OSCLMemAutoPtr, 415
 OSCLMemAutoPtr, 416
 OSCLMemAutoPtr
 ~OSCLMemAutoPtr, 416
 _Ownership, 418
 allocate, 417
 deallocate, 417
 get, 417
 operator *, 417
 operator->, 417
 operator=, 417
 OSCLMemAutoPtr, 416
 release, 417
 setWithoutOwnership, 417
 takeOwnership, 418
 OsclMemBasicAllocator, 419
 OsclMemBasicAllocator
 allocate, 419
 deallocate, 419
 OsclMemBasicAllocDestructDealloc, 420
 OsclMemBasicAllocDestructDealloc
 allocate, 420
 deallocate, 420
 destruct_and_dealloc, 420
 OsclMemGlobalAuditObject, 421
 OsclMemAudit, 414
 OsclMemGlobalAuditObject
 audit_type, 421
 getGlobalMemAuditObject, 421
 OsclMem, 421
 OsclMemInit
 osclmemory, 59
 osclmemory
 _OSCL_CLEANUP_BASE_CLASS, 47
 _OSCL_TRAP_NEW, 47
 _oscl_audit_calloc, 56
 _oscl_audit_free, 56
 _oscl_audit_malloc, 56
 _oscl_audit_new, 56
 _oscl_audit_realloc, 57
 _oscl_calloc, 57
 _oscl_default_audit_calloc, 57
 _oscl_default_audit_malloc, 57
 _oscl_default_audit_new, 57
 _oscl_default_audit_realloc, 57
 _oscl_free, 57
 _oscl_malloc, 57
 _oscl_realloc, 57
 ALLOC_NODE_FLAG, 59
 COMPUTE_MEM_ALIGN_SIZE, 48
 DEFAULT_MM_AUDIT_MODE, 49
 DEFAULT_POSTFILL_PATTERN, 49
 DEFAULT_PREFILL_PATTERN, 49
 FENCE_PATTERN, 49
 MEM_ALIGN_SIZE, 49
 MIN_FENCE_SIZE, 49
 MM_ALLOC_MAX_QUERY_-
 FILENAME_LEN, 49
 MM_ALLOC_MAX_QUERY_TAG_LEN,
 49
 MM_AllocNodeAutoPtr, 56

MM_AUDIT_ALLOC_NODE_-
 ENABLE_FLAG, 49
 MM_AUDIT_ALLOC_NODE_-
 SUPPORT, 49
 MM_AUDIT_FAILURE_SIMULATION_-
 SUPPORT, 49
 MM_AUDIT_FENCE_SUPPORT, 49
 MM_AUDIT_FILL_SUPPORT, 49
 MM_AUDIT_INCLUDE_ALL_HEAP_-
 VALIDATION, 49
 MM_AUDIT_POSTFILL_FLAG, 49
 MM_AUDIT_PREFILL_FLAG, 49
 MM_AUDIT_SUPPRESS_FILENAME_-
 FLAG, 49
 MM_AUDIT_VALIDATE_ALL_HEAP_-
 FLAG, 49
 MM_AUDIT_VALIDATE_BLOCK, 49
 MM_AUDIT_VALIDATE_ON_FREE_-
 FLAG, 49
 MM_StatsNodeTagTreeType, 56
 MMAuditCharAutoPtr, 56
 MMAuditUint8AutoPtr, 56
 operator delete, 57
 operator delete[], 57
 operator new, 57
 operator new[], 57
 OSCL_ALLOC_DELETE, 49
 OSCL_ALLOC_NEW, 50
 OSCL_ARRAY_DELETE, 50
 OSCL_ARRAY_NEW, 50
 OSCL_AUDIT_ARRAY_NEW, 50
 OSCL_AUDIT_CALLOC, 51
 OSCL_AUDIT_MALLOC, 51
 OSCL_AUDIT_NEW, 51
 OSCL_AUDIT_REALLOC, 52
 OSCL_CALLOC, 52
 oscl_calloc, 52
 OSCL_CLEANUP_BASE_CLASS, 52
 OSCL_DEFAULT_FREE, 53
 OSCL_DEFAULT_MALLOC, 53
 OSCL_DELETE, 53
 OSCL_DISABLE_WARNING_-
 RETURN_TYPE_NOT_UDT, 53
 OSCL_DISABLE_WARNING_-
 TRUNCATE_DEBUG_MESSAGE,
 53
 OSCL_FREE, 53
 oscl_free, 53
 OSCL_HAS_GLOBAL_NEW_DELETE,
 53
 OSCL_MALLOC, 54
 oscl_malloc, 54
 oscl_mem_aligned_size, 57
 oscl_memcpy, 58

oscl_memmove, 58
 oscl_memmove32, 58
 oscl_memset, 59
 OSCL_NEW, 54
 OSCL_PLACEMENT_NEW, 54
 OSCL_REALLOC, 54
 oscl_realloc, 54
 OSCL_TRAP_ALLOC_NEW, 54
 OSCL_TRAP_AUDIT_NEW, 55
 OSCL_TRAP_NEW, 55
 OsclMemInit, 59
 OsclMemStatsNodeAutoPtr, 56
 OsclTagTreeType, 56
 TagTree_Allocator, 56
 OsclMemoryFragment, 422
 OsclMemoryFragment
 len, 422
 ptr, 422
 OsclMemPoolAllocator, 423
 OsclMemPoolAllocator, 423
 OsclMemPoolAllocator
 ~OsclMemPoolAllocator, 423
 CreateMemPool, 423
 DestroyMemPool, 423
 oscl_mem_aligned_size, 423
 OsclMemPoolAllocator, 423
 OsclMemPoolFixedChunkAllocator, 424
 OsclMemPoolFixedChunkAllocator, 425
 OsclMemPoolFixedChunkAllocator
 ~OsclMemPoolFixedChunkAllocator, 425
 addRef, 425
 allocate, 425
 CancelFreeChunkAvailableCallback, 425
 createmempool, 425
 deallocate, 426
 destroymempool, 426
 enablenullpointerreturn, 426
 iCheckNextAvailableFreeChunk, 427
 iChunkSize, 427
 iChunkSizeMemAligned, 427
 iEnableNullPtrReturn, 427
 iFreeMemChunkList, 427
 iMemPool, 427
 iMemPoolAllocator, 427
 iNextAvailableContextData, 427
 iNumChunk, 427
 iObserver, 427
 iRefCount, 427
 notifyfreechunkavailable, 426
 OsclMemPoolFixedChunkAllocator, 425
 removeRef, 426
 OsclMemPoolFixedChunkAllocatorObserver,
 428

OsclMemPoolFixedChunkAllocatorObserver
 ~OsclMemPoolFixedChunkAllocatorObserver, 428
 freechunkavailable, 428
OsclMemPoolResizableAllocator, 429
 OsclMemPoolResizableAllocator, 430
OsclMemPoolResizableAllocator
 ~OsclMemPoolResizableAllocator, 430
 addnewmempoolbuffer, 430
 addRef, 430
 allocate, 431
 allocateblock, 431
 CancelFreeChunkAvailableCallback, 431
 CancelFreeMemoryAvailableCallback, 431
 deallocate, 431
 deallocateblock, 431
 destroyallmempoolbuffers, 431
 enablenullpointerreturn, 431
 findfreeblock, 432
 getAllocatedSize, 432
 getAvailableSize, 432
 getBufferSize, 432
 getLargestContiguousFreeBlockSize, 432
 getMemPoolBufferAllocatedSize, 432
 getMemPoolBufferSize, 432
 iBlockInfoAlignedSize, 434
 iBufferInfoAlignedSize, 434
 iCheckFreeMemoryAvailable, 434
 iCheckNextAvailable, 434
 iEnableNullPtrReturn, 434
 iExpectedNumBlocksPerBuffer, 434
 iFreeMemContextData, 434
 iFreeMemPoolObserver, 434
 iMaxNewMemPoolBufferSz, 434
 iMemPoolBufferAllocator, 434
 iMemPoolBufferList, 434
 iMemPoolBufferNumLimit, 434
 iMemPoolBufferSize, 434
 iNextAvailableContextData, 434
 iObserver, 434
 iRefCount, 434
 iRequestedAvailableFreeMemSize, 434
 iRequestedNextAvailableSize, 434
 memoryPoolBufferMgmtOverhead, 432
 notifyfreeblockavailable, 432
 notifyfreememoryavailable, 432
OsclMemPoolResizableAllocator, 430
 removeRef, 433
 setMaxSzForNewMemPoolBuffer, 433
 trim, 433
 validateblock, 433
OsclMemPoolResizableAllocator::MemPoolBlockInfo, 435

OsclMemPoolResizableAllocator::MemPool-
 BlockInfo
 iBlockBuffer, 435
 iBlockPostFence, 435
 iBlockPreFence, 435
 iBlockSize, 435
 iNextFreeBlock, 435
 iParentBuffer, 435
 iPrevFreeBlock, 435
OsclMemPoolResizableAllocator::MemPoolBufferInfo, 436
OsclMemPoolResizableAllocator::MemPool-
 BufferInfo
 iAllocatedSz, 436
 iBufferPostFence, 436
 iBufferPreFence, 436
 iBufferSize, 436
 iEndAddr, 436
 iNextFreeBlock, 436
 iNumOutstanding, 436
 iStartAddr, 436
OsclMemPoolResizableAllocatorMemoryObserver, 437
OsclMemPoolResizableAllocatorMemory-
 Observer
~OsclMemPoolResizableAllocatorMemoryObserver, 437
 freememoryavailable, 437
OsclMemPoolResizableAllocatorObserver, 438
OsclMemPoolResizableAllocatorObserver
~OsclMemPoolResizableAllocatorObserver, 438
 freeblockavailable, 438
OsclMemStatsNode, 439
 OsclMemStatsNode, 439
OsclMemStatsNode
~OsclMemStatsNode, 439
 operator delete, 439
 operator new, 439
 OsclMemStatsNode, 439
 pMMFIParam, 439
 pMMStats, 439
 reset, 439
 tag, 439
OsclMemStatsNodeAutoPtr
 osclmemory, 56
OsclMutex, 440
 OsclMutex, 440
OsclMutex
~OsclMutex, 440
 Close, 440
 Create, 440
 Lock, 441
 OsclMutex, 440

TryLock, 441
 Unlock, 441
OsclNameString, 442
 OsclNameString, 442
OsclNameString
 MaxLen, 442
 OsclNameString, 442
 Set, 442
 Str, 442
OsclNativeFile, 443
 Oscl_FileServer, 186
 OsclNativeFile, 444
OsclNativeFile
 ~OsclNativeFile, 444
 Close, 444
 EndOfFile, 444
 Flush, 444
 GetError, 444
 GetReadAsyncNumElements, 444
 HasAsyncRead, 444
 Mode, 444
 Open, 444
 OsclNativeFile, 444
 Read, 444
 ReadAsync, 444
 ReadAsyncCancel, 444
 Seek, 445
 Size, 445
 Tell, 445
 Write, 445
OsclNativeFileParams, 446
 OsclNativeFileParams, 446
OsclNativeFileParams
 iAsyncReadBufferSize, 446
 iNativeAccessMode, 446
 iNativeBufferSize, 446
 OsclNativeFileParams, 446
OsclNetworkAddress, 447
 OsclNetworkAddress, 447
OsclNetworkAddress
 ipAddr, 447
 operator==, 447
 OsclNetworkAddress, 447
 port, 447
OsclNoYieldMutex
 oscl_mutex.h, 699
OsclNullLock, 448
OsclNullLock
 ~OsclNullLock, 448
 Lock, 448
 Unlock, 448
OsclPending
 osclerror, 88
OsclPriorityLink, 449
 OsclPriorityLink
 iPriority, 449
OsclPriorityList, 450
 OsclPriorityList, 450
OsclPriorityList
 Head, 450
 Insert, 450
 IsHead, 450
 IsTail, 450
 OsclPriorityList, 450
 Tail, 450
OsclPriorityQueue, 451
 OsclPriorityQueue, 452
OsclPriorityQueue
 ~OsclPriorityQueue, 452
 c, 454
 comp, 454
 compare_EQ, 452
 compare_LT, 452
 const_reference, 452
 container_type, 452
 empty, 453
 find_heap, 453
 iterator, 452
 oscl_pqueue_test, 454
 OsclPriorityQueue, 452
 pop, 453
 pop_heap, 453
 push, 453
 push_heap, 453
 remove, 453
 reserve, 453
 size, 453
 swap, 453
 top, 453
 validate, 454
 value_type, 452
 vec, 454
OsclPriorityQueueBase, 455
 Oscl_Vector_Base, 281
OsclPriorityQueueBase
 ~OsclPriorityQueueBase, 455
 construct, 455
 find_heap, 455
 pop_heap, 455
 push_heap, 455
 remove, 455
osclproc
 EPVThreadContext_InThread, 102
 EPVThreadContext_NonOsclThread, 102
 EPVThreadContext_OsclThread, 102
 EPVThreadContext_Undetermined, 102
 OSCL_PERF_SUMMARY_LOGGING,
 101

OSCL_REQUEST_ERR_CANCEL, 102
 OSCL_REQUEST_ERR_GENERAL, 102
 OSCL_REQUEST_ERR_NONE, 102
 OSCL_REQUEST_PENDING, 102
 OSCL_ZEROIZE, 101
 OsclPtrAdd, 102
 OsclPtrSub, 102
 PV_SCHED_CHECK_Q, 101
 PV_SCHED_ENABLE_AO_STATS, 101
 PV_SCHED_ENABLE_LOOP_STATS,
 101
 PV_SCHED_ENABLE_PERF_-
 LOGGING, 101
 PV_SCHED_ENABLE_THREAD_-
 CONTEXT_CHECKS, 101
 PV_SCHED_FAIR_SCHEDULING, 101
 PV_SCHED_LOG_Q, 101
 PVEEXECNAMELEN, 101
 PVSCHEDNAMELEN, 101
 QUE_ITER_BEGIN, 101
 QUE_ITER_END, 101
 TOsclReady, 102
 TPVThreadContext, 102
 OsclProcStatus, 456
 ALREADY_SUSPENDED_ERROR, 456
 BAD_THREADID_ADDR_ERROR, 456
 EXCEED_MAX_COUNT_VARIABLE_-
 ERROR, 457
 EXCEED_MAX_SEM_COUNT_ERROR,
 457
 INVALID_ACCESS_ERROR, 457
 INVALID_ARGUMENT_ERROR, 457
 INVALID_FUNCTION_ERROR, 457
 INVALID_HANDLE_ERROR, 457
 INVALID_OPERATION_ERROR, 457
 INVALID_PARAM_ERROR, 456
 INVALID_POINTER_ERROR, 457
 INVALID_PRIORITY_ERROR, 456
 INVALID_THREAD_ERROR, 456
 INVALID_THREAD_ID_ERROR, 456
 MAX_THRDS_REACHED_ERROR, 456
 MUTEX_LOCKED_ERROR, 457
 NO_PERMISSION_ERROR, 456
 NOT_ENOUGH_MEMORY_ERROR, 456
 NOT_ENOUGH_RESOURCES_ERROR,
 456
 NOT_IMPLEMENTED, 457
 NOT_SUSPENDED_ERROR, 456
 OTHER_ERROR, 456
 OUTOFMEMORY_ERROR, 456
 PSHARED_ATTRIBUTE_SETTING_-
 ERROR, 457
 PSHARED_NOT_ZERO_ERROR, 457
 RELOCK_MUTEX_ERROR, 457

 SEM_NOT_SIGNALLED_ERROR, 457
 SUCCESS_ERROR, 456
 SYSTEM_RESOURCES_-
 UNAVAILABLE_ERROR, 457
 THREAD_1_INACTIVE_ERROR, 456
 THREAD_BLOCK_ERROR, 457
 THREAD_NOT_OWN_MUTEX_-
 ERROR, 457
 TOO_MANY_THREADS_ERROR, 456
 WAIT_ABANDONED_ERROR, 457
 WAIT_TIMEOUT_ERROR, 457
 OsclProcStatus
 eOsclProcError, 456
 OsclPtr, 458
 OsclPtr, 458
 OsclPtr
 Append, 458
 Length, 458
 OsclPtr, 458
 Ptr, 458
 Set, 458
 SetLength, 458
 Zero, 458
 OsclPtrAdd
 osclproc, 102
 OsclPtrC, 460
 OsclPtrC, 461
 OsclPtrC
 Left, 461
 Length, 461
 OsclPtrC, 461
 Ptr, 461
 Right, 461
 Set, 461
 SetLength, 461
 Zero, 461
 OsclPtrSub
 osclproc, 102
 OsclRand, 462
 OsclRand
 Rand, 462
 Seed, 462
 OsclReadyAlloc, 463
 OsclReadyAlloc
 allocate, 463
 allocate_fl, 463
 deallocate, 463
 OsclReadyCompare, 464
 PVActiveBase, 591
 OsclReadyCompare
 compare, 464
 OsclReadyQ, 465
 OsclExecSchedulerCommonBase, 385
 PVActiveBase, 591

PVActiveStats, 592
OsclReadyQ
 Callback, 466
 Construct, 466
 Depth, 466
 IsIn, 466
 PendComplete, 466
 PopTop, 466
 RegisterForCallback, 466
 Remove, 466
 ThreadLogoff, 466
 ThreadLogon, 466
 TimerCallback, 466
 Top, 466
 WaitAndPopTop, 466
 WaitForRequestComplete, 466
OsclReadySetPosition
 PVActiveBase, 591
OsclRecv
 osclconfig_io.h, 789
OsclRecvFrom
 osclconfig_io.h, 789
OsclRecvFromMethod, 467
OsclRecvFromMethod
 ~OsclRecvFromMethod, 467
 GetRecvData, 467
 NewL, 467
 RecvFrom, 467
 RecvFromRequest, 467
OsclRecvFromRequest, 469
 OsclRecvFromRequest, 469
 OsclSocketI, 519
OsclRecvFromRequest
 GetRecvData, 469
 OsclRecvFromRequest, 469
 RecvFrom, 469
 Success, 469
OsclRecvMethod, 471
OsclRecvMethod
 ~OsclRecvMethod, 471
 GetRecvData, 471
 NewL, 471
 Recv, 471
 RecvRequest, 471
OsclRecvRequest, 472
 OsclRecvRequest, 472
 OsclSocketI, 519
OsclRecvRequest
 GetRecvData, 472
 OsclRecvRequest, 472
 Recv, 472
 Success, 472
OsclRefCounter, 473
OsclRefCounter
 ~OsclRefCounter, 473
 addRef, 473
 getCount, 473
 removeRef, 473
OsclRefCounterDA, 475
 OsclRefCounterDA, 475
OsclRefCounterDA
 ~OsclRefCounterDA, 475
 addRef, 476
 getCount, 476
 OsclRefCounterDA, 475
 removeRef, 476
OsclRefCounterMemFrag, 477
 OsclRefCounterMemFrag, 477
OsclRefCounterMemFrag
 ~OsclRefCounterMemFrag, 477
 getCapacity, 478
 getCount, 478
 getMemFrag, 478
 getMemFragPtr, 478
 getMemFragSize, 478
 getRefCounter, 478
 operator=, 478
 OsclRefCounterMemFrag, 477
OsclRefCounterMTDA, 479
 OsclRefCounterMTDA, 479
OsclRefCounterMTDA
 ~OsclRefCounterMTDA, 479
 addRef, 480
 getCount, 480
 OsclRefCounterMTDA, 479
 removeRef, 480
OsclRefCounterMTSA, 481
 OsclRefCounterMTSA, 481
OsclRefCounterMTSA
 ~OsclRefCounterMTSA, 481
 addRef, 482
 getCount, 482
 OsclRefCounterMTSA, 481
 removeRef, 482
OsclRefCounterSA, 483
 OsclRefCounterSA, 483
OsclRefCounterSA
 ~OsclRefCounterSA, 483
 addRef, 484
 getCount, 484
 OsclRefCounterSA, 483
 removeRef, 484
OsclRegistryAccessClient, 485
 OsclRegistryAccessClient, 485
 OsclRegistryClientImpl, 493
 OsclRegistryServTlsImpl, 496
OsclRegistryAccessClient
 ~OsclRegistryAccessClient, 485

Close, 485
 Connect, 485
 GetFactories, 485
 GetFactory, 485
 OsclRegistryAccessClient, 485
 OsclRegistryAccessClientImpl, 487
 OsclRegistryAccessClientTlsImpl, 488
 OsclRegistryAccessElement, 489
 OsclRegistryAccessElement
 iFactory, 489
 iMimeType, 489
 OsclRegistryClient, 490
 OsclRegistryClient, 490
 OsclRegistryClientImpl, 493
 OsclRegistryServTlsImpl, 496
 OsclRegistryClient
 ~OsclRegistryClient, 490
 Close, 490
 Connect, 490
 OsclRegistryClient, 490
 Register, 490
 UnRegister, 491
 OsclRegistryClientImpl, 492
 OsclRegistryClientImpl
 Close, 493
 Connect, 493
 GetFactories, 493
 GetFactory, 493
 OsclRegistryAccessClient, 493
 OsclRegistryClient, 493
 Register, 493
 UnRegister, 493
 OsclRegistryClientTlsImpl, 494
 OsclRegistryServTlsImpl, 495
 OsclRegistryServTlsImpl, 496
 OsclRegistryServTlsImpl
 ~OsclRegistryServTlsImpl, 496
 Close, 496
 Connect, 496
 GetFactories, 496
 GetFactory, 496
 OsclRegistryAccessClient, 496
 OsclRegistryClient, 496
 OsclRegistryServTlsImpl, 496
 Register, 496
 UnRegister, 496
 OsclReturnCode
 osclerror, 89
 OsclScheduler, 497
 OsclErrorTrapImp, 366
 OsclExecScheduler, 378
 OsclExecSchedulerCommonBase, 385
 OsclScheduler
 Cleanup, 497
 Init, 497
 OsclSchedulerCommonBase
 PVActiveBase, 591
 OsclSchedulerObserver, 498
 OsclSchedulerObserver
 ~OsclSchedulerObserver, 498
 OsclSchedulerReadyCallback, 498
 OsclSchedulerTimerCallback, 498
 OsclSchedulerReadyCallback
 OsclSchedulerObserver, 498
 OsclSchedulerTimerCallback
 OsclSchedulerObserver, 498
 OsclScopedLock, 499
 OsclScopedLock, 499
 OsclScopedLock
 ~OsclScopedLock, 499
 OsclScopedLock, 499
 OsclSelect, 500
 OsclSelect, 501
 OsclSelect
 iErrAlloc, 501
 iHeapCheck, 501
 iOscIBase, 501
 iOscIErrorTrap, 501
 iOscILogger, 501
 iOscIMemory, 501
 iOscIScheduler, 501
 iOutputFile, 501
 iSchedulerAlloc, 501
 iSchedulerName, 501
 iSchedulerReserve, 501
 OsclSelect, 501
 OsclSemaphore, 502
 OsclSemaphore, 502
 OsclSemaphore
 ~OsclSemaphore, 502
 Close, 502
 Create, 502
 OsclSemaphore, 502
 Signal, 503
 TryWait, 503
 Wait, 503
 OsclSend
 osclconfig_io.h, 790
 OsclSendMethod, 504
 OsclSendMethod
 ~OsclSendMethod, 504
 GetSendData, 504
 NewL, 504
 Send, 504
 SendRequest, 504
 OsclSendRequest, 505
 OsclSendRequest, 505
 OsclSocketI, 519

OsclSendRequest
 GetSendData, 505
OsclSendRequest, 505
 Send, 505
 Success, 505
OsclSendTo
`osclconfig_io.h`, 790
OsclSendToMethod, 506
OsclSendToMethod
`~OsclSendToMethod`, 506
 GetSendData, 506
 NewL, 506
 SendTo, 506
 SendToRequest, 506
OsclSendToRequest, 507
`OsclSendToRequest`, 507
`OsclSocketI`, 519
OsclSendToRequest
 GetSendData, 507
`OsclSendToRequest`, 507
 SendTo, 507
 Success, 507
OsclSetNonBlocking
`osclconfig_io.h`, 790
OsclSetRecvBufferSize
`osclconfig_io.h`, 790
OsclSharedPtr, 508
`OsclSharedPtr`, 509
OsclSharedPtr
`~OsclSharedPtr`, 509
 get_count, 509
 GetRefCounter, 509
 GetRep, 509
 operator *, 509
 operator TheClass *, 510
 operator->, 510
 operator=, 510
`OsclSharedPtr`, 509
 Unbind, 510
OsclShutdown
`osclconfig_io.h`, 790
OsclShutdownMethod, 511
OsclShutdownMethod
`~OsclShutdownMethod`, 511
 NewL, 511
 Shutdown, 511
 ShutdownRequest, 511
OsclShutdownRequest, 512
`OsclShutdownRequest`, 512
`OsclSocketI`, 519
OsclShutdownRequest
`OsclShutdownRequest`, 512
 Shutdown, 512
OsclSingleton, 513
OsclSingleton, 513
`~OsclSingleton`, 513
`_Ptr`, 514
 operator *, 513
 operator->, 513
`OsclSingleton`, 513
 set, 513
OsclSingletonRegistry, 515
OsclSingletonRegistry
`getInstance`, 515
`lockAndGetInstance`, 515
`OsclBase`, 515
`registerInstance`, 515
`registerInstanceAndUnlock`, 515
OsclSocket
`osclconfig_io.h`, 790
OsclSocketCleanup
`osclconfig_io.h`, 791
OsclSocketI, 516
`OsclSocketRequestAO`, 534
`OsclSocketServI`, 538
OsclSocketI
`~OsclSocketI`, 517
 Accept, 517
 Bind, 517
 Close, 517
 Connect, 517
 Join, 517
 Listen, 517
 Logger, 517
 MakeAddr, 518
 NewL, 518
 Open, 518
`OsclAcceptRequest`, 519
`OsclConnectRequest`, 519
`OsclRecvFromRequest`, 519
`OsclRecvRequest`, 519
`OsclSendRequest`, 519
`OsclSendToRequest`, 519
`OsclShutdownRequest`, 519
`OsclTCPSocket`, 519
`OsclUDPSocket`, 519
 ProcessAccept, 518
 ProcessConnect, 518
 ProcessRecv, 518
 ProcessRecvFrom, 518
 ProcessSend, 518
 ProcessSendTo, 518
 ProcessShutdown, 518
 Recv, 518
 RecvFrom, 518
 RecvFromSuccess, 518
 RecvSuccess, 518

Send, 518
 SendSuccess, 519
 SendTo, 519
 SendToSuccess, 519
 SetRecvBufferSize, 519
 Shutdown, 519
 Socket, 519
OsclSocketIBase, 521
 OsclSocketIBase, 522
OsclSocketIBase
 ~**OsclSocketIBase**, 522
 Accept, 522
 Bind, 522
 BindAsync, 522
 CancelAccept, 523
 CancelBind, 523
 CancelConnect, 523
 CancelFxn, 523
 CancelListen, 523
 CancelRecv, 523
 CancelRecvFrom, 523
 CancelSend, 523
 CancelSendTo, 523
 CancelShutdown, 523
 Close, 523
 Connect, 523
 GetShutdown, 523
 HasAsyncBind, 523
 HasAsyncListen, 523
 iAlloc, 525
 iSocketServ, 525
 IsOpen, 523
 Join, 523
 Listen, 523
 ListenAsync, 523
 Open, 524
 OsclSocketIBase, 522
 OsclSocketMethod, 525
 OsclSocketRequest, 525
 OsclSocketRequestAO, 525
 OsclTCPSocket, 525
 OsclUDPSocket, 525
 Recv, 524
 RecvFrom, 524
 RecvFromSuccess, 524
 RecvSuccess, 524
 Send, 524
 SendSuccess, 524
 SendTo, 524
 SendToSuccess, 524
 Shutdown, 525
OsclSocketMethod, 526
 OsclIPSocketI, 400
 OsclSocketIBase, 525
OsclSocketMethod
 ~**OsclSocketMethod**, 527
 Abort, 527
 AbortAll, 527
 Alloc, 527
 CancelMethod, 527
 ConstructL, 527
 iContainer, 528
 iSocketFxn, 528
 iSocketRequestAO, 528
 MethodDone, 527
 OsclSocketMethod, 527
 Run, 527
 StartMethod, 528
OsclSocketObserver, 529
OsclSocketObserver
 ~**OsclSocketObserver**, 529
 HandleSocketEvent, 529
OsclSocketRequest, 530
 OsclSocketIBase, 525
 OsclSocketRequest, 530
 OsclSocketRequestAO, 534
 OsclSocketServI, 538
OsclSocketRequest
 Activate, 530
 CancelRequest, 530
 Complete, 530
 Fxn, 530
 iParam, 530
 iSocketI, 530
 iSocketRequestAO, 530
 OsclSocketRequest, 530
OsclSocketRequestAO, 531
 OsclIPSocketI, 400
 OsclSocketIBase, 525
 OsclSocketRequestAO, 532
OsclSocketRequestAO
 ~**OsclSocketRequestAO**, 532
 Abort, 532
 Alloc, 532
 CleanupParam, 532
 ConstructL, 532
 DoCancel, 532
 GetSocketError, 532
 iContainer, 534
 Id, 533
 iParam, 534
 iParamSize, 534
 iSocketError, 534
 NewRequest, 533
 OsclSocketI, 534
 OsclSocketMethod, 534

OsclSocketRequest, 534
 OsclSocketRequestAO, 532
 RequestDone, 533
 Run, 533
 SocketI, 533
 SocketObserver, 533
 Success, 533
OsclSocketSelect
 osclconfig_io.h, 791
OsclSocketServ, 535
 OsclSocketServI, 538
OsclSocketServ
 ~OsclSocketServ, 535
 Close, 535
 Connect, 535
 NewL, 536
 OsclDNS, 536
 OsclTCPSocket, 536
 OsclUDPSocket, 536
OsclSocketServI, 537
 OsclSocketServRequestList, 541
OsclSocketServI
 Close, 537
 Connect, 537
 IsServerThread, 538
 LoopbackSocket, 538
 NewL, 538
 OsclDNSI, 538
 OsclSocketI, 538
 OsclSocketRequest, 538
 OsclSocketServ, 538
 OsclSocketServRequestList, 538
 OsclTCPSocketI, 538
 OsclUDPSocketI, 538
OsclSocketServIBase, 539
 ESocketServ_Connected, 539
 ESocketServ_Error, 540
 ESocketServ_Idle, 539
 OsclSocketServIBase, 540
OsclSocketServIBase
 ~OsclSocketServIBase, 540
 Close, 540
 Connect, 540
 iAlloc, 540
 iLogger, 540
 iServError, 540
 iServState, 540
 IsServConnected, 540
 OsclSocketServIBase, 540
 State, 540
 TSocketServState, 539
OsclSocketServRequestList, 541
 OsclSocketServI, 538
 OsclSocketServRequestList, 541

 OsclSocketServRequestList
 Add, 541
 Close, 541
 Open, 541
 OsclSocketServI, 541
 OsclSocketServRequestList, 541
 Remove, 541
 StartCancel, 541
 WaitOnRequests, 541
 Wakeup, 541
OsclSocketServRequestQElem, 543
OsclSocketServRequestQElem, 543
OsclSocketServRequestQElem
 iCancel, 543
 iSelect, 543
 iSocketRequest, 543
 OsclSocketServRequestQElem, 543
OsclSocketStartup
 osclconfig_io.h, 791
OsclSuccess
 osclerror, 88
OsclTagTreeType
 osclmemory, 56
OsclTCPSocket, 544
 OsclSocketI, 519
 OsclSocketIBase, 525
 OsclSocketServ, 536
OsclTCPSocket
 ~OsclTCPSocket, 545
 Accept, 545
 Bind, 545
 BindAsync, 545
 CancelAccept, 545
 CancelBind, 546
 CancelConnect, 546
 CancelListen, 546
 CancelRecv, 546
 CancelSend, 546
 CancelShutdown, 546
 Close, 546
 Connect, 547
 GetAcceptedSocketL, 547
 GetRecvData, 547
 GetSendData, 547
 Listen, 548
 ListenAsync, 548
 NewL, 548
 Recv, 548
 Send, 549
 Shutdown, 549
OsclTCPSocketI, 550
 OsclSocketServI, 538
OsclTCPSocketI
 ~OsclTCPSocketI, 551

Accept, 551
 BindAsync, 551
 CancelAccept, 551
 CancelBind, 551
 CancelConnect, 551
 CancelListen, 551
 CancelRecv, 551
 CancelSend, 551
 CancelShutdown, 551
 Close, 551
 Connect, 551
 GetAcceptedSocketL, 551
 GetRecvData, 551
 GetSendData, 551
 Listen, 551
 ListenAsync, 552
 NewL, 552
 Recv, 552
 Send, 552
 Shutdown, 552
OscIThread, 553
 OscIThread, 553
OscIThread
 ~OscIThread, 553
 CompareId, 553
 Create, 554
 EnableKill, 554
 Exit, 554
 GetId, 554
 GetPriority, 555
 OscIThread, 553
 Resume, 555
 SetPriority, 555
 SleepMillsec, 555
 Suspend, 555
 Terminate, 556
OscIThread_State
 oscI_thread.h, 763
OscIThreadLock, 557
 OscIThreadLock, 557
OscIThreadLock
 ~OscIThreadLock, 557
 Lock, 557
 OscIThreadLock, 557
 Unlock, 557
OscIThreadPriority
 oscI_thread.h, 763
OscITickCount, 558
OscITickCount
 MsecToTicks, 558
 TickCount, 558
 TickCountFrequency, 558
 TickCountPeriod, 558
 TicksToMsec, 558

OSCLTICKCOUNT_MAX_TICKS
 osclutil, 66
OscITimer, 560
 OscITimer, 561
OscITimer
 ~OscITimer, 561
 callback_timer_type, 561
 CallbackTimer< Alloc >, 562
 Cancel, 561
 Clear, 561
 OscITimer, 561
 Request, 561
 SetExactFrequency, 561
 SetFrequency, 562
 SetObserver, 562
 TimerBaseElapsed, 562
OscITimerCompare, 563
 OscIExecSchedulerCommonBase, 385
OscITimerCompare
 compare, 563
OscITimerObject, 564
 OscIExecSchedulerCommonBase, 387
 OscITimerObject, 565
 PVActiveBase, 591
 PVActiveStats, 592
 PVThreadContext, 611
OscITimerObject
 ~OscITimerObject, 565
 AddToScheduler, 565
 After, 565
 Cancel, 565
 DoCancel, 565
 IsBusy, 566
 OscITimerObject, 565
 Priority, 566
 RemoveFromScheduler, 566
 RunError, 566
 RunIfNotReady, 566
 SetBusy, 566
 SetStatus, 566
 Status, 567
 StatusRef, 567
OscITimerObserver, 568
OscITimerObserver
 ~OscITimerObserver, 568
 TimeoutOccurred, 568
OscITimerQ, 569
OscITimerQ
 Add, 569
 Construct, 569
 IsIn, 569
 Pop, 569
 PopTop, 569
 Remove, 569

Top, [569](#)
OsclTLS, [570](#)
 OsclTLS, [570](#)
OsclTLS
 ~OsclTLS, [570](#)
 _Ptr, [571](#)
 operator *, [570](#)
 operator->, [570](#)
 OsclTLS, [570](#)
 set, [570](#)
OsclTLSEEx, [572](#)
 OsclTLSEEx, [572](#)
OsclTLSEEx
 ~OsclTLSEEx, [572](#)
 _Ptr, [573](#)
 operator *, [572](#)
 operator->, [572](#)
 OsclTLSEEx, [572](#)
 set, [572](#)
OsclTLSRegistry, [574](#)
OsclTLSRegistry
 getInstance, [574](#)
 OsclBase, [574](#)
 registerInstance, [574](#)
OsclTLSRegistryEx, [575](#)
OsclTLSRegistryEx
 getInstance, [575](#)
 registerInstance, [575](#)
OsclTrapItem, [576](#)
 OsclTrapItem, [576](#)
OsclTrapItem
 OsclTrapItem, [576](#)
 OsclTrapStack, [576](#)
 OsclTrapStackItem, [576](#)
OsclTrapOperation
 osclerror, [89](#)
OsclTrapStack, [577](#)
 OsclErrorTrapImp, [366](#)
 OsclTrapItem, [576](#)
OsclTrapStack
 OsclError, [577](#)
 OsclErrorTrap, [577](#)
 OsclErrorTrapImp, [577](#)
OsclTrapStackItem, [578](#)
 OsclTrapItem, [576](#)
 OsclTrapStackItem, [578](#)
OsclTrapStackItem
 iCBase, [578](#)
 iNext, [578](#)
 iTAny, [578](#)
 iTrapOperation, [578](#)
 OsclTrapStackItem, [578](#)
OsclUDPSocket, [579](#)
OsclSocketI, [519](#)
 OsclSocketIBase, [525](#)
 OsclSocketServ, [536](#)
OsclUDPSocket
 ~OsclUDPSocket, [579](#)
 Bind, [580](#)
 BindAsync, [580](#)
 CancelBind, [580](#)
 CancelRecvFrom, [580](#)
 CancelSendTo, [580](#)
 Close, [580](#)
 GetRecvData, [581](#)
 GetSendData, [581](#)
 Join, [581](#)
 NewL, [581](#)
 RecvFrom, [582](#)
 SendTo, [582](#)
 SetRecvBufferSize, [582](#)
OsclUDPSocketI, [584](#)
 OsclSocketServI, [538](#)
OsclUDPSocketI
 ~OsclUDPSocketI, [585](#)
 BindAsync, [585](#)
 CancelBind, [585](#)
 CancelRecvFrom, [585](#)
 CancelSendTo, [585](#)
 Close, [585](#)
 GetRecvData, [585](#)
 GetSendData, [585](#)
 NewL, [585](#)
 RecvFrom, [585](#)
 SendTo, [585](#)
OsclUid32
 oscl_uuid.h, [774](#)
OsclUnMakeSockAddr
 osclconfig_io.h, [791](#)
osclutil
 ~OSCL_HeapString, [80](#)
 ~OSCL_StackString, [80](#)
 ~OSCL_wHeapString, [80](#)
 ~OSCL_wStackString, [80](#)
 APPEND_MEDIA_AT_END, [80](#)
 BufferFreeFuncPtr, [66](#)
 extract_string, [66](#)
 get_cstr, [66, 67](#)
 get_maxsize, [67](#)
 get_size, [67, 68](#)
 get_str, [68](#)
 GetBufferState, [68](#)
 GetFragment, [69](#)
 MediaTimestamp, [66](#)
 operator=, [69, 70](#)
 oscl_abs, [70](#)
 OSCL_ASCII_CASE_MAGIC_BIT, [80](#)
 oscl_asin, [70](#)

oscl_atan, 70
 oscl_cos, 70
 oscl_exp, 71
 oscl_floor, 71
 OSCL_HeapString, 71
 oscl_isdigit, 66
 oscl_log, 72
 oscl_log10, 72
 oscl_pow, 72
 oscl_sin, 72
 oscl_snprintf, 72
 oscl_sqrt, 72
 OSCL_StackString, 72, 73
 oscl_str_escape_xml, 73
 oscl_str_is_valid_utf8, 74
 oscl_str_need_escape_xml, 74
 oscl_str_truncate_utf8, 74
 oscl_str_unescape_uri, 75
 oscl_tan, 76
 OSCL_TStrPtrLen, 66
 oscl_UncodeToUTF8, 76
 oscl_UTF8ToUnicode, 76
 oscl_vsnprintf, 77, 79
 OSCL_wHeapString, 79
 OSCL_wStackString, 79
 OsclComponentFactory, 66
 OSCLTICKCOUNT_MAX_TICKS, 66
 PV_atof, 79
 PV_atoi, 79
 set, 79, 80
 skip_to_line_term, 80
 skip_to_whitespace, 80
 skip_whitespace, 80
 skip_whitespace_and_line_term, 80
 StrCSumPtrLen, 66
 StrPtrLen, 66
 WStrPtrLen, 66
 OsclUuid, 586
 OsclUuid, 587
 OsclUuid
 data1, 587
 data2, 587
 data3, 587
 data4, 587
 operator!=, 587
 operator=, 587
 operator==, 587
 OsclUuid, 587
 OsclValidInetAddr
 osclconfig_io.h, 791
 other
 Oscl_TAlloc::rebind, 272
 OTHER_ERROR
 OsclProcStatus, 456

OUTOFMEMORY_ERROR
 OsclProcStatus, 456

pad
 MM_AllocBlockFence, 142
 MM_AllocBlockHdr, 143

pair_citerator_citerator
 Oscl_Map, 207

pair_iterator_bool
 Oscl_Map, 207
 Oscl_TagTree, 258

pair_iterator_iterator
 Oscl_Map, 207

pAllocInfo
 MM_AllocNode, 146

parent
 Oscl_Rb_Tree_Node_Base, 243
 Oscl_TagTree::Node, 268

pAudit
 OsclAuditCB, 309

pBasePosition
 OsclBinStream, 327

peakNumAllocs
 MM_Stats_t, 160

peakNumBytes
 MM_Stats_t, 160

PendComplete
 OsclActiveObject, 300
 OsclExecSchedulerCommonBase, 384
 OsclReadyQ, 466

PendForExec
 OsclActiveObject, 300

per_allocation_overhead
 MM_AuditOverheadStats, 156

perms
 oscl_stat_buf, 247

pFileName
 MM_AllocInfo, 145

pMemBlock
 MM_AllocInfo, 145
 MM_AllocQueryInfo, 147

pMMFIParam
 OsclMemStatsNode, 439

pMMStats
 OsclMemStatsNode, 439

pNext
 MM_AllocNode, 146

pNode
 MM_AllocBlockHdr, 143

pointer
 MemAllocator, 141
 Oscl_Map, 207
 Oscl_Queue, 225
 Oscl_Rb_Tree, 232

Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
 Oscl_TAlloc, 270
 Oscl_Vector, 274
Pop
 OsclError, 360
 OsclTimerQ, 569
pop
 Oscl_Queue, 226
 Oscl_Queue_Base, 228
 OsclPriorityQueue, 453
pop_back
 Oscl_Vector, 276
 Oscl_Vector_Base, 280
pop_heap
 OsclPriorityQueue, 453
 OsclPriorityQueueBase, 455
PopDealloc
 OsclError, 360, 361
PopTop
 OsclReadyQ, 466
 OsclTimerQ, 569
port
 OsclNetworkAddress, 447
PositionInBlock
 OsclBinStream, 326
pPosition
 OsclBinStream, 327
pPrev
 MM_AllocNode, 146
Priority
 OsclActiveObject, 300
 OsclTimerObject, 566
ProcessAccept
 OsclSocketI, 518
ProcessConnect
 OsclSocketI, 518
ProcessRecv
 OsclSocketI, 518
ProcessRecvFrom
 OsclSocketI, 518
ProcessSend
 OsclSocketI, 518
ProcessSendTo
 OsclSocketI, 518
ProcessShutdown
 OsclSocketI, 518
pRootNode
 MM_AllocBlockHdr, 143
pruneSubtree
 MM_Audit_Imp, 154
PSHARED_ATTRIBUTE_SETTING_ERROR
OsclProcStatus, 457
PSHARED_NOT_ZERO_ERROR
 OsclProcStatus, 457
pStats
 MM_Stats_CB, 158
pStatsNode
 MM_AllocInfo, 145
 OsclAuditCB, 309
Ptr
 OsclPtr, 458
 OsclPtrC, 461
ptr
 OsclMemoryFragment, 422
 StrPtrLen, 624
 WStrPtrLen, 634
push
 Oscl_Queue, 226
 Oscl_Queue_Base, 228
 OsclPriorityQueue, 453
push_back
 Oscl_Vector, 277
 Oscl_Vector_Base, 280
push_front
 Oscl_Vector, 277
 Oscl_Vector_Base, 280
push_heap
 OsclPriorityQueue, 453
 OsclPriorityQueueBase, 455
PushL
 OsclError, 361
PV8601TIME_BUFFER_SIZE
 osclbase, 43
PV8601timeStrBuf
 osclbase, 32
PV8601ToRFC822
 osclbase, 41
PV_atof
 osclutil, 79
PV_atoi
 osclutil, 79
PV_CHAR_CLOSE_BRACKET
 oscl_uuid.h, 774
PV_CHAR_COMMA
 oscl_uuid.h, 774
PV_DNS_IS_THREAD
 oscl_dns_tuneables.h, 652
PV_DNS_SERVER
 oscl_dns_tuneables.h, 652
PV_DYNAMIC_LOADING_CONFIG_FILE_PATH
 osclconfig_lib.h, 794
PV_OSCL_SOCKET_1MB_RECV_BUF
 oscl_socket_tuneables.h, 748

PV_OSCL_SOCKET_SERVER_LOGGER_-OUTPUT
oscl_socket_tuneables.h, 748
PV_OSCL_SOCKET_STATS_LOGGING
oscl_socket_tuneables.h, 748
PV_RUNTIME_LIB_FILENAME_-EXTENSION
osclconfig_lib.h, 794
PV_SCED_CHECK_Q
osclproc, 101
PV_SCED_ENABLE_AO_STATS
osclproc, 101
PV_SCED_ENABLE_LOOP_STATS
osclproc, 101
PV_SCED_ENABLE_PERF_LOGGING
osclproc, 101
PV_SCED_ENABLE_THREAD_-CONTEXT_CHECKS
osclproc, 101
PV_SCED_FAIR_SCHEDULING
osclproc, 101
PV_SCED_LOG_Q
osclproc, 101
PV_SOCKET_REQUEST_AO_PRIORITY
oscl_socket_tuneables.h, 748
PV_SOCKET_SERVER
oscl_socket_tuneables.h, 748
PV_SOCKET_SERVER_AO_INTERVAL_-MSEC
oscl_socket_tuneables.h, 749
PV_SOCKET_SERVER_AO_PRIORITY
oscl_socket_tuneables.h, 749
PV_SOCKET_SERVER_IS_THREAD
oscl_socket_tuneables.h, 749
PV_SOCKET_SERVER_SELECT
oscl_socket_tuneables.h, 749
PV_SOCKET_SERVER_SELECT_-LOOPBACK_SOCKET
oscl_socket_tuneables.h, 749
PV_SOCKET_SERVER_SELECT_-TIMEOUT_MSEC
oscl_socket_tuneables.h, 749
PV_SOCKET_SERVER_THREAD_-PRIORITY
oscl_socket_tuneables.h, 749
PV_SOCKET_SERVI_STATS
oscl_socket_tuneables.h, 749
PVActiveBase, 588
OsclExecSchedulerBase, 379
OsclExecSchedulerCommonBase, 387
PVActiveBase, 589
PVActiveStats, 592
PVThreadContext, 611
PVActiveBase
~PVActiveBase, 589
Activate, 589
AddToScheduler, 589
Cancel, 589
Destroy, 589
DoCancel, 589
iAddedNum, 591
iBusy, 591
iName, 591
iPVActiveStats, 591
iPVReadyQLink, 591
IsAdded, 589
IsInAnyQ, 590
iStatus, 591
iThreadContext, 591
OsclActiveObject, 591
OsclExecScheduler, 591
OsclReadyCompare, 591
OsclReadyQ, 591
OsclReadySetPosition, 591
OsclSchedulerCommonBase, 591
OsclTimerObject, 591
PVActiveBase, 589
PVActiveStats, 591
RemoveFromScheduler, 590
Run, 590
RunError, 590
PVActiveStats, 592
OsclExecSchedulerCommonBase, 387
PVActiveBase, 591
PVActiveStats
OsclActiveObject, 592
OsclExecScheduler, 592
OsclExecSchedulerCommonBase, 592
OsclReadyQ, 592
OsclTimerObject, 592
PVActiveBase, 592
PVCleanupStack
_OsclHeapBase, 107
PVError_DoLeave
oscl_error_imp_fatalerror.h, 660
oscl_error_imp_jumps.h, 662
osclerror, 88
PVERROR_IMP_JUMPS
osclerror, 88
PVERRORTRAP_REGISTRY
osclerror, 88
PVERRORTRAP_REGISTRY_ID
osclerror, 89
PVEXECNAMELEN
osclproc, 101
PVLogger, 593
~PVLogger, 594
AddAppender, 594

AddFilter, 594
 alloc_type, 594
 Cleanup, 595
 DisableAppenderInheritance, 595
 filter_status_type, 594
 GetLoggerObject, 595
 GetLogLevel, 595
 GetNumAppenders, 595
 GetParent, 596
 Init, 596
 IsActive, 596
 log_level_type, 594
 LogMsgBuffers, 596
 LogMsgBuffersV, 596
 LogMsgString, 597
 LogMsgStringV, 597
 message_id_type, 594
 PVLogger, 594
 PVLoggerRegistry, 598
 RemoveAppender, 597
 SetLogLevel, 598
 SetLogLevelAndPropagate, 598
 SetParent, 598
pvlogger.h, 819

- _PVLOGGER_LOGBIN, 821
- _PVLOGGER_LOGBIN_V, 821
- _PVLOGGER_LOGMSG, 821
- _PVLOGGER_LOGMSG_V, 821
- PVLOGGER_ENABLE, 821
- PVLOGGER_INST_LEVEL, 822
- PVLOGGER_INST_LEVEL_SUPPORT, 822
- PVLOGGER_LEVEL_UNINITIALIZED, 825
- PVLOGGER_LOG_USE_ONLY, 822
- PVLOGGER_LOGBIN, 822
- PVLOGGER_LOGBIN_PVLOGMSG_INST_HLDBG, 822
- PVLOGGER_LOGBIN_PVLOGMSG_INST_LLDBG, 823
- PVLOGGER_LOGBIN_PVLOGMSG_INST_MLDBG, 823
- PVLOGGER_LOGBIN_PVLOGMSG_INST_PROF, 823
- PVLOGGER_LOGBIN_PVLOGMSG_INST_REL, 823
- PVLOGGER_LOGBIN_V, 823
- PVLOGGER_LOGBIN_V_-
PVLOGMSG_INST_HLDBG, 823
- PVLOGGER_LOGBIN_V_-
PVLOGMSG_INST_LLDBG, 823
- PVLOGGER_LOGBIN_V_-
PVLOGMSG_INST_MLDBG, 823
- PVLOGGER_LOGBIN_V_-
PVLOGMSG_INST_PROF, 823

pvlogger_accessories.h, 827

- PVLOGGER_FILTER_ACCEPT, 827
- PVLOGGER_FILTER_NEUTRAL, 827
- PVLOGGER_FILTER_REJECT, 827

pvlogger_c.h, 828

- PVLOGGER_C_INST_LEVEL, 829
- pvLogger_GetLoggerObject, 829
- pvLogger_IsActive, 829

pvLogger_LogMsgString, 829
 PVLOGMSG_C_ALERT, 829
 PVLOGMSG_C_CRIT, 829
 PVLOGMSG_C_EMERG, 829
 PVLOGMSG_C_ERR, 829
 PVLOGMSG_C_INFO, 829
 PVLOGMSG_C_INST_HLDBG, 829
 PVLOGMSG_C_INST_LLDBG, 829
 PVLOGMSG_C_INST_MLDBG, 829
 PVLOGMSG_C_INST_PROF, 829
 PVLOGMSG_C_INST_REL, 829
 PVLOGMSG_C_NOTICE, 829
 PVLOGMSG_C_STACK_DEBUG, 829
 PVLOGMSG_C_STACK_TRACE, 829
 PVLOGMSG_C_WARNING, 829
PVLOGGER_C_INST_LEVEL
 pvlogger_c.h, 829
PVLOGGER_ENABLE
 pvlogger.h, 821
PVLOGGER_FILTER_ACCEPT
 pvlogger_accessories.h, 827
PVLOGGER_FILTER_NEUTRAL
 pvlogger_accessories.h, 827
PVLOGGER_FILTER_REJECT
 pvlogger_accessories.h, 827
pvLogger_GetLoggerObject
 pvlogger_c.h, 829
PVLOGGER_INST_LEVEL
 pvlogger.h, 822
PVLOGGER_INST_LEVEL_SUPPORT
 pvlogger.h, 822
pvLogger_IsActive
 pvlogger_c.h, 829
PVLOGGER_LEVEL_UNINITIALIZED
 pvlogger.h, 825
PVLOGGER_LOG_USE_ONLY
 pvlogger.h, 822
PVLOGGER_LOGBIN
 pvlogger.h, 822
PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 HLDBG
 pvlogger.h, 822
PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 LLDBG
 pvlogger.h, 823
PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 MLDBG
 pvlogger.h, 823
PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 PROF
 pvlogger.h, 823
PVLOGGER_LOGBIN_PVLOGMSG_INST_-
 REL
 pvlogger.h, 823
 PVLOGMSG_C_ALERT, 829
 PVLOGMSG_C_CRIT, 829
 PVLOGMSG_C_EMERG, 829
 PVLOGMSG_C_ERR, 829
 PVLOGMSG_C_INFO, 829
 PVLOGMSG_C_INST_HLDBG, 829
 PVLOGMSG_C_INST_LLDBG, 829
 PVLOGMSG_C_INST_MLDBG, 829
 PVLOGMSG_C_INST_PROF, 829
 PVLOGMSG_C_INST_REL, 829
 PVLOGMSG_C_NOTICE, 829
 PVLOGMSG_C_STACK_DEBUG, 829
 PVLOGMSG_C_STACK_TRACE, 829
 PVLOGMSG_C_WARNING, 829
PVLOGGER_LOGBIN_V
 pvlogger.h, 823
PVLOGGER_LOGBIN_V_PVLOGMSG_-
 INST_HLDBG
 pvlogger.h, 823
PVLOGGER_LOGBIN_V_PVLOGMSG_-
 INST_LLDBG
 pvlogger.h, 823
PVLOGGER_LOGBIN_V_PVLOGMSG_-
 INST_PROF
 pvlogger.h, 823
PVLOGGER_LOGBIN_V_PVLOGMSG_-
 INST_REL
 pvlogger.h, 823
PVLOGGER_LOGBIN_V_PVLOGMSG_V_-
 INST_MLDBG
 pvlogger.h, 823
PVLOGGER_LOGMSG
 pvlogger.h, 823
PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_HLDBG
 pvlogger.h, 823
PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_LLDBG
 pvlogger.h, 824
PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_MLDBG
 pvlogger.h, 824
PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_PROF
 pvlogger.h, 824
PVLOGGER_LOGMSG_PVLOGMSG_-
 INST_REL
 pvlogger.h, 824
PVLOGGER_LOGMSG_V
 pvlogger.h, 824
PVLOGGER_LOGMSG_V_PVLOGMSG_-
 INST_HLDBG
 pvlogger.h, 824
PVLOGGER_LOGMSG_V_PVLOGMSG_-
 INST_LLDBG
 pvlogger.h, 824
PVLOGGER_LOGMSG_V_PVLOGMSG_-
 INST_MLDBG
 pvlogger.h, 824
PVLOGGER_LOGMSG_V_PVLOGMSG_-
 INST_PROF
 pvlogger.h, 824
PVLOGGER_LOGMSG_V_PVLOGMSG_-
 INST_REL
 pvlogger.h, 824
pvLogger_LogMsgString
 pvlogger_c.h, 829
pvlogger_registry.h, 830

PVLoggerAppender, 599
 PVLoggerAppender
 ~PVLoggerAppender, 599
 AppendBuffers, 599
 AppendString, 599
 message_id_type, 599
 PVLoggerFilter, 600
 PVLoggerFilter
 ~PVLoggerFilter, 601
 filter_status_type, 600
 FilterOpaqueMessge, 601
 FilterString, 601
 log_level_type, 600
 message_id_type, 600
 PVLoggerLayout, 602
 PVLoggerLayout
 ~PVLoggerLayout, 602
 FormatOpaqueMessage, 602
 FormatString, 602
 message_id_type, 602
 PVLoggerRegistry, 604
 PVLogger, 598
 PVLoggerRegistry, 604
 PVLoggerRegistry
 ~PVLoggerRegistry, 604
 alloc_type, 604
 CreatePVLogger, 605
 GetPVLoggerObject, 605
 GetPVLoggerRegistry, 605
 log_level_type, 604
 PVLoggerRegistry, 604
 SetNodeLogLevelExplicit, 605
 PVLOGMSG_ALERT
 pvlogger.h, 825
 PVLOGMSG_C_ALERT
 pvlogger_c.h, 829
 PVLOGMSG_C_CRIT
 pvlogger_c.h, 829
 PVLOGMSG_C_EMERG
 pvlogger_c.h, 829
 PVLOGMSG_C_ERR
 pvlogger_c.h, 829
 PVLOGMSG_C_INFO
 pvlogger_c.h, 829
 PVLOGMSG_C_INST_HLDBG
 pvlogger_c.h, 829
 PVLOGMSG_C_INST_LLDBG
 pvlogger_c.h, 829
 PVLOGMSG_C_INST_MLDBG
 pvlogger_c.h, 829
 PVLOGMSG_C_INST_PROF
 pvlogger_c.h, 829
 PVLOGMSG_C_INST_REL
 pvlogger_c.h, 829
 PVLOGMSG_NONFATAL_ERROR
 pvlogger.h, 826
 PVLOGMSG_NOTICE
 pvlogger.h, 826
 PVLOGMSG_STACK_TRACE
 pvlogger.h, 826
 PVLOGMSG_STATISTIC
 pvlogger.h, 826
 PVLOGMSG_VERBOSE
 pvlogger.h, 826
 PVLOGMSG_WARNING
 pvlogger.h, 826
 PVMEM_INST_LEVEL
 osclbase, 32
 osclconfig_memory.h, 797
 PVNETWORKADDRESS_LEN
 oscl_socket_types.h, 750
 PVosclBase_Cleanup
 osclbase, 42
 PVosclBase_Init
 osclbase, 42
 PVSCHEDNAMELEN
 osclproc, 101
 PVSchedulerStopper, 607

OsclExecSchedulerCommonBase, 387
 PVSchedulerStopper, 607
PVSchedulerStopper
 ~PVSchedulerStopper, 607
 PVSchedulerStopper, 607
PVSOCK_ERR_BAD_PARAM
 oscl_socket_imp_pv.h, 733
PVSOCK_ERR_NOT_IMPLEMENTED
 oscl_socket_imp_pv.h, 733
PVSOCK_ERR_SERV_NOT_CONNECTED
 oscl_socket_imp_pv.h, 733
PVSOCK_ERR SOCK_NO_SERV
 oscl_socket_imp_pv.h, 733
PVSOCK_ERR SOCK_NOT_CONNECTED
 oscl_socket_imp_pv.h, 733
PVSOCK_ERR SOCK_NOT_OPEN
 oscl_socket_imp_pv.h, 733
PVSockBufRecv, 608
 PVSockBufRecv, 608
PVSockBufRecv
 iLen, 608
 iMaxLen, 608
 iPtr, 608
 PVSockBufRecv, 608
PVSockBufSend, 609
 PVSockBufSend, 609
PVSockBufSend
 iLen, 609
 iPtr, 609
 PVSockBufSend, 609
PVThreadContext, 610
 OsclExecSchedulerCommonBase, 387
PVThreadContext, 610
PVThreadContext
~PVThreadContext, 610
EnterThreadContext, 610
ExitThreadContext, 610
Id, 610
IsSameThreadContext, 610
OsclActiveObject, 611
OsclCoeActiveScheduler, 611
OsclCoeActiveSchedulerBase, 611
OsclExecScheduler, 611
OsclExecSchedulerBase, 611
OsclExecSchedulerCommonBase, 611
OsclTimerObject, 611
PVActiveBase, 611
PVThreadContext, 610
ThreadHasScheduler, 611

QUE_ITER_BEGIN
osclproc, 101
QUE_ITER_END
osclproc, 101

Rand
OsclRand, 462
Read
Oscl_File, 177
OsclAsyncFile, 306
OsclBinIStreamBigEndian, 315
OsclFileCache, 390
OsclNativeFile, 444
read
OSCL_String, 251
OSCL_wString, 294
Read_uint16
OsclBinIStreamBigEndian, 315
OsclBinIStreamLittleEndian, 318
Read_uint32
OsclBinIStreamBigEndian, 315
OsclBinIStreamLittleEndian, 318
Read_uint8
OsclBinIStream, 312
ReadAsync
OsclNativeFile, 444
ReadAsyncCancel
OsclNativeFile, 444
rebalance
Oscl_Rb_Tree_Base, 234
rebalance_for_erase
Oscl_Rb_Tree_Base, 234
Recv
OsclRecvMethod, 471
OsclRecvRequest, 472
OsclSocketI, 518
OsclSocketIBase, 524
OsclTCPSocket, 548
OsclTCPSocketI, 552
RecvFrom
OsclRecvFromMethod, 467
OsclRecvFromRequest, 469
OsclSocketI, 518
OsclSocketIBase, 524
OsclUDPSocket, 582
OsclUDPSocketI, 585
RecvFromParam, 612
RecvFromParam, 612
RecvFromParam
iAddr, 612
iBufRecv, 612
iFlags, 612
iMultiMaxLen, 612
iPacketLen, 612
iPacketSource, 612
RecvFromParam, 612
RecvFromRequest
OsclRecvFromMethod, 467
RecvFromSuccess

OsclSocketI, 518
 OsclSocketIBase, 524
RecvParam, 614
 RecvParam, 614
RecvParam
 iBufRecv, 614
 iFlags, 614
 RecvParam, 614
RecvRequest
 OsclRecvMethod, 471
RecvSuccess
 OsclSocketI, 518
 OsclSocketIBase, 524
red
 Oscl_Rb_Tree_Node_Base, 242
RedBl
 Oscl_Rb_Tree_Node_Base, 242
refcount
 CHeapRep, 126
reference
 Oscl_Map, 207
 Oscl_Queue, 225
 Oscl_Rb_Tree, 232
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
 Oscl_TAlloc, 270
 Oscl_Vector, 274
Register
 OsclComponentRegistry, 332
 OsclRegistryClient, 490
 OsclRegistryClientImpl, 493
 OsclRegistryServTlsImpl, 496
RegisterForCallback
 OsclExecScheduler, 377
 OsclReadyQ, 466
registerInstance
 OsclSingletonRegistry, 515
 OsclTLSRegistry, 574
 OsclTLSRegistryEx, 575
registerInstanceAndUnlock
 OsclSingletonRegistry, 515
release
 OsclExclusiveArrayPtr, 370
 OsclExclusivePtr, 373
 OsclExclusivePtrA, 376
 OSCLMemAutoPtr, 417
RELOCK_MUTEX_ERROR
 OsclProcStatus, 457
Remove
 OsclDoubleLink, 354
 OsclReadyQ, 466
 OsclSocketServRequestList, 541
 OsclTimerQ, 569
remove
 OsclPriorityQueue, 453
 OsclPriorityQueueBase, 455
remove_element
 Oscl_Linked_List, 200
 Oscl_Linked_List_Base, 204
 Oscl_MTLLinked_List, 216
remove_ref
 CHeapRep, 126
removeALLAllocNodes
 MM_Audit_Imp, 154
removeAllocNode
 MM_Audit_Imp, 154
RemoveAppender
 PVLogger, 597
RemoveFromScheduler
 OsclActiveObject, 300
 OsclTimerObject, 566
 PVActiveBase, 590
RemoveRef
 DNSRequestParam, 130
removeRef
 Oscl_DefAllocWithRefCounter, 169
 OsclMemPoolFixedChunkAllocator, 426
 OsclMemPoolResizableAllocator, 433
 OsclRefCounter, 473
 OsclRefCounterDA, 476
 OsclRefCounterMTDA, 480
 OsclRefCounterMTSA, 482
 OsclRefCounterSA, 484
Request
 OsclTimer, 561
RequestCanceled
 OsclExecSchedulerCommonBase, 384
RequestDone
 OsclDNSRequestAO, 352
 OsclSocketRequestAO, 533
reserve
 Oscl_Queue_Base, 228
 Oscl_Vector_Base, 281
 OsclPriorityQueue, 453
ReserveSpace
 OsclBinStream, 326
Reset
 OsclDoubleListBase, 357
reset
 BufferState, 115
 MM_FailInsertParam, 157
 MM_Stats_t, 160
 OsclMemStatsNode, 439
ResetLogPerf
 OsclExecSchedulerCommonBase, 384
Resume

OsclThread, 555
 ResumeScheduler
 OsclExecSchedulerCommonBase, 384
 retrieveParentTag
 MM_Audit_Imp, 154
 retrieveParentTagLength
 MM_Audit_Imp, 154
 RFC822ToPV8601
 osclbase, 42
 Right
 OsclPtrC, 461
 right
 Oscl_Rb_Tree_Node_Base, 243
 rotate_left
 Oscl_Rb_Tree_Base, 234
 rotate_right
 Oscl_Rb_Tree_Base, 234
 Run
 CallbackTimer, 120
 OsclDNSMethod, 347
 OsclDNSRequestAO, 352
 OsclSocketMethod, 527
 OsclSocketRequestAO, 533
 PVActiveBase, 590
 RunError
 OsclActiveObject, 300
 OsclTimerObject, 566
 PVActiveBase, 590
 RunIfNotReady
 OsclActiveObject, 301
 OsclTimerObject, 566
 RunSchedulerNonBlocking
 OsclExecScheduler, 377

 save_registry
 TLSStorageOps, 631
 second
 Oscl_Pair, 223
 SECONDS
 osclbase, 33
 Seed
 OsclRand, 462
 Seek
 Oscl_File, 178
 OsclAsyncFile, 306
 OsclBinStream, 326
 OsclFileCache, 390
 OsclNativeFile, 445
 seek_type
 Oscl_File, 175
 SEEKCUR
 Oscl_File, 175
 SEEKEND
 Oscl_File, 175

 seekFromCurrentPosition
 OsclBinStream, 326
 SEEKSET
 Oscl_File, 175
 self
 Oscl_Map, 207
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 Oscl_TagTree::const_iterator, 262
 Oscl_TagTree::iterator, 265
 SEM_NOT_SIGNALLED_ERROR
 OsclProcStatus, 457
 Send
 OsclSendMethod, 504
 OsclSendRequest, 505
 OsclSocketI, 518
 OsclSocketIBase, 524
 OsclTCPSocket, 549
 OsclTCPSocketI, 552
 SendParam, 615
 SendParam, 615
 SendParam
 iBufSend, 615
 iFlags, 615
 iXferLen, 615
 SendParam, 615
 SendRequest
 OsclSendMethod, 504
 SendSuccess
 OsclSocketI, 519
 OsclSocketIBase, 524
 SendTo
 OsclSendToMethod, 506
 OsclSendToRequest, 507
 OsclSocketI, 519
 OsclSocketIBase, 524
 OsclUDPSocket, 582
 OsclUDPSocketI, 585
 SendToParam, 616
 SendToParam, 616
 SendToParam
 ~SendToParam, 616
 iAddr, 616
 iBufSend, 616
 iFlags, 616
 iXferLen, 616
 SendToParam, 616
 SendToRequest
 OsclSendToMethod, 506
 SendToSuccess
 OsclSocketI, 519
 OsclSocketIBase, 524
 Serv
 OsclDNSRequestAO, 352

Set
 OsclDoubleRunner, 358
 OsclNameString, 442
 OsclPtr, 458
 OsclPtrC, 461

set
 CHheapRep, 126
 CStackRep, 128
 OSCL_FastString, 172
 OSCL_HeapStringA, 193
 OSCL_wFastString, 283
 OSCL_wHeapStringA, 289
 OsclExclusiveArrayPtr, 370
 OsclExclusivePtr, 373
 OsclExclusivePtrA, 376
 OsclSingleton, 513
 OsclTLS, 570
 OsclTLSEEx, 572
 osclutil, 79, 80

set_from_ntp_time
 TimeValue, 629

set_from_system_time
 NTPTime, 164

set_int64
 Oscl_Int64_Utils, 195

set_len
 OSCL_String, 251
 OSCL_wString, 295

set_length
 OSCL_FastString, 172
 OSCL_wFastString, 283

set_next
 Oscl_Opaque_Type_Alloc_LL, 220

set_r
 CFastRep, 124

set_rep
 CHheapRep, 126
 OSCL_String, 251, 252
 OSCL_wString, 295

set_to_current_time
 NTPTime, 164
 TimeValue, 629

set_to_zero
 TimeValue, 629

set_uint64
 Oscl_Int64_Utils, 195

set_w
 CFastRep, 124

setAllocNodeFlag
 MM_AllocBlockHdr, 143

SetAsyncReadBufferSize
 Oscl_File, 178

SetBusy
 OsclActiveObject, 301

OsclTimerObject, 566
 setCheckSum
 StrCSumPtrLen, 621

SetExactFrequency
 OsclTimer, 561

SetFileHandle
 Oscl_File, 178

SetFrequency
 OsclTimer, 562

SetInUse
 OsclAsyncFileBuffer, 308

SetLength
 OsclPtr, 458
 OsclPtrC, 461

SetLoggingEnable
 Oscl_File, 179

SetLogLevel
 PVLogger, 598

SetLogLevelAndPropagate
 PVLogger, 598

setMaxSzForNewMemPoolBuffer
 OsclMemPoolResizableAllocator, 433

SetNativeAccessMode
 Oscl_File, 179

SetNativeBufferSize
 Oscl_File, 179

SetNodeLogLevelExplicit
 PVLoggerRegistry, 605

SetObserver
 OsclTimer, 562

SetOffset
 OsclAsyncFileBuffer, 308
 OsclDoubleListBase, 357

SetParent
 PVLogger, 598

SetPriority
 OsclThread, 555

setPtrLen
 StrCSumPtrLen, 621
 StrPtrLen, 624
 WStrPtrLen, 634

SetPVCacheSize
 Oscl_File, 179

SetRecvBufferSize
 OsclIPSocketI, 400
 OsclSocketI, 519
 OsclUDPSocket, 582

SetScheduler
 OsclExecSchedulerCommonBase, 384

SetStatus
 OsclActiveObject, 301
 OsclTimerObject, 566

SetSummaryStatsLoggingEnable
 Oscl_File, 179

SetTimestamp
 MediaData, 139
 SetToHead
 OsclDoubleRunner, 358
 SetToTail
 OsclDoubleRunner, 358
 setWithoutOwnership
 OSCLMemAutoPtr, 417
 ShowStats
 OsclExecSchedulerCommonBase, 384
 ShowSummaryStats
 OsclExecSchedulerCommonBase, 384
 Shutdown
 OsclShutdownMethod, 511
 OsclShutdownRequest, 512
 OsclSocketI, 519
 OsclSocketIBase, 525
 OsclTCPSocket, 549
 OsclTCPSocketI, 552
 ShutdownParam, 617
 ShutdownParam, 617
 ShutdownParam
 iHow, 617
 ShutdownParam, 617
 ShutdownRequest
 OsclShutdownMethod, 511
 Signal
 OsclSemaphore, 503
 Size
 Oscl_File, 179
 OsclAsyncFile, 306
 OsclNativeFile, 445
 size
 CFastRep, 124
 CHHeapRep, 126
 CStackRep, 128
 MM_AllocBlockHdr, 143
 MM_AllocInfo, 145
 MM_AllocQueryInfo, 147
 Oscl_Map, 210
 Oscl_Queue_Base, 228
 Oscl_Rb_Tree, 232
 Oscl_TagTree, 260
 Oscl_Vector_Base, 281
 OsclPriorityQueue, 453
 StrPtrLen, 624
 WStrPtrLen, 634
 size_type
 Oscl_Map, 207
 Oscl_Queue, 225
 Oscl_Rb_Tree, 232
 Oscl_Tag_Base, 256
 Oscl_TagTree, 258
 Oscl_TAlloc, 270
 sizeof_T
 Oscl_Linked_List_Base, 204
 Oscl_Queue_Base, 229
 Oscl_Vector_Base, 281
 skip_to_line_term
 osclutil, 80
 skip_to_whitespace
 osclutil, 80
 skip_whitespace
 osclutil, 80
 skip_whitespace_and_line_term
 osclutil, 80
 SLEEP_ONE_SEC
 osclconfig_util.h, 817
 SleepMillisec
 OsclThread, 555
 Socket
 OsclSocketI, 519
 SocketI
 OsclSocketRequestAO, 533
 SocketObserver
 OsclSocketRequestAO, 533
 SocketRequestParam, 618
 SocketRequestParam, 619
 SocketRequestParam
 iFxn, 619
 SocketRequestParam, 619
 SocketServ
 OsclIPSocketI, 400
 sort_children
 Oscl_TagTree::Node, 268
 specialFragBuffer
 OsclBinStream, 327
 Start
 OsclFileStats, 392
 Start_on_creation
 oscl_thread.h, 763
 StartAsyncRead
 OsclAsyncFileBuffer, 308
 StartCancel
 OsclSocketServRequestList, 541
 StartMethod
 OsclDNSMethod, 347
 OsclSocketMethod, 528
 StartNativeScheduler
 OsclExecSchedulerCommonBase, 384
 StartScheduler
 OsclExecSchedulerCommonBase, 384
 State
 OsclSocketServIBase, 540
 state
 OsclBinStream, 327
 state_t
 OsclBinStream, 325

StaticJump
 OsclJump, 401

stats_overhead
 MM_AuditOverheadStats, 156

Status
 OsclActiveObject, 301
 OsclTimerObject, 567

status_t
 BufFragStatusClass, 119

StatusRef
 OsclActiveObject, 301
 OsclTimerObject, 567

StopScheduler
 OsclExecSchedulerCommonBase, 384

Str
 OsclNameString, 442

StrCSumPtrLen, 620
 osclutil, 66
 StrCSumPtrLen, 621

StrCSumPtrLen
 checkSum, 621
 CheckSumType, 621
 getCheckSum, 621
 isCIEquivalentTo, 621
 operator!=, 621
 operator=, 621
 operator==, 621
 setCheckSum, 621
 setPtrLen, 621
 StrCSumPtrLen, 621

StrPtrLen, 623
 osclutil, 66
 StrPtrLen, 624

StrPtrLen
 c_str, 624
 isCIEquivalentTo, 624
 isCIPrefixOf, 624
 isLetter, 624
 len, 624
 length, 624
 operator!=, 624
 operator=, 624
 operator==, 624
 ptr, 624
 setPtrLen, 624
 size, 624
 StrPtrLen, 624

Success
 OsclDNSRequestAO, 353
 OsclRecvFromRequest, 469
 OsclRecvRequest, 472
 OsclSendRequest, 505
 OsclSendToRequest, 507
 OsclSocketRequestAO, 533

SUCCESS_ERROR
 OsclProcStatus, 456

Suspend
 OsclThread, 555

Suspend_on_creation
 oscl_thread.h, 763

SuspendScheduler
 OsclExecSchedulerCommonBase, 385

swap
 Oscl_Opaque_Type_Compare, 221
 OsclPriorityQueue, 453

SYSTEM_RESOURCES_UNAVAILABLE_-
 ERROR
 OsclProcStatus, 457

tag
 MM_AllocQueryInfo, 147
 MM_Stats_CB, 158
 Oscl_Tag, 253
 Oscl_TagTree::Node, 268
 OsclMemStatsNode, 439

tag_ancestor
 Oscl_Tag_Base, 256

tag_base_type
 Oscl_Tag_Base, 256
 Oscl_TagTree, 258

tag_base_unit
 Oscl_Tag_Base, 256

tag_cmp
 Oscl_Tag_Base, 256

tag_copy
 Oscl_Tag_Base, 256

tag_depth
 Oscl_Tag_Base, 256

tag_len
 Oscl_Tag_Base, 256

tag_type
 Oscl_TagTree, 258

tagAllocator
 Oscl_Tag, 253

TagTree_Allocator
 osclmemory, 56

Tail
 OsclDoubleList, 355
 OsclPriorityList, 450

tail
 Oscl_Linked_List_Base, 204

takeOwnership
 OSCLMemAutoPtr, 418

TDNSRequestParamAllocator
 oscl_dns_param.h, 650

Tell
 Oscl_File, 180
 OsclAsyncFile, 306

OsclFileCache, 390
 OsclNativeFile, 445
 tellg
 OsclBinStream, 326
 Terminate
 OsclThread, 556
 the_list
 Oscl_MTLinked_List, 216
 THREAD_1_INACTIVE_ERROR
 OsclProcStatus, 456
 THREAD_BLOCK_ERROR
 OsclProcStatus, 457
 THREAD_NOT_OWN_MUTEX_ERROR
 OsclProcStatus, 457
 ThreadHasScheduler
 PVThreadContext, 611
 ThreadLogoff
 OsclReadyQ, 466
 ThreadLogon
 OsclReadyQ, 466
 ThreadPriorityAboveNormal
 oscl_thread.h, 764
 ThreadPriorityBelowNormal
 oscl_thread.h, 763
 ThreadPriorityHighest
 oscl_thread.h, 764
 ThreadPriorityLow
 oscl_thread.h, 763
 ThreadPriorityLowest
 oscl_thread.h, 763
 ThreadPriorityNormal
 oscl_thread.h, 763
 ThreadPriorityTimeCritical
 oscl_thread.h, 764
 TickCount
 OsclTickCount, 558
 TickCountFrequency
 OsclTickCount, 558
 TickCountPeriod
 OsclTickCount, 558
 TicksToMsec
 OsclTickCount, 558
 TimeoutOccurred
 OsclTimerObserver, 568
 TimerBaseElapsed
 CallbackTimerObserver, 122
 OsclTimer, 562
 TimerCallback
 OsclReadyQ, 466
 timestamp
 MediaData, 139
 TimeUnits
 osclbase, 33
 TimeValue, 625
 TimeValue, 626, 627
 TimeValue
 get_local_time, 627
 get_pv8601_str_time, 627
 get_rfc822_gmtime_str, 627
 get_sec, 628
 get_str_ctime, 628
 get_timeval_ptr, 628
 get_usec, 628
 is_zero, 628
 NTPTime, 630
 operator *=, 629
 operator!=, 630
 operator+=, 629
 operator-=, 629
 operator<, 630
 operator<=, 630
 operator=, 629
 operator==, 630
 operator>, 630
 operator>=, 630
 set_from_ntp_time, 629
 set_to_current_time, 629
 set_to_zero, 629
 TimeValue, 626, 627
 to_msec, 629
 TLSStorageOps, 631
 TLSStorageOps
 get_registry, 631
 save_registry, 631
 to_msec
 TimeValue, 629
 to_system_time
 NTPTime, 164
 TOO_MANY_FRAGS
 BuffFragStatusClass, 119
 TOO_MANY_THREADS_ERROR
 OsclProcStatus, 456
 Top
 OsclJump, 401
 OsclReadyQ, 466
 OsclTimerQ, 569
 top
 OsclPriorityQueue, 453
 TOsclBasicLockObject
 osclconfig_unix_common.h, 812
 osclconfig_unix_nj.h, 816
 TOsclConditionObject
 osclconfig_proc_unix_common.h, 804
 osclconfig_proc_unix_nj.h, 806
 TOsclFileHandle
 osclio, 93
 TOsclFileOp
 osclio, 94

TOsclHostent
osclconfig_io.h, 791

TOsclMutexObject
osclconfig_proc_unix_common.h, 804
osclconfig_proc_unix_nj.h, 806

TOsclReady
osclproc, 102

TOsclSemaphoreObject
osclconfig_proc_unix_common.h, 804
osclconfig_proc_unix_nj.h, 806

TOsclSockAddr
osclconfig_io.h, 791

TOsclSockAddrLen
osclconfig_io.h, 791

TOsclSocket
osclconfig_io.h, 791

TOsclSocketServStatEvent
oscl_socket_stats.h, 746

TOsclSocketStatEvent
oscl_socket_stats.h, 746

TOsclThreadFuncArg
osclconfig_proc_unix_common.h, 804
osclconfig_proc_unix_nj.h, 806

TOsclThreadFuncPtr
oscl_thread.h, 763

TOsclThreadFuncRet
osclconfig_proc_unix_common.h, 804
osclconfig_proc_unix_nj.h, 806

TOsclThreadId
osclconfig_proc_unix_common.h, 804
osclconfig_proc_unix_nj.h, 806

TOsclThreadObject
osclconfig_proc_unix_common.h, 804
osclconfig_proc_unix_nj.h, 806

TOsclTlsKey
osclbase, 33
osclconfig_unix_common.h, 812
osclconfig_unix_nj.h, 816

totalbytes
oscl_fstat, 187

totalNumAllocs
MM_Stats_t, 160

totalNumBytes
MM_Stats_t, 160

TOtherExecStats
OsclExecSchedulerCommonBase, 382

TPVDNSEvent
osclio, 94

TPVDNSFxn
osclio, 95

TPVSocketEvent
oscl_socket_types.h, 750

TPVSocketFxn
oscl_socket_types.h, 750

TPVSocketShutdown
oscl_socket_types.h, 751

TPVThreadContext
osclproc, 102

Trap
OsclErrorTrapImp, 365

TrapNoTls
OsclErrorTrapImp, 365

TReadyQueLink
TReadyQueLink, 632

TReadyQueLink
iAOPriority, 632
iIsIn, 632
iSeqNum, 632
iTimeQueuedTicks, 632
iTimeToRunTicks, 632
TReadyQueLink, 632

trim
OsclMemPoolResizableAllocator, 433

TryLock
OsclMutex, 441

TryWait
OsclSemaphore, 503

TSocketServState
OsclSocketServIBase, 539

TSymbianAccessMode
Oscl_File, 175

uint
osclbase, 33

UINT64
osclconfig_unix_common.h, 812
osclconfig_unix_nj.h, 816

uint64
osclbase, 33

UINT64_HILO
osclconfig_unix_common.h, 812
osclconfig_unix_nj.h, 816

Unbind
OsclSharedPtr, 510

UninstallScheduler
OsclExecSchedulerCommonBase, 385

unix_ntp_offset
osclbase, 43

Unlock
OsclLockBase, 404
OsclMutex, 441
OsclNullLock, 448
OsclThreadLock, 557

UnRegister
OsclRegistryClient, 491
OsclRegistryClientImpl, 493
OsclRegistryServTlsImpl, 496

Unregister

OsclComponentRegistry, 332
 UnTrap
 OsclErrorTrapImp, 365
 update
 MM_Stats_t, 160
 UpdateData
 OsclAsyncFileBuffer, 308
 updateStatsNode
 MM_Audit_Imp, 154
 updateStatsNodeInFailure
 MM_Audit_Imp, 154
 UpdateTimers
 OsclExecSchedulerCommonBase, 385
 UpdateTimersMsec
 OsclExecSchedulerCommonBase, 385
 upper_bound
 Oscl_Map, 210, 211
 Oscl_Rb_Tree, 232
 USEC_PER_SEC
 osclbase, 43

 validate
 MM_Audit_Imp, 154
 OsclPriorityQueue, 454
 validate_all_heap
 MM_Audit_Imp, 154
 validateblock
 OsclMemPoolResizableAllocator, 433
 Value
 OsclAOStatus, 303
 value
 Oscl_Rb_Tree_Node, 241
 Oscl_TagTree::Node, 268
 value_comp
 Oscl_Map, 211
 value_compare
 Oscl_Map::value_compare, 212
 value_type
 Oscl_Map, 207
 Oscl_Queue, 225
 Oscl_Rb_Tree, 232
 Oscl_Rb_Tree_Const_Iterator, 236
 Oscl_Rb_Tree_Iterator, 239
 Oscl_Rb_Tree_Node, 241
 Oscl_TagTree, 258
 Oscl_TAlloc, 270
 Oscl_Vector, 274
 OsclPriorityQueue, 452
 vec
 OsclPriorityQueue, 454

 Wait
 OsclSemaphore, 503
 WAIT_ABANDONED_ERROR

 OsclProcStatus, 457
 WAIT_TIMEOUT_ERROR
 OsclProcStatus, 457
 WaitAndPopTop
 OsclReadyQ, 466
 WaitForReadyAO
 OsclExecSchedulerCommonBase, 385
 WaitForRequestComplete
 OsclReadyQ, 466
 WaitOnRequests
 OsclSocketServRequestList, 541
 Wakeup
 OsclSocketServRequestList, 541
 writable
 CFastRep, 124
 Write
 Oscl_File, 180
 OsclAsyncFile, 306
 OsclFileCache, 390
 OsclNativeFile, 445
 write
 OSCL_String, 252
 OSCL_wString, 295
 OsclBinOStream, 319
 WriteUnsignedLong
 OsclBinOStreamBigEndian, 321
 OsclBinOStreamLittleEndian, 323
 WriteUnsignedShort
 OsclBinOStreamBigEndian, 321
 OsclBinOStreamLittleEndian, 323
 WStrPtrLen, 633
 osclutil, 66
 WStrPtrLen, 634
 WStrPtrLen
 c_str, 634
 isCIEquivalentTo, 634
 len, 634
 length, 634
 operator!=, 634
 operator=, 634
 operator==, 634
 ptr, 634
 setPtrLen, 634
 size, 634
 WStrPtrLen, 634

 xsubi
 MM_FailInsertParam, 157

 Zero
 OsclPtr, 458
 OsclPtrC, 461