



# Open CORE™

OSCL API

Build Version: CORE\_9.002.1.1

Jun 13, 2010



# Contents

<b>1</b>	<b>Todo List</b>	<b>1</b>
<b>2</b>	<b>Module Index</b>	<b>3</b>
2.1	Modules . . . . .	3
<b>3</b>	<b>Data Structure Index</b>	<b>5</b>
3.1	Class Hierarchy . . . . .	5
<b>4</b>	<b>Data Structure Index</b>	<b>11</b>
4.1	Data Structures . . . . .	11
<b>5</b>	<b>File Index</b>	<b>17</b>
5.1	File List . . . . .	17
<b>6</b>	<b>Module Documentation</b>	<b>23</b>
6.1	OSCL config . . . . .	23
6.1.1	Define Documentation . . . . .	25
6.1.1.1	OSCL_HAS_ANDROID_SUPPORT . . . . .	25
6.1.1.2	OSCL_HAS_BERKELEY_SOCKETS . . . . .	25
6.1.1.3	OSCL_HAS_IPHONE_SUPPORT . . . . .	25
6.1.1.4	OSCL_HAS_MSWIN_PARTIAL_SUPPORT . . . . .	25
6.1.1.5	OSCL_HAS_MSWIN_SUPPORT . . . . .	25
6.1.1.6	OSCL_HAS_PTHREAD_SUPPORT . . . . .	25
6.1.1.7	OSCL_HAS_PV_C_OS_API_MEMORY_FUNCS . . . . .	25
6.1.1.8	OSCL_HAS_PV_C_OS_SUPPORT . . . . .	25
6.1.1.9	OSCL_HAS_PV_C_OS_TIME_FUNCS . . . . .	25
6.1.1.10	OSCL_HAS_SAVAJE_IO_SUPPORT . . . . .	25
6.1.1.11	OSCL_HAS_SAVAJE_SUPPORT . . . . .	25
6.1.1.12	OSCL_HAS_SEM_TIMEDWAIT_SUPPORT . . . . .	25
6.1.1.13	OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION . . . . .	25

6.1.1.14	OSCL_HAS_SYMBIAN_DNS_SERVER . . . . .	25
6.1.1.15	OSCL_HAS_SYMBIAN_ERRORTRAP . . . . .	25
6.1.1.16	OSCL_HAS_SYMBIAN_MATH . . . . .	25
6.1.1.17	OSCL_HAS_SYMBIAN_MEMORY_FUNCS . . . . .	25
6.1.1.18	OSCL_HAS_SYMBIAN_SCHEDULER . . . . .	25
6.1.1.19	OSCL_HAS_SYMBIAN_SOCKET_SERVER . . . . .	25
6.1.1.20	OSCL_HAS_SYMBIAN_SUPPORT . . . . .	25
6.1.1.21	OSCL_HAS_SYMBIAN_TIMERS . . . . .	25
6.1.1.22	OSCL_HAS_UNIX_SUPPORT . . . . .	25
6.1.1.23	OSCL_HAS_UNIX_TIME_FUNCS . . . . .	25
6.1.2	Typedef Documentation . . . . .	25
6.1.2.1	<u>_int16_check_</u> . . . . .	25
6.1.2.2	<u>_int32_check_</u> . . . . .	25
6.1.2.3	<u>_int8_check_</u> . . . . .	25
6.1.2.4	<u>_uint16_check_</u> . . . . .	25
6.1.2.5	<u>_uint32_check_</u> . . . . .	25
6.1.2.6	<u>_uint8_check_</u> . . . . .	25
6.2	OSCL Base . . . . .	26
6.2.1	Detailed Description . . . . .	33
6.2.2	Define Documentation . . . . .	34
6.2.2.1	ALLOC_AND_CONSTRUCT . . . . .	34
6.2.2.2	ALLOCATE . . . . .	34
6.2.2.3	EPV_ARM_GNUC . . . . .	34
6.2.2.4	EPV_ARM_MSEVC . . . . .	34
6.2.2.5	EPV_ARM_RVCT . . . . .	34
6.2.2.6	NULL . . . . .	34
6.2.2.7	OSCL_ABS . . . . .	34
6.2.2.8	OSCL_ASSERT . . . . .	35
6.2.2.9	OSCL_COND_EXPORT_REF . . . . .	35
6.2.2.10	OSCL_COND_IMPORT_REF . . . . .	35
6.2.2.11	OSCL_CONST_CAST . . . . .	35
6.2.2.12	OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT . . . . .	35
6.2.2.13	OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE . . . . .	35
6.2.2.14	OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE . . . . .	35
6.2.2.15	OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE . . . . .	35
6.2.2.16	OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE . . . . .	35

6.2.2.17	OSCL_DLL_ENTRY_POINT . . . . .	35
6.2.2.18	OSCL_DLL_ENTRY_POINT_DEFAULT . . . . .	36
6.2.2.19	OSCL_DYNAMIC_CAST . . . . .	36
6.2.2.20	OSCL_HAS_SINGLETON_SUPPORT . . . . .	36
6.2.2.21	OSCL_INLINE . . . . .	36
6.2.2.22	OSCL_MAX . . . . .	36
6.2.2.23	OSCL_MIN . . . . .	36
6.2.2.24	OSCL_REINTERPRET_CAST . . . . .	36
6.2.2.25	OSCL_STATIC_CAST . . . . .	36
6.2.2.26	OSCL_TLS_BASE_SLOTS . . . . .	36
6.2.2.27	OSCL_TLS_MAX_SLOTS . . . . .	36
6.2.2.28	OSCL_UNUSED_ARG . . . . .	36
6.2.2.29	OSCL_UNUSED_RETURN . . . . .	37
6.2.2.30	OSCL_VIRTUAL_BASE . . . . .	37
6.2.2.31	PVMEM_INST_LEVEL . . . . .	37
6.2.3	Typedef Documentation . . . . .	37
6.2.3.1	c_bool . . . . .	37
6.2.3.2	CtimeStrBuf . . . . .	37
6.2.3.3	int64 . . . . .	37
6.2.3.4	ISO8601timeStrBuf . . . . .	37
6.2.3.5	mbchar . . . . .	37
6.2.3.6	octet . . . . .	37
6.2.3.7	OSCL_TCHAR . . . . .	37
6.2.3.8	oscl_wchar . . . . .	37
6.2.3.9	OsclAny . . . . .	37
6.2.3.10	OsclFloat . . . . .	38
6.2.3.11	OsclSizeT . . . . .	38
6.2.3.12	PV8601timeStrBuf . . . . .	38
6.2.3.13	TOsclTlsKey . . . . .	38
6.2.3.14	uint . . . . .	38
6.2.3.15	uint64 . . . . .	38
6.2.4	Enumeration Type Documentation . . . . .	38
6.2.4.1	TimeUnits . . . . .	38
6.2.5	Function Documentation . . . . .	38
6.2.5.1	_OSCL_Abort . . . . .	38
6.2.5.2	big_endian_to_host . . . . .	38

6.2.5.3	Bind . . . . .	39
6.2.5.4	Bind . . . . .	39
6.2.5.5	get_count . . . . .	39
6.2.5.6	GetRefCounter . . . . .	39
6.2.5.7	GetRep . . . . .	39
6.2.5.8	host_to_big_endian . . . . .	39
6.2.5.9	host_to_little_endian . . . . .	39
6.2.5.10	ISO8601ToRFC822 . . . . .	40
6.2.5.11	little_endian_to_host . . . . .	40
6.2.5.12	operator TheClass *	40
6.2.5.13	operator* . . . . .	40
6.2.5.14	operator+ . . . . .	40
6.2.5.15	operator+ . . . . .	40
6.2.5.16	operator- . . . . .	40
6.2.5.17	operator- . . . . .	40
6.2.5.18	operator- . . . . .	40
6.2.5.19	operator-> . . . . .	40
6.2.5.20	operator= . . . . .	41
6.2.5.21	operator== . . . . .	41
6.2.5.22	OSCL Assert . . . . .	41
6.2.5.23	oscl_CIstrcmp . . . . .	41
6.2.5.24	oscl_CIstrcmp . . . . .	41
6.2.5.25	oscl_CIstrncmp . . . . .	42
6.2.5.26	oscl_CIstrncmp . . . . .	42
6.2.5.27	oscl_isLetter . . . . .	42
6.2.5.28	oscl_streat . . . . .	42
6.2.5.29	oscl_streat . . . . .	43
6.2.5.30	oscl_strchr . . . . .	43
6.2.5.31	oscl_strchr . . . . .	43
6.2.5.32	oscl_strchr . . . . .	43
6.2.5.33	oscl_strchr . . . . .	43
6.2.5.34	oscl_strcmp . . . . .	44
6.2.5.35	oscl_strcmp . . . . .	44
6.2.5.36	oscl_strlen . . . . .	44
6.2.5.37	oscl_strlen . . . . .	44
6.2.5.38	oscl_strncat . . . . .	45

6.2.5.39	oscl_strncat . . . . .	45
6.2.5.40	oscl_strcmp . . . . .	45
6.2.5.41	oscl_strncmp . . . . .	46
6.2.5.42	oscl_strncpy . . . . .	46
6.2.5.43	oscl_strncpy . . . . .	47
6.2.5.44	oscl strrchr . . . . .	47
6.2.5.45	oscl strrchr . . . . .	47
6.2.5.46	oscl strrchr . . . . .	47
6.2.5.47	oscl strrchr . . . . .	47
6.2.5.48	oscl strset . . . . .	47
6.2.5.49	oscl strset . . . . .	48
6.2.5.50	oscl strstr . . . . .	48
6.2.5.51	oscl strstr . . . . .	48
6.2.5.52	oscl strstr . . . . .	48
6.2.5.53	oscl strstr . . . . .	48
6.2.5.54	oscl_tolower . . . . .	48
6.2.5.55	oscl_tolower . . . . .	49
6.2.5.56	OsclSharedPtr . . . . .	49
6.2.5.57	OsclSharedPtr . . . . .	49
6.2.5.58	PV8601ToRFC822 . . . . .	49
6.2.5.59	PVOsclBase_Cleanup . . . . .	49
6.2.5.60	PVOsclBase_Init . . . . .	49
6.2.5.61	RFC822ToPV8601 . . . . .	50
6.2.5.62	Unbind . . . . .	50
6.2.5.63	~OsclSharedPtr . . . . .	50
6.2.6	Variable Documentation . . . . .	51
6.2.6.1	CTIME_BUFFER_SIZE . . . . .	51
6.2.6.2	ISO8601TIME_BUFFER_SIZE . . . . .	51
6.2.6.3	MSEC_PER_SEC . . . . .	51
6.2.6.4	OSCL_TLS_ID_BASE_LAST . . . . .	51
6.2.6.5	OSCL_TLS_ID_ERRORHOOK . . . . .	51
6.2.6.6	OSCL_TLS_ID_MAGICNUM . . . . .	51
6.2.6.7	OSCL_TLS_ID_OSCLREGISTRY . . . . .	51
6.2.6.8	OSCL_TLS_ID_PAYLOADPARSER . . . . .	51
6.2.6.9	OSCL_TLS_ID_PVERRORTRAP . . . . .	51
6.2.6.10	OSCL_TLS_ID_PVLOGGER . . . . .	51

6.2.6.11	OSCL_TLS_ID_PVMFRECOGNIZER	51
6.2.6.12	OSCL_TLS_ID_PVSCHEDULER	51
6.2.6.13	OSCL_TLS_ID_SDPMEDIAPARSER	51
6.2.6.14	OSCL_TLS_ID_SQLITE3	51
6.2.6.15	OSCL_TLS_ID_TEST	51
6.2.6.16	OSCL_TLS_ID_WMDRM	51
6.2.6.17	PV8601TIME_BUFFER_SIZE	51
6.2.6.18	unix_ntp_offset	51
6.2.6.19	USEC_PER_SEC	51
6.3	OSCL Memory	52
6.3.1	Define Documentation	55
6.3.1.1	_OSCL_CLEANUP_BASE_CLASS	55
6.3.1.2	_OSCL_TRAP_NEW	55
6.3.1.3	COMPUTE_MEM_ALIGN_SIZE	56
6.3.1.4	DEFAULT_MM_AUDIT_MODE	56
6.3.1.5	DEFAULT_POSTFILL_PATTERN	56
6.3.1.6	DEFAULT_PREFILL_PATTERN	56
6.3.1.7	FENCE_PATTERN	56
6.3.1.8	MEM_ALIGN_SIZE	56
6.3.1.9	MIN_FENCE_SIZE	56
6.3.1.10	MM_ALLOC_MAX_QUERY_FILENAME_LEN	56
6.3.1.11	MM_ALLOC_MAX_QUERY_TAG_LEN	56
6.3.1.12	MM_AUDIT_ALLOC_NODE_ENABLE_FLAG	56
6.3.1.13	MM_AUDIT_ALLOC_NODE_SUPPORT	56
6.3.1.14	MM_AUDIT_FAILURE_SIMULATION_SUPPORT	56
6.3.1.15	MM_AUDIT_FENCE_SUPPORT	56
6.3.1.16	MM_AUDIT_FILL_SUPPORT	56
6.3.1.17	MM_AUDIT_INCLUDE_ALL_HEAP_VALIDATION	56
6.3.1.18	MM_AUDIT_POSTFILL_FLAG	56
6.3.1.19	MM_AUDIT_PREFILL_FLAG	56
6.3.1.20	MM_AUDIT_SUPPRESS_FILENAME_FLAG	56
6.3.1.21	MM_AUDIT_VALIDATE_ALL_HEAP_FLAG	56
6.3.1.22	MM_AUDIT_VALIDATE_BLOCK	56
6.3.1.23	MM_AUDIT_VALIDATE_ON_FREE_FLAG	56
6.3.1.24	OSCL_ALLOC_DELETE	56
6.3.1.25	OSCL_ALLOC_NEW	57

6.3.1.26	OSCL_ARRAY_DELETE . . . . .	57
6.3.1.27	OSCL_ARRAY_NEW . . . . .	57
6.3.1.28	OSCL_AUDIT_ARRAY_NEW . . . . .	58
6.3.1.29	OSCL_AUDIT_CALLOC . . . . .	58
6.3.1.30	OSCL_AUDIT_MALLOC . . . . .	59
6.3.1.31	OSCL_AUDIT_NEW . . . . .	59
6.3.1.32	OSCL_AUDIT_REALLOC . . . . .	59
6.3.1.33	oscl_malloc . . . . .	60
6.3.1.34	OSCL_CALLOC . . . . .	60
6.3.1.35	OSCL_CLEANUP_BASE_CLASS . . . . .	60
6.3.1.36	OSCL_DEFAULT_FREE . . . . .	60
6.3.1.37	OSCL_DEFAULT_MALLOC . . . . .	60
6.3.1.38	OSCL_DELETE . . . . .	60
6.3.1.39	OSCL_DISABLE_WARNING_RETURN_TYPE_NOT_UDT . . . . .	61
6.3.1.40	OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE . . . . .	61
6.3.1.41	OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE . . . . .	61
6.3.1.42	OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE . . . . .	61
6.3.1.43	OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE . . . . .	61
6.3.1.44	OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE . . . . .	61
6.3.1.45	oscl_free . . . . .	61
6.3.1.46	OSCL_FREE . . . . .	61
6.3.1.47	oscl_malloc . . . . .	61
6.3.1.48	OSCL_MALLOC . . . . .	61
6.3.1.49	OSCL_NEW . . . . .	62
6.3.1.50	OSCL_PLACEMENT_NEW . . . . .	62
6.3.1.51	oscl_realloc . . . . .	62
6.3.1.52	OSCL_REALLOC . . . . .	62
6.3.1.53	OSCL_TRAP_ALLOC_NEW . . . . .	62
6.3.1.54	OSCL_TRAP_AUDIT_NEW . . . . .	63
6.3.1.55	OSCL_TRAP_NEW . . . . .	63
6.3.2	Typedef Documentation . . . . .	64
6.3.2.1	MM_AllocNodeAutoPtr . . . . .	64
6.3.2.2	MM_StatsNodeTagTreeType . . . . .	64
6.3.2.3	MMAuditCharAutoPtr . . . . .	64
6.3.2.4	MMAuditUint8AutoPtr . . . . .	64
6.3.2.5	OsclMemStatsNodeAutoPtr . . . . .	64

6.3.2.6	OsclTagTreeType . . . . .	64
6.3.2.7	TagTree_Allocator . . . . .	64
6.3.3	Function Documentation . . . . .	64
6.3.3.1	_oscl_calloc . . . . .	64
6.3.3.2	_oscl_default_new . . . . .	64
6.3.3.3	_oscl_free . . . . .	64
6.3.3.4	_oscl_malloc . . . . .	64
6.3.3.5	_oscl_realloc . . . . .	64
6.3.3.6	operator delete . . . . .	64
6.3.3.7	operator delete[] . . . . .	65
6.3.3.8	operator new . . . . .	65
6.3.3.9	operator new[] . . . . .	65
6.3.3.10	oscl_mem_aligned_size . . . . .	65
6.3.3.11	oscl_memcmp . . . . .	65
6.3.3.12	oscl_memcpy . . . . .	65
6.3.3.13	oscl_memmove . . . . .	66
6.3.3.14	oscl_memmove32 . . . . .	66
6.3.3.15	oscl_memset . . . . .	66
6.3.3.16	OsclMemInit . . . . .	67
6.3.4	Variable Documentation . . . . .	67
6.3.4.1	ALLOC_NODE_FLAG . . . . .	67
6.4	OSCL Util . . . . .	68
6.4.1	Define Documentation . . . . .	74
6.4.1.1	MAX_NUMBER_OF_BYTE_PER_UTF8 . . . . .	74
6.4.1.2	oscl_isdigit . . . . .	74
6.4.1.3	OSCLTICKCOUNT_MAX_TICKS . . . . .	74
6.4.2	Typedef Documentation . . . . .	74
6.4.2.1	BufferFreeFuncPtr . . . . .	74
6.4.2.2	MediaTimestamp . . . . .	74
6.4.2.3	OSCL_TStrPtrLen . . . . .	74
6.4.2.4	OsclComponentFactory . . . . .	74
6.4.2.5	StrCSumPtrLen . . . . .	74
6.4.2.6	StrPtrLen . . . . .	75
6.4.2.7	WStrPtrLen . . . . .	75
6.4.3	Enumeration Type Documentation . . . . .	75
6.4.3.1	TOSCL_StringOp . . . . .	75

6.4.3.2	TOSCL_wStringOp . . . . .	75
6.4.4	Function Documentation . . . . .	75
6.4.4.1	extract_string . . . . .	75
6.4.4.2	extract_string . . . . .	75
6.4.4.3	get_cstr . . . . .	75
6.4.4.4	get_cstr . . . . .	76
6.4.4.5	get_cstr . . . . .	76
6.4.4.6	get_cstr . . . . .	76
6.4.4.7	get_maxsize . . . . .	76
6.4.4.8	get_maxsize . . . . .	76
6.4.4.9	get_maxsize . . . . .	76
6.4.4.10	get_maxsize . . . . .	77
6.4.4.11	get_size . . . . .	77
6.4.4.12	get_size . . . . .	77
6.4.4.13	get_size . . . . .	77
6.4.4.14	get_size . . . . .	77
6.4.4.15	get_str . . . . .	77
6.4.4.16	get_str . . . . .	78
6.4.4.17	get_str . . . . .	78
6.4.4.18	get_str . . . . .	78
6.4.4.19	GetBufferState . . . . .	78
6.4.4.20	GetFragment . . . . .	78
6.4.4.21	operator= . . . . .	78
6.4.4.22	operator= . . . . .	78
6.4.4.23	operator= . . . . .	79
6.4.4.24	operator= . . . . .	79
6.4.4.25	operator= . . . . .	79
6.4.4.26	operator= . . . . .	79
6.4.4.27	operator= . . . . .	79
6.4.4.28	operator= . . . . .	79
6.4.4.29	operator= . . . . .	80
6.4.4.30	operator= . . . . .	80
6.4.4.31	operator= . . . . .	80
6.4.4.32	operator= . . . . .	80
6.4.4.33	oscl_abs . . . . .	80
6.4.4.34	oscl_asin . . . . .	80

6.4.4.35	oscl_atan . . . . .	80
6.4.4.36	oscl_cos . . . . .	81
6.4.4.37	oscl_exp . . . . .	81
6.4.4.38	oscl_floor . . . . .	81
6.4.4.39	OSCL_HeapString . . . . .	81
6.4.4.40	OSCL_HeapString . . . . .	81
6.4.4.41	OSCL_HeapString . . . . .	81
6.4.4.42	OSCL_HeapString . . . . .	82
6.4.4.43	OSCL_HeapString . . . . .	82
6.4.4.44	oscl_log . . . . .	82
6.4.4.45	oscl_log10 . . . . .	82
6.4.4.46	oscl_pow . . . . .	82
6.4.4.47	oscl_sin . . . . .	82
6.4.4.48	oscl_snprintf . . . . .	83
6.4.4.49	oscl_snprintf . . . . .	83
6.4.4.50	oscl_sqrt . . . . .	83
6.4.4.51	OSCL_StackString . . . . .	83
6.4.4.52	OSCL_StackString . . . . .	83
6.4.4.53	OSCL_StackString . . . . .	83
6.4.4.54	OSCL_StackString . . . . .	83
6.4.4.55	OSCL_StackString . . . . .	84
6.4.4.56	oscl_str_escape_xml . . . . .	84
6.4.4.57	oscl_str_is_valid_utf8 . . . . .	84
6.4.4.58	oscl_str_need_escape_xml . . . . .	85
6.4.4.59	oscl_str_truncate_utf8 . . . . .	85
6.4.4.60	oscl_str_unescape_uri . . . . .	86
6.4.4.61	oscl_str_unescape_uri . . . . .	86
6.4.4.62	oscl_tan . . . . .	86
6.4.4.63	oscl_UnicodeToUTF8 . . . . .	87
6.4.4.64	oscl_UTF8ToUnicode . . . . .	87
6.4.4.65	oscl_vsnprintf . . . . .	88
6.4.4.66	oscl_vsnprintf . . . . .	88
6.4.4.67	OSCL_wHeapString . . . . .	88
6.4.4.68	OSCL_wHeapString . . . . .	88
6.4.4.69	OSCL_wHeapString . . . . .	88
6.4.4.70	OSCL_wHeapString . . . . .	88

6.4.4.71	OSCL_wHeapString . . . . .	88
6.4.4.72	OSCL_wStackString . . . . .	88
6.4.4.73	OSCL_wStackString . . . . .	88
6.4.4.74	OSCL_wStackString . . . . .	88
6.4.4.75	OSCL_wStackString . . . . .	88
6.4.4.76	OSCL_wStackString . . . . .	89
6.4.4.77	PV_atof . . . . .	89
6.4.4.78	PV_atof . . . . .	89
6.4.4.79	PV_atoi . . . . .	89
6.4.4.80	PV_atoi . . . . .	89
6.4.4.81	PV_atoi . . . . .	89
6.4.4.82	set . . . . .	89
6.4.4.83	set . . . . .	89
6.4.4.84	set . . . . .	89
6.4.4.85	set . . . . .	89
6.4.4.86	set . . . . .	90
6.4.4.87	set . . . . .	90
6.4.4.88	set . . . . .	90
6.4.4.89	set . . . . .	90
6.4.4.90	set . . . . .	90
6.4.4.91	set . . . . .	90
6.4.4.92	set . . . . .	91
6.4.4.93	set . . . . .	91
6.4.4.94	skip_to_line_term . . . . .	91
6.4.4.95	skip_to_whitespace . . . . .	91
6.4.4.96	skip_whitespace . . . . .	91
6.4.4.97	skip_whitespace . . . . .	91
6.4.4.98	skip_whitespace . . . . .	91
6.4.4.99	skip_whitespace_and_line_term . . . . .	91
6.4.4.100	~OSCL_HeapString . . . . .	91
6.4.4.101	~OSCL_StackString . . . . .	91
6.4.4.102	~OSCL_wHeapString . . . . .	91
6.4.4.103	~OSCL_wStackString . . . . .	92
6.4.5	Variable Documentation . . . . .	92
6.4.5.1	APPEND_MEDIA_AT_END . . . . .	92
6.4.5.2	OSCL_ASCII_CASE_MAGIC_BIT . . . . .	92

6.5 OSCL Error . . . . .	93
6.5.1 Define Documentation . . . . .	96
6.5.1.1 _PV_TRAP . . . . .	96
6.5.1.2 _PV_TRAP . . . . .	96
6.5.1.3 _PV_TRAP . . . . .	96
6.5.1.4 _PV_TRAP_NO_TLS . . . . .	97
6.5.1.5 _PV_TRAP_NO_TLS . . . . .	97
6.5.1.6 _PV_TRAP_NO_TLS . . . . .	97
6.5.1.7 internalLeave . . . . .	97
6.5.1.8 OSCL_BAD_ALLOC_EXCEPTION_CODE . . . . .	97
6.5.1.9 OSCL_CATCH . . . . .	97
6.5.1.10 OSCL_CATCH_ANY . . . . .	98
6.5.1.11 OSCL_ERR_NONE . . . . .	98
6.5.1.12 OSCL_FIRST_CATCH . . . . .	98
6.5.1.13 OSCL_FIRST_CATCH_ANY . . . . .	98
6.5.1.14 OSCL_JUMP_MAX_JUMP_MARKS . . . . .	98
6.5.1.15 OSCL_LAST_CATCH . . . . .	98
6.5.1.16 OSCL_LEAVE . . . . .	99
6.5.1.17 OSCL_MAX_TRAP_LEVELS . . . . .	99
6.5.1.18 OSCL_TRAPSTACK_POP . . . . .	99
6.5.1.19 OSCL_TRAPSTACK_POPDEALLOC . . . . .	99
6.5.1.20 OSCL_TRAPSTACK_PUSH . . . . .	99
6.5.1.21 OSCL_TRY . . . . .	99
6.5.1.22 OSCL_TRY_NO_TLS . . . . .	99
6.5.1.23 OsclErrAlreadyExists . . . . .	99
6.5.1.24 OsclErrAlreadyInstalled . . . . .	99
6.5.1.25 OsclErrArgument . . . . .	99
6.5.1.26 OsclErrBadHandle . . . . .	100
6.5.1.27 OsclErrBusy . . . . .	100
6.5.1.28 OsclErrCancelled . . . . .	100
6.5.1.29 OsclErrCorrupt . . . . .	100
6.5.1.30 OsclErrGeneral . . . . .	100
6.5.1.31 OsclErrInvalidState . . . . .	100
6.5.1.32 OsclErrNoHandler . . . . .	100
6.5.1.33 OsclErrNoMemory . . . . .	100
6.5.1.34 OsclErrNone . . . . .	100

6.5.1.35	OsclErrNoResources . . . . .	100
6.5.1.36	OsclErrNotInstalled . . . . .	100
6.5.1.37	OsclErrNotReady . . . . .	100
6.5.1.38	OsclErrNotSupported . . . . .	100
6.5.1.39	OsclErrOverflow . . . . .	101
6.5.1.40	OsclErrSystemCallFailed . . . . .	101
6.5.1.41	OsclErrThreadContextIncorrect . . . . .	101
6.5.1.42	OsclErrTimeout . . . . .	101
6.5.1.43	OsclErrUnderflow . . . . .	101
6.5.1.44	OsclFailure . . . . .	101
6.5.1.45	OsclPending . . . . .	101
6.5.1.46	OsclSuccess . . . . .	101
6.5.1.47	PVError_DoLeave . . . . .	101
6.5.1.48	PVError_DoLeave . . . . .	101
6.5.1.49	PVError_DoLeave . . . . .	101
6.5.1.50	PVERROR_IMP_JUMPS . . . . .	101
6.5.1.51	PVERRORTRAP_REGISTRY . . . . .	101
6.5.1.52	PVERRORTRAP_REGISTRY_ID . . . . .	101
6.5.2	Typedef Documentation . . . . .	101
6.5.2.1	OsclLeaveCode . . . . .	101
6.5.2.2	OsclReturnCode . . . . .	101
6.5.2.3	OsclTrapOperation . . . . .	101
6.5.3	Function Documentation . . . . .	101
6.5.3.1	OSCL_GetLastError . . . . .	101
6.5.3.2	OSCL_IsErnoSupported . . . . .	102
6.5.3.3	OSCL_SetLastError . . . . .	102
6.5.3.4	OSCL_StrError . . . . .	102
6.6	OSCL IO . . . . .	103
6.6.1	Define Documentation . . . . .	108
6.6.1.1	OSCL_FILE_CHAR_PATH_DELIMITER . . . . .	108
6.6.1.2	OSCL_FILE_STATS_LOGGER_NODE . . . . .	108
6.6.1.3	OSCL_FILE_WCHAR_PATH_DELIMITER . . . . .	108
6.6.1.4	OSCL_IO_EXTENSION_MAXLEN . . . . .	108
6.6.1.5	OSCL_IO_FILENAME_MAXLEN . . . . .	108
6.6.1.6	TOsclFileOffsetInt32 . . . . .	108
6.6.2	Typedef Documentation . . . . .	108

6.6.2.1	OSCL_FSSTAT . . . . .	108
6.6.2.2	OSCL_STAT_BUF . . . . .	108
6.6.2.3	TOsclFileHandle . . . . .	108
6.6.3	Enumeration Type Documentation . . . . .	108
6.6.3.1	OSCL_FILEMGMT_ERR_TYPE . . . . .	108
6.6.3.2	OSCL_FILEMGMT_MODES . . . . .	108
6.6.3.3	OSCL_FILEMGMT_PERMS . . . . .	109
6.6.3.4	TOsclFileOp . . . . .	109
6.6.3.5	TPVDNSEvent . . . . .	109
6.6.3.6	TPVDNSFxn . . . . .	110
6.6.4	Function Documentation . . . . .	110
6.6.4.1	Accept . . . . .	110
6.6.4.2	Bind . . . . .	110
6.6.4.3	Bind . . . . .	110
6.6.4.4	BindAsync . . . . .	111
6.6.4.5	BindAsync . . . . .	111
6.6.4.6	CancelAccept . . . . .	111
6.6.4.7	CancelBind . . . . .	111
6.6.4.8	CancelBind . . . . .	111
6.6.4.9	CancelConnect . . . . .	112
6.6.4.10	CancelGetHostByName . . . . .	112
6.6.4.11	CancelListen . . . . .	112
6.6.4.12	CancelRecv . . . . .	112
6.6.4.13	CancelRecvFrom . . . . .	112
6.6.4.14	CancelSend . . . . .	112
6.6.4.15	CancelSendTo . . . . .	113
6.6.4.16	CancelShutdown . . . . .	113
6.6.4.17	Close . . . . .	113
6.6.4.18	Close . . . . .	113
6.6.4.19	Close . . . . .	113
6.6.4.20	Connect . . . . .	113
6.6.4.21	Connect . . . . .	114
6.6.4.22	GetAcceptedSocketL . . . . .	114
6.6.4.23	GetHostByName . . . . .	114
6.6.4.24	GetPeerName . . . . .	115
6.6.4.25	GetPeerName . . . . .	115

6.6.4.26	GetRecvData . . . . .	115
6.6.4.27	GetRecvData . . . . .	116
6.6.4.28	GetSendData . . . . .	116
6.6.4.29	GetSendData . . . . .	116
6.6.4.30	Join . . . . .	116
6.6.4.31	JoinMulticastGroup . . . . .	117
6.6.4.32	Listen . . . . .	117
6.6.4.33	ListenAsync . . . . .	117
6.6.4.34	oscl_chdir . . . . .	117
6.6.4.35	oscl_chdir . . . . .	118
6.6.4.36	oscl_getcwd . . . . .	118
6.6.4.37	oscl_get cwd . . . . .	118
6.6.4.38	oscl_mkdir . . . . .	118
6.6.4.39	oscl_mkdir . . . . .	119
6.6.4.40	oscl_rename . . . . .	119
6.6.4.41	oscl_rename . . . . .	119
6.6.4.42	oscl_rmdir . . . . .	119
6.6.4.43	oscl_rmdir . . . . .	120
6.6.4.44	oscl_stat . . . . .	120
6.6.4.45	oscl_stat . . . . .	120
6.6.4.46	oscl_stats . . . . .	120
6.6.4.47	oscl_stats . . . . .	121
6.6.4.48	OsclExtractFilenameFromFullPath . . . . .	121
6.6.4.49	OsclExtractFilenameFromFullPath . . . . .	121
6.6.4.50	OsclGetFileAttributes . . . . .	121
6.6.4.51	OsclGetFileAttributes . . . . .	122
6.6.4.52	OsclGetFileCreationTime . . . . .	122
6.6.4.53	OsclGetFileCreationTime . . . . .	122
6.6.4.54	OsclGetFileLastAccessTime . . . . .	123
6.6.4.55	OsclGetFileLastAccessTime . . . . .	123
6.6.4.56	OsclGetFileLastWriteTime . . . . .	123
6.6.4.57	OsclGetFileLastWriteTime . . . . .	124
6.6.4.58	OsclGetFileSize . . . . .	124
6.6.4.59	OsclGetFileSize . . . . .	124
6.6.4.60	Recv . . . . .	124
6.6.4.61	RecvFrom . . . . .	125

6.6.4.62	Send . . . . .	125
6.6.4.63	SendTo . . . . .	126
6.6.4.64	SetMulticastTTL . . . . .	126
6.6.4.65	SetOptionToReuseAddress . . . . .	126
6.6.4.66	SetOptionToReuseAddress . . . . .	126
6.6.4.67	SetRecvBufferSize . . . . .	127
6.6.4.68	SetTOS . . . . .	127
6.6.4.69	SetTOS . . . . .	127
6.6.4.70	Shutdown . . . . .	128
6.6.4.71	ThreadLogoff . . . . .	128
6.6.4.72	ThreadLogoff . . . . .	128
6.6.4.73	ThreadLogon . . . . .	128
6.6.4.74	ThreadLogon . . . . .	128
6.6.4.75	~OsclDNS . . . . .	128
6.6.4.76	~OsclDNSObserver . . . . .	129
6.6.4.77	~OsclSocketServ . . . . .	129
6.6.4.78	~OsclTCPSocket . . . . .	129
6.6.4.79	~OsclUDPSocket . . . . .	129
6.6.5	Friends . . . . .	129
6.6.5.1	OsclDNS . . . . .	129
6.6.5.2	OsclDNSRequestAO . . . . .	129
6.6.5.3	OsclTCPSocket . . . . .	129
6.6.5.4	OsclUDPSocket . . . . .	129
6.7	OSCL Proc . . . . .	130
6.7.1	Define Documentation . . . . .	132
6.7.1.1	OSCL_PERF_SUMMARY_LOGGING . . . . .	132
6.7.1.2	OSCL_ZEROIZE . . . . .	132
6.7.1.3	PV_SCHED_CHECK_Q . . . . .	132
6.7.1.4	PV_SCHED_ENABLE_LOOP_STATS . . . . .	132
6.7.1.5	PV_SCHED_ENABLE_PERF_LOGGING . . . . .	132
6.7.1.6	PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS . . . . .	132
6.7.1.7	PV_SCHED_FAIR_SCHEDULING . . . . .	132
6.7.1.8	PV_SCHED_LOG_Q . . . . .	132
6.7.1.9	PVEXECNAMELEN . . . . .	132
6.7.1.10	PVSCHEDNAMELEN . . . . .	132
6.7.1.11	QUE_ITER_BEGIN . . . . .	132

6.7.1.12	QUE_ITER_END . . . . .	132
6.7.2	Typedef Documentation . . . . .	133
6.7.2.1	TOsclReady . . . . .	133
6.7.3	Enumeration Type Documentation . . . . .	133
6.7.3.1	TPVThreadContext . . . . .	133
6.7.4	Function Documentation . . . . .	133
6.7.4.1	OsclPtrAdd . . . . .	133
6.7.4.2	OsclPtrSub . . . . .	133
6.7.5	Variable Documentation . . . . .	133
6.7.5.1	OSCL_REQUEST_ERR_CANCEL . . . . .	133
6.7.5.2	OSCL_REQUEST_ERR_GENERAL . . . . .	133
6.7.5.3	OSCL_REQUEST_ERR_NONE . . . . .	133
6.7.5.4	OSCL_REQUEST_PENDING . . . . .	133
6.8	OSCL Init . . . . .	134
<b>7</b>	<b>Data Structure Documentation</b>	<b>135</b>
7.1	_OsclBasicAllocator Class Reference . . . . .	135
7.1.1	Detailed Description . . . . .	135
7.1.2	Constructor & Destructor Documentation . . . . .	136
7.1.2.1	~_OsclBasicAllocator . . . . .	136
7.1.3	Member Function Documentation . . . . .	136
7.1.3.1	allocate . . . . .	136
7.1.3.2	deallocate . . . . .	136
7.2	_OsclHeapBase Class Reference . . . . .	137
7.2.1	Detailed Description . . . . .	138
7.2.2	Constructor & Destructor Documentation . . . . .	138
7.2.2.1	~_OsclHeapBase . . . . .	138
7.2.2.2	_OsclHeapBase . . . . .	138
7.2.2.3	_OsclHeapBase . . . . .	138
7.2.3	Friends And Related Function Documentation . . . . .	138
7.2.3.1	PVCleanupStack . . . . .	138
7.3	AcceptParam Class Reference . . . . .	139
7.3.1	Constructor & Destructor Documentation . . . . .	139
7.3.1.1	AcceptParam . . . . .	139
7.3.2	Field Documentation . . . . .	139
7.3.2.1	iBlankSocket . . . . .	139
7.4	allocator Class Reference . . . . .	140

7.4.1	Detailed Description . . . . .	140
7.5	AllPassFilter Class Reference . . . . .	141
7.5.1	Detailed Description . . . . .	141
7.5.2	Member Typedef Documentation . . . . .	141
7.5.2.1	filter_status_type . . . . .	141
7.5.2.2	log_level_type . . . . .	141
7.5.2.3	message_id_type . . . . .	141
7.5.3	Constructor & Destructor Documentation . . . . .	142
7.5.3.1	AllPassFilter . . . . .	142
7.5.3.2	~AllPassFilter . . . . .	142
7.5.4	Member Function Documentation . . . . .	142
7.5.4.1	FilterOpaqueMessge . . . . .	142
7.5.4.2	FilterString . . . . .	142
7.6	BindParam Class Reference . . . . .	143
7.6.1	Constructor & Destructor Documentation . . . . .	143
7.6.1.1	BindParam . . . . .	143
7.6.2	Field Documentation . . . . .	143
7.6.2.1	iAddr . . . . .	143
7.7	BufferFragment Class Reference . . . . .	144
7.8	BufferMgr Class Reference . . . . .	145
7.8.1	Constructor & Destructor Documentation . . . . .	145
7.8.1.1	~BufferMgr . . . . .	145
7.8.2	Member Function Documentation . . . . .	145
7.8.2.1	BufferReleased . . . . .	145
7.9	BufferState Class Reference . . . . .	146
7.9.1	Constructor & Destructor Documentation . . . . .	146
7.9.1.1	BufferState . . . . .	146
7.9.1.2	BufferState . . . . .	146
7.9.2	Member Function Documentation . . . . .	146
7.9.2.1	bind . . . . .	146
7.9.2.2	bind . . . . .	146
7.9.2.3	decrement_refcnt . . . . .	146
7.9.2.4	get_buf_mgr . . . . .	146
7.9.2.5	get_free_function . . . . .	146
7.9.2.6	get_ptr . . . . .	146
7.9.2.7	get_refcount . . . . .	146

7.9.2.8	increment_refcnt . . . . .	146
7.9.2.9	reset . . . . .	147
7.10	BufFragGroup< ChainClass, max_frags > Class Template Reference . . . . .	148
7.10.1	Constructor & Destructor Documentation . . . . .	148
7.10.1.1	BufFragGroup . . . . .	148
7.10.1.2	~BufFragGroup . . . . .	149
7.10.2	Member Function Documentation . . . . .	149
7.10.2.1	AddFragment . . . . .	149
7.10.2.2	AppendNext . . . . .	149
7.10.2.3	Clear . . . . .	149
7.10.2.4	GetLength . . . . .	149
7.10.2.5	GetMaxFrags . . . . .	149
7.10.2.6	GetNext . . . . .	149
7.10.2.7	GetNumFrags . . . . .	149
7.10.3	Field Documentation . . . . .	150
7.10.3.1	buffer_states . . . . .	150
7.10.3.2	fragments . . . . .	150
7.10.3.3	length . . . . .	150
7.10.3.4	next . . . . .	150
7.10.3.5	num_fragments . . . . .	150
7.11	BufFragStatusClass Class Reference . . . . .	151
7.11.1	Member Enumeration Documentation . . . . .	151
7.11.1.1	status_t . . . . .	151
7.12	CallbackTimer< Alloc > Class Template Reference . . . . .	152
7.12.1	Constructor & Destructor Documentation . . . . .	152
7.12.1.1	CallbackTimer . . . . .	152
7.12.1.2	~CallbackTimer . . . . .	152
7.12.2	Member Function Documentation . . . . .	152
7.12.2.1	Run . . . . .	152
7.13	CallbackTimerObserver Class Reference . . . . .	154
7.13.1	Constructor & Destructor Documentation . . . . .	154
7.13.1.1	~CallbackTimerObserver . . . . .	154
7.13.2	Member Function Documentation . . . . .	154
7.13.2.1	TimerBaseElapsed . . . . .	154
7.14	CFastRep Class Reference . . . . .	155
7.14.1	Detailed Description . . . . .	155

7.14.2	Constructor & Destructor Documentation . . . . .	156
7.14.2.1	CFastRep . . . . .	156
7.14.3	Member Function Documentation . . . . .	156
7.14.3.1	append . . . . .	156
7.14.3.2	append . . . . .	156
7.14.3.3	set_r . . . . .	156
7.14.3.4	set_r . . . . .	156
7.14.3.5	set_w . . . . .	156
7.14.3.6	set_w . . . . .	156
7.14.4	Field Documentation . . . . .	156
7.14.4.1	buffer . . . . .	156
7.14.4.2	maxsize . . . . .	156
7.14.4.3	overwrite . . . . .	156
7.14.4.4	size . . . . .	156
7.14.4.5	writable . . . . .	156
7.15	CHheapRep Class Reference . . . . .	157
7.15.1	Detailed Description . . . . .	157
7.15.2	Constructor & Destructor Documentation . . . . .	158
7.15.2.1	CHheapRep . . . . .	158
7.15.3	Member Function Documentation . . . . .	158
7.15.3.1	add_ref . . . . .	158
7.15.3.2	append . . . . .	158
7.15.3.3	append . . . . .	158
7.15.3.4	append_rep . . . . .	158
7.15.3.5	append_rep . . . . .	158
7.15.3.6	assign . . . . .	158
7.15.3.7	remove_ref . . . . .	158
7.15.3.8	set . . . . .	158
7.15.3.9	set . . . . .	158
7.15.3.10	set_rep . . . . .	158
7.15.3.11	set_rep . . . . .	158
7.15.4	Field Documentation . . . . .	158
7.15.4.1	buffer . . . . .	158
7.15.4.2	maxsize . . . . .	159
7.15.4.3	refcount . . . . .	159
7.15.4.4	size . . . . .	159

---

<b>7.16 ConnectParam Class Reference</b> . . . . .	160
<b>7.16.1 Constructor &amp; Destructor Documentation</b> . . . . .	160
<b>7.16.1.1 ConnectParam</b> . . . . .	160
<b>7.16.2 Field Documentation</b> . . . . .	160
<b>7.16.2.1 iAddr</b> . . . . .	160
<b>7.17 Oscl_TagTree&lt; T, Alloc &gt;::const_iterator Struct Reference</b> . . . . .	161
<b>7.17.1 Member Typedef Documentation</b> . . . . .	162
<b>7.17.1.1 mapiter</b> . . . . .	162
<b>7.17.1.2 pointer</b> . . . . .	162
<b>7.17.1.3 reference</b> . . . . .	162
<b>7.17.1.4 self</b> . . . . .	162
<b>7.17.2 Constructor &amp; Destructor Documentation</b> . . . . .	162
<b>7.17.2.1 const_iterator</b> . . . . .	162
<b>7.17.2.2 const_iterator</b> . . . . .	162
<b>7.17.2.3 const_iterator</b> . . . . .	162
<b>7.17.3 Member Function Documentation</b> . . . . .	162
<b>7.17.3.1 operator!=</b> . . . . .	162
<b>7.17.3.2 operator*</b> . . . . .	162
<b>7.17.3.3 operator++</b> . . . . .	162
<b>7.17.3.4 operator++</b> . . . . .	162
<b>7.17.3.5 operator--</b> . . . . .	163
<b>7.17.3.6 operator--</b> . . . . .	163
<b>7.17.3.7 operator-&gt;</b> . . . . .	163
<b>7.17.3.8 operator==</b> . . . . .	163
<b>7.17.4 Field Documentation</b> . . . . .	163
<b>7.17.4.1 mapit</b> . . . . .	163
<b>7.18 CStackRep Class Reference</b> . . . . .	164
<b>7.18.1 Detailed Description</b> . . . . .	164
<b>7.18.2 Constructor &amp; Destructor Documentation</b> . . . . .	164
<b>7.18.2.1 CStackRep</b> . . . . .	164
<b>7.18.3 Member Function Documentation</b> . . . . .	164
<b>7.18.3.1 append</b> . . . . .	164
<b>7.18.3.2 append</b> . . . . .	164
<b>7.18.3.3 set</b> . . . . .	164
<b>7.18.3.4 set</b> . . . . .	164
<b>7.18.4 Field Documentation</b> . . . . .	164

7.18.4.1	buffer . . . . .	164
7.18.4.2	maxsize . . . . .	165
7.18.4.3	size . . . . .	165
7.19	DNSRequestParam Class Reference . . . . .	166
7.19.1	Constructor & Destructor Documentation . . . . .	166
7.19.1.1	~DNSRequestParam . . . . .	166
7.19.1.2	DNSRequestParam . . . . .	166
7.19.2	Member Function Documentation . . . . .	166
7.19.2.1	Destroy . . . . .	166
7.19.2.2	RemoveRef . . . . .	167
7.19.3	Field Documentation . . . . .	167
7.19.3.1	iDNSRequest . . . . .	167
7.19.3.2	iFxn . . . . .	167
7.19.3.3	iRefCount . . . . .	167
7.20	GetHostByNameParam Class Reference . . . . .	168
7.20.1	Member Enumeration Documentation . . . . .	168
7.20.1.1	"@0 . . . . .	168
7.20.2	Constructor & Destructor Documentation . . . . .	169
7.20.2.1	~GetHostByNameParam . . . . .	169
7.20.3	Member Function Documentation . . . . .	169
7.20.3.1	canPersistMoreHostAddresses . . . . .	169
7.20.3.2	Create . . . . .	169
7.20.3.3	Destroy . . . . .	169
7.20.3.4	PersistHostAddress . . . . .	169
7.20.4	Field Documentation . . . . .	169
7.20.4.1	iAddr . . . . .	169
7.20.4.2	iAddressList . . . . .	169
7.20.4.3	iName . . . . .	169
7.21	HeapBase Class Reference . . . . .	170
7.21.1	Detailed Description . . . . .	171
7.21.2	Constructor & Destructor Documentation . . . . .	171
7.21.2.1	HeapBase . . . . .	171
7.21.2.2	~HeapBase . . . . .	171
7.21.3	Member Function Documentation . . . . .	171
7.21.3.1	operator delete . . . . .	171
7.21.3.2	operator delete[] . . . . .	171

---

7.21.3.3 operator new . . . . .	171
7.21.3.4 operator new . . . . .	171
7.21.3.5 operator new[] . . . . .	171
7.21.3.6 operator new[] . . . . .	171
7.22 internalLeave Class Reference . . . . .	172
7.22.1 Field Documentation . . . . .	172
7.22.1.1 a . . . . .	172
7.23 Oscl_TagTree< T, Alloc >::iterator Struct Reference . . . . .	173
7.23.1 Member Typedef Documentation . . . . .	174
7.23.1.1 mapiter . . . . .	174
7.23.1.2 pointer . . . . .	174
7.23.1.3 reference . . . . .	174
7.23.1.4 self . . . . .	174
7.23.2 Constructor & Destructor Documentation . . . . .	174
7.23.2.1 iterator . . . . .	174
7.23.2.2 iterator . . . . .	174
7.23.2.3 iterator . . . . .	174
7.23.3 Member Function Documentation . . . . .	174
7.23.3.1 operator!= . . . . .	174
7.23.3.2 operator* . . . . .	174
7.23.3.3 operator++ . . . . .	174
7.23.3.4 operator++ . . . . .	174
7.23.3.5 operator-- . . . . .	175
7.23.3.6 operator-- . . . . .	175
7.23.3.7 operator-> . . . . .	175
7.23.3.8 operator== . . . . .	175
7.23.4 Field Documentation . . . . .	175
7.23.4.1 mapit . . . . .	175
7.24 LinkedListElement< LLClass > Class Template Reference . . . . .	176
7.24.1 Detailed Description . . . . .	176
7.24.2 Constructor & Destructor Documentation . . . . .	176
7.24.2.1 LinkedListElement . . . . .	176
7.24.3 Field Documentation . . . . .	176
7.24.3.1 data . . . . .	176
7.24.3.2 next . . . . .	176
7.25 ListenParam Class Reference . . . . .	177

7.25.1	Constructor & Destructor Documentation . . . . .	177
7.25.1.1	ListenParam . . . . .	177
7.25.2	Field Documentation . . . . .	177
7.25.2.1	iQSize . . . . .	177
7.26	MediaData< ChainClass, max_frags, local_bufsize > Class Template Reference . . . . .	178
7.26.1	Constructor & Destructor Documentation . . . . .	179
7.26.1.1	MediaData . . . . .	179
7.26.1.2	~MediaData . . . . .	179
7.26.2	Member Function Documentation . . . . .	179
7.26.2.1	AddLocalFragment . . . . .	179
7.26.2.2	Clear . . . . .	179
7.26.2.3	GetAvailableBufferSize . . . . .	179
7.26.2.4	GetLocalBufsize . . . . .	179
7.26.2.5	GetLocalFragment . . . . .	179
7.26.2.6	GetMediaFragment . . . . .	180
7.26.2.7	GetMediaSize . . . . .	180
7.26.2.8	GetNumMediaFrags . . . . .	180
7.26.2.9	GetTimestamp . . . . .	180
7.26.2.10	IsLocalData . . . . .	180
7.26.2.11	SetTimestamp . . . . .	180
7.26.3	Field Documentation . . . . .	181
7.26.3.1	available_localbuf . . . . .	181
7.26.3.2	localbuf . . . . .	181
7.26.3.3	num_reserved_fragments . . . . .	181
7.26.3.4	timestamp . . . . .	181
7.27	MediaStatusClass Class Reference . . . . .	182
7.28	MemAllocator< T > Class Template Reference . . . . .	183
7.28.1	Member Typedef Documentation . . . . .	183
7.28.1.1	pointer . . . . .	183
7.28.2	Constructor & Destructor Documentation . . . . .	183
7.28.2.1	~MemAllocator . . . . .	183
7.28.3	Member Function Documentation . . . . .	183
7.28.3.1	allocate . . . . .	183
7.28.3.2	deallocate . . . . .	183
7.29	OsclMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference . . . . .	184
7.29.1	Field Documentation . . . . .	184

---

7.29.1.1	iBlockBuffer . . . . .	184
7.29.1.2	iBlockPostFence . . . . .	184
7.29.1.3	iBlockPreFence . . . . .	184
7.29.1.4	iBlockSize . . . . .	184
7.29.1.5	iNextFreeBlock . . . . .	184
7.29.1.6	iParentBuffer . . . . .	184
7.29.1.7	iPrevFreeBlock . . . . .	184
7.30	OsclMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference . . . . .	185
7.30.1	Field Documentation . . . . .	185
7.30.1.1	iAllocatedSz . . . . .	185
7.30.1.2	iBufferPostFence . . . . .	185
7.30.1.3	iBufferPreFence . . . . .	185
7.30.1.4	iBufferSize . . . . .	185
7.30.1.5	iEndAddr . . . . .	185
7.30.1.6	iNextFreeBlock . . . . .	185
7.30.1.7	iNumOutstanding . . . . .	185
7.30.1.8	iPrevAllocBlock . . . . .	185
7.30.1.9	iStartAddr . . . . .	185
7.31	MM_AllocBlockFence Struct Reference . . . . .	186
7.31.1	Constructor & Destructor Documentation . . . . .	186
7.31.1.1	MM_AllocBlockFence . . . . .	186
7.31.2	Member Function Documentation . . . . .	186
7.31.2.1	check_fence . . . . .	186
7.31.2.2	fill_fence . . . . .	186
7.31.3	Field Documentation . . . . .	186
7.31.3.1	pad . . . . .	186
7.32	MM_AllocBlockHdr Struct Reference . . . . .	187
7.32.1	Constructor & Destructor Documentation . . . . .	187
7.32.1.1	MM_AllocBlockHdr . . . . .	187
7.32.1.2	MM_AllocBlockHdr . . . . .	187
7.32.2	Member Function Documentation . . . . .	187
7.32.2.1	isAllocNodePtr . . . . .	187
7.32.2.2	setAllocNodeFlag . . . . .	187
7.32.3	Field Documentation . . . . .	187
7.32.3.1	pad . . . . .	187
7.32.3.2	pNode . . . . .	187

7.32.3.3	pRootNode . . . . .	187
7.32.3.4	size . . . . .	187
7.33	MM_AllocInfo Struct Reference . . . . .	189
7.33.1	Constructor & Destructor Documentation . . . . .	189
7.33.1.1	MM_AllocInfo . . . . .	189
7.33.1.2	~MM_AllocInfo . . . . .	189
7.33.2	Member Function Documentation . . . . .	189
7.33.2.1	operator delete . . . . .	189
7.33.2.2	operator new . . . . .	189
7.33.2.3	operator new . . . . .	189
7.33.3	Field Documentation . . . . .	190
7.33.3.1	allocNum . . . . .	190
7.33.3.2	bSetFailure . . . . .	190
7.33.3.3	lineNo . . . . .	190
7.33.3.4	pFileName . . . . .	190
7.33.3.5	pMemBlock . . . . .	190
7.33.3.6	pStatsNode . . . . .	190
7.33.3.7	size . . . . .	190
7.34	MM_AllocNode Struct Reference . . . . .	191
7.34.1	Constructor & Destructor Documentation . . . . .	191
7.34.1.1	MM_AllocNode . . . . .	191
7.34.1.2	~MM_AllocNode . . . . .	191
7.34.2	Member Function Documentation . . . . .	191
7.34.2.1	operator delete . . . . .	191
7.34.2.2	operator new . . . . .	191
7.34.2.3	operator new . . . . .	191
7.34.3	Field Documentation . . . . .	192
7.34.3.1	pAllocInfo . . . . .	192
7.34.3.2	pNext . . . . .	192
7.34.3.3	pPrev . . . . .	192
7.35	MM_AllocQueryInfo Struct Reference . . . . .	193
7.35.1	Field Documentation . . . . .	193
7.35.1.1	allocNum . . . . .	193
7.35.1.2	fileName . . . . .	193
7.35.1.3	lineNo . . . . .	193
7.35.1.4	pMemBlock . . . . .	193

7.35.1.5	size . . . . .	193
7.35.1.6	tag . . . . .	193
7.36	MM_Audit_Imp Class Reference . . . . .	194
7.37	MM_AuditOverheadStats Struct Reference . . . . .	195
7.37.1	Field Documentation . . . . .	195
7.37.1.1	per_allocation_overhead . . . . .	195
7.37.1.2	stats_overhead . . . . .	195
7.38	MM_FailInsertParam Struct Reference . . . . .	196
7.38.1	Constructor & Destructor Documentation . . . . .	196
7.38.1.1	MM_FailInsertParam . . . . .	196
7.38.2	Member Function Documentation . . . . .	196
7.38.2.1	operator delete . . . . .	196
7.38.2.2	operator new . . . . .	196
7.38.2.3	operator new . . . . .	196
7.38.2.4	reset . . . . .	196
7.38.3	Field Documentation . . . . .	197
7.38.3.1	nAllocNum . . . . .	197
7.38.3.2	xsubi . . . . .	197
7.39	MM_Stats_CB Struct Reference . . . . .	198
7.39.1	Constructor & Destructor Documentation . . . . .	198
7.39.1.1	MM_Stats_CB . . . . .	198
7.39.2	Member Function Documentation . . . . .	198
7.39.2.1	operator delete . . . . .	198
7.39.2.2	operator new . . . . .	198
7.39.2.3	operator new . . . . .	198
7.39.3	Field Documentation . . . . .	198
7.39.3.1	num_child_nodes . . . . .	198
7.39.3.2	pStats . . . . .	198
7.39.3.3	tag . . . . .	198
7.40	MM_Stats_t Struct Reference . . . . .	200
7.40.1	Constructor & Destructor Documentation . . . . .	200
7.40.1.1	MM_Stats_t . . . . .	200
7.40.1.2	MM_Stats_t . . . . .	200
7.40.2	Member Function Documentation . . . . .	200
7.40.2.1	operator delete . . . . .	200
7.40.2.2	operator new . . . . .	200

7.40.2.3	operator new . . . . .	200
7.40.2.4	reset . . . . .	201
7.40.2.5	update . . . . .	201
7.40.3	Field Documentation . . . . .	201
7.40.3.1	numAllocFails . . . . .	201
7.40.3.2	numAllocs . . . . .	201
7.40.3.3	numBytes . . . . .	201
7.40.3.4	peakNumAllocs . . . . .	201
7.40.3.5	peakNumBytes . . . . .	201
7.40.3.6	totalNumAllocs . . . . .	201
7.40.3.7	totalNumBytes . . . . .	201
7.41	Oscl_TagTree< T, Alloc >::Node Struct Reference . . . . .	202
7.41.1	Member Typedef Documentation . . . . .	202
7.41.1.1	children_type . . . . .	202
7.41.2	Constructor & Destructor Documentation . . . . .	202
7.41.2.1	Node . . . . .	202
7.41.3	Member Function Documentation . . . . .	202
7.41.3.1	depth . . . . .	202
7.41.3.2	sort_children . . . . .	202
7.41.4	Field Documentation . . . . .	203
7.41.4.1	children . . . . .	203
7.41.4.2	parent . . . . .	203
7.41.4.3	tag . . . . .	203
7.41.4.4	value . . . . .	203
7.42	NTPTime Class Reference . . . . .	204
7.42.1	Detailed Description . . . . .	205
7.42.2	Constructor & Destructor Documentation . . . . .	205
7.42.2.1	NTPTime . . . . .	205
7.42.2.2	NTPTime . . . . .	205
7.42.2.3	NTPTime . . . . .	205
7.42.2.4	NTPTime . . . . .	205
7.42.2.5	NTPTime . . . . .	206
7.42.2.6	NTPTime . . . . .	206
7.42.3	Member Function Documentation . . . . .	206
7.42.3.1	get_lower32 . . . . .	206
7.42.3.2	get_middle32 . . . . .	206

7.42.3.3	get_upper32 . . . . .	206
7.42.3.4	get_value . . . . .	206
7.42.3.5	operator+= . . . . .	206
7.42.3.6	operator- . . . . .	206
7.42.3.7	operator= . . . . .	207
7.42.3.8	operator= . . . . .	207
7.42.3.9	set_from_system_time . . . . .	207
7.42.3.10	set_to_current_time . . . . .	207
7.42.3.11	to_system_time . . . . .	207
7.43	Oscl_Alloc Class Reference . . . . .	208
7.43.1	Constructor & Destructor Documentation . . . . .	208
7.43.1.1	~Oscl_Alloc . . . . .	208
7.43.2	Member Function Documentation . . . . .	208
7.43.2.1	allocate . . . . .	208
7.43.2.2	allocate_fl . . . . .	208
7.44	Oscl_Dealloc Class Reference . . . . .	209
7.44.1	Constructor & Destructor Documentation . . . . .	209
7.44.1.1	~Oscl_Dealloc . . . . .	209
7.44.2	Member Function Documentation . . . . .	209
7.44.2.1	deallocate . . . . .	209
7.45	Oscl_DefAlloc Class Reference . . . . .	210
7.45.1	Member Function Documentation . . . . .	210
7.45.1.1	allocate . . . . .	210
7.45.1.2	allocate_fl . . . . .	210
7.45.1.3	deallocate . . . . .	210
7.46	Oscl_DefAllocWithRefCounter< DefAlloc > Class Template Reference . . . . .	211
7.46.1	Detailed Description . . . . .	211
7.46.2	Member Function Documentation . . . . .	211
7.46.2.1	addRef . . . . .	211
7.46.2.2	Delete . . . . .	211
7.46.2.3	getCount . . . . .	212
7.46.2.4	New . . . . .	212
7.46.2.5	removeRef . . . . .	212
7.47	OSCL_FastString Class Reference . . . . .	213
7.47.1	Detailed Description . . . . .	213
7.47.2	Member Typedef Documentation . . . . .	214

7.47.2.1	chartype . . . . .	214
7.47.2.2	optype . . . . .	214
7.47.2.3	other_chartype . . . . .	214
7.47.3	Constructor & Destructor Documentation . . . . .	214
7.47.3.1	OSCL_FastString . . . . .	214
7.47.3.2	OSCL_FastString . . . . .	214
7.47.3.3	OSCL_FastString . . . . .	214
7.47.3.4	OSCL_FastString . . . . .	214
7.47.3.5	~OSCL_FastString . . . . .	215
7.47.4	Member Function Documentation . . . . .	215
7.47.4.1	get_cstr . . . . .	215
7.47.4.2	get_maxsize . . . . .	215
7.47.4.3	get_size . . . . .	215
7.47.4.4	get_str . . . . .	215
7.47.4.5	operator= . . . . .	215
7.47.4.6	operator= . . . . .	215
7.47.4.7	set . . . . .	216
7.47.4.8	set . . . . .	216
7.47.4.9	set_length . . . . .	216
7.47.5	Friends And Related Function Documentation . . . . .	216
7.47.5.1	OSCL_String . . . . .	216
7.48	Oscl_File Class Reference . . . . .	217
7.48.1	Member Enumeration Documentation . . . . .	218
7.48.1.1	mode_type . . . . .	218
7.48.1.2	seek_type . . . . .	218
7.48.1.3	TSymbianAccessMode . . . . .	219
7.48.2	Constructor & Destructor Documentation . . . . .	219
7.48.2.1	Oscl_File . . . . .	219
7.48.2.2	Oscl_File . . . . .	219
7.48.2.3	Oscl_File . . . . .	219
7.48.2.4	~Oscl_File . . . . .	219
7.48.3	Member Function Documentation . . . . .	219
7.48.3.1	AddFixedCache . . . . .	219
7.48.3.2	Close . . . . .	220
7.48.3.3	EndOfFile . . . . .	220
7.48.3.4	Flush . . . . .	220

7.48.3.5	GetError . . . . .	220
7.48.3.6	Handle . . . . .	220
7.48.3.7	Open . . . . .	221
7.48.3.8	Open . . . . .	221
7.48.3.9	Read . . . . .	221
7.48.3.10	RemoveFixedCache . . . . .	222
7.48.3.11	Seek . . . . .	222
7.48.3.12	SetAsyncReadBufferSize . . . . .	222
7.48.3.13	SetCacheObserver . . . . .	222
7.48.3.14	SetFileHandle . . . . .	222
7.48.3.15	SetLoggingEnable . . . . .	223
7.48.3.16	SetNativeAccessMode . . . . .	223
7.48.3.17	SetNativeBufferSize . . . . .	223
7.48.3.18	SetPVCacheSize . . . . .	223
7.48.3.19	SetSize . . . . .	224
7.48.3.20	SetSummaryStatsLoggingEnable . . . . .	224
7.48.3.21	Size . . . . .	224
7.48.3.22	Tell . . . . .	224
7.48.3.23	Write . . . . .	224
7.48.4	Friends And Related Function Documentation . . . . .	225
7.48.4.1	asyncfilereadcancel_test . . . . .	225
7.48.4.2	asyncfilereadwrite_test . . . . .	225
7.48.4.3	largeasyncfilereadwrite_test . . . . .	225
7.48.4.4	OsclFileCache . . . . .	225
7.48.4.5	OsclFileCacheBuffer . . . . .	225
7.49	Oscl_FileFind Class Reference . . . . .	226
7.49.1	Detailed Description . . . . .	226
7.49.2	Member Enumeration Documentation . . . . .	226
7.49.2.1	element_type . . . . .	226
7.49.2.2	error_type . . . . .	226
7.49.3	Constructor & Destructor Documentation . . . . .	227
7.49.3.1	Oscl_FileFind . . . . .	227
7.49.3.2	~Oscl_FileFind . . . . .	227
7.49.4	Member Function Documentation . . . . .	227
7.49.4.1	Close . . . . .	227
7.49.4.2	FindFirst . . . . .	227

7.49.4.3	FindFirst . . . . .	228
7.49.4.4	FindNext . . . . .	228
7.49.4.5	FindNext . . . . .	228
7.49.4.6	GetElementType . . . . .	229
7.49.4.7	GetLastError . . . . .	229
7.50	Oscl_FileServer Class Reference . . . . .	230
7.50.1	Constructor & Destructor Documentation . . . . .	230
7.50.1.1	Oscl_FileServer . . . . .	230
7.50.1.2	~Oscl_FileServer . . . . .	230
7.50.2	Member Function Documentation . . . . .	230
7.50.2.1	Close . . . . .	230
7.50.2.2	Connect . . . . .	230
7.50.2.3	Oscl_DeleteFile . . . . .	231
7.50.2.4	Oscl_DeleteFile . . . . .	231
7.50.3	Friends And Related Function Documentation . . . . .	231
7.50.3.1	Oscl_File . . . . .	231
7.50.3.2	OsclNativeFile . . . . .	231
7.51	oscl_fsstat Struct Reference . . . . .	232
7.51.1	Field Documentation . . . . .	232
7.51.1.1	freebytes . . . . .	232
7.51.1.2	totalbytes . . . . .	232
7.52	OSCL_HeapString< Alloc > Class Template Reference . . . . .	233
7.52.1	Detailed Description . . . . .	234
7.52.2	Member Typedef Documentation . . . . .	234
7.52.2.1	chartype . . . . .	234
7.52.2.2	otype . . . . .	234
7.52.2.3	other_chartype . . . . .	234
7.52.3	Friends And Related Function Documentation . . . . .	234
7.52.3.1	OSCL_String . . . . .	234
7.53	OSCL_HeapStringA Class Reference . . . . .	235
7.53.1	Detailed Description . . . . .	236
7.53.2	Member Typedef Documentation . . . . .	236
7.53.2.1	chartype . . . . .	236
7.53.2.2	otype . . . . .	236
7.53.2.3	other_chartype . . . . .	236
7.53.3	Constructor & Destructor Documentation . . . . .	236

---

7.53.3.1	OSCL_HeapStringA . . . . .	236
7.53.3.2	OSCL_HeapStringA . . . . .	236
7.53.3.3	OSCL_HeapStringA . . . . .	236
7.53.3.4	OSCL_HeapStringA . . . . .	237
7.53.3.5	OSCL_HeapStringA . . . . .	237
7.53.3.6	OSCL_HeapStringA . . . . .	237
7.53.3.7	OSCL_HeapStringA . . . . .	237
7.53.3.8	~OSCL_HeapStringA . . . . .	237
7.53.4	Member Function Documentation . . . . .	237
7.53.4.1	get_cstr . . . . .	237
7.53.4.2	get_maxsize . . . . .	237
7.53.4.3	get_size . . . . .	238
7.53.4.4	get_str . . . . .	238
7.53.4.5	operator= . . . . .	238
7.53.4.6	operator= . . . . .	238
7.53.4.7	operator= . . . . .	238
7.53.4.8	set . . . . .	238
7.53.4.9	set . . . . .	238
7.53.4.10	set . . . . .	239
7.53.5	Friends And Related Function Documentation . . . . .	239
7.53.5.1	OSCL_String . . . . .	239
7.54	Oscl_Int64_Utils Class Reference . . . . .	240
7.54.1	Detailed Description . . . . .	240
7.54.2	Member Function Documentation . . . . .	241
7.54.2.1	get_int64_lower32 . . . . .	241
7.54.2.2	get_int64_middle32 . . . . .	241
7.54.2.3	get_int64_upper32 . . . . .	241
7.54.2.4	get_uint64_lower32 . . . . .	241
7.54.2.5	get_uint64_middle32 . . . . .	241
7.54.2.6	get_uint64_upper32 . . . . .	241
7.54.2.7	set_int64 . . . . .	241
7.54.2.8	set_uint64 . . . . .	241
7.55	Oscl_Less< T > Struct Template Reference . . . . .	242
7.55.1	Member Function Documentation . . . . .	242
7.55.1.1	operator() . . . . .	242
7.56	Oscl_Linked_List< LLClass, Alloc > Class Template Reference . . . . .	243

7.56.1	Detailed Description	243
7.56.2	Constructor & Destructor Documentation	243
7.56.2.1	Oscl_Linked_List	243
7.56.2.2	~Oscl_Linked_List	244
7.56.3	Member Function Documentation	244
7.56.3.1	add_element	244
7.56.3.2	add_to_front	244
7.56.3.3	check_list	244
7.56.3.4	clear	244
7.56.3.5	dequeue_element	245
7.56.3.6	get_element	245
7.56.3.7	get_first	245
7.56.3.8	get_index	245
7.56.3.9	get_next	245
7.56.3.10	get_num_elements	246
7.56.3.11	insert_element	246
7.56.3.12	move_to_end	246
7.56.3.13	move_to_front	246
7.56.3.14	remove_element	247
7.56.3.15	remove_element	247
7.57	Oscl_Linked_List_Base Class Reference	248
7.57.1	Detailed Description	248
7.57.2	Constructor & Destructor Documentation	249
7.57.2.1	~Oscl_Linked_List_Base	249
7.57.3	Member Function Documentation	249
7.57.3.1	add_element	249
7.57.3.2	add_to_front	249
7.57.3.3	check_list	249
7.57.3.4	construct	249
7.57.3.5	destroy	250
7.57.3.6	get_element	250
7.57.3.7	get_first	250
7.57.3.8	get_index	250
7.57.3.9	get_next	250
7.57.3.10	insert_element	251
7.57.3.11	move_to_end	251

7.57.3.12 move_to_front . . . . .	251
7.57.3.13 remove_element . . . . .	251
7.57.3.14 remove_element . . . . .	252
7.57.4 Field Documentation . . . . .	252
7.57.4.1 head . . . . .	252
7.57.4.2 iterator . . . . .	252
7.57.4.3 num_elements . . . . .	252
7.57.4.4 sizeof_T . . . . .	252
7.57.4.5 tail . . . . .	252
7.58 Oscl_Map< Key, T, Alloc, Compare > Class Template Reference . . . . .	253
7.58.1 Detailed Description . . . . .	254
7.58.2 Member Typedef Documentation . . . . .	255
7.58.2.1 const_iterator . . . . .	255
7.58.2.2 const_reference . . . . .	255
7.58.2.3 iterator . . . . .	255
7.58.2.4 key_compare . . . . .	255
7.58.2.5 key_type . . . . .	255
7.58.2.6 pair_citerator_citerator . . . . .	255
7.58.2.7 pair_iterator_bool . . . . .	255
7.58.2.8 pair_iterator_iterator . . . . .	255
7.58.2.9 pointer . . . . .	255
7.58.2.10 reference . . . . .	255
7.58.2.11 self . . . . .	255
7.58.2.12 size_type . . . . .	255
7.58.2.13 value_type . . . . .	255
7.58.3 Constructor & Destructor Documentation . . . . .	255
7.58.3.1 Oscl_Map . . . . .	255
7.58.3.2 Oscl_Map . . . . .	256
7.58.4 Member Function Documentation . . . . .	256
7.58.4.1 begin . . . . .	256
7.58.4.2 begin . . . . .	256
7.58.4.3 clear . . . . .	256
7.58.4.4 count . . . . .	256
7.58.4.5 empty . . . . .	256
7.58.4.6 end . . . . .	256
7.58.4.7 end . . . . .	256

7.58.4.8 equal_range . . . . .	257
7.58.4.9 equal_range . . . . .	257
7.58.4.10 erase . . . . .	257
7.58.4.11 erase . . . . .	257
7.58.4.12 erase . . . . .	257
7.58.4.13 find . . . . .	257
7.58.4.14 find . . . . .	257
7.58.4.15 insert . . . . .	258
7.58.4.16 insert . . . . .	258
7.58.4.17 insert . . . . .	258
7.58.4.18 key_comp . . . . .	258
7.58.4.19 lower_bound . . . . .	258
7.58.4.20 lower_bound . . . . .	258
7.58.4.21 max_size . . . . .	258
7.58.4.22 operator= . . . . .	258
7.58.4.23 operator[] . . . . .	259
7.58.4.24 size . . . . .	259
7.58.4.25 upper_bound . . . . .	259
7.58.4.26 upper_bound . . . . .	259
7.58.4.27 value_comp . . . . .	259
7.59 Oscl_MTLinked_List< LLClass, Alloc, TheLock > Class Template Reference . . . . .	260
7.59.1 Detailed Description . . . . .	260
7.59.2 Constructor & Destructor Documentation . . . . .	260
7.59.2.1 Oscl_MTLinked_List . . . . .	260
7.59.2.2 ~Oscl_MTLinked_List . . . . .	260
7.59.3 Member Function Documentation . . . . .	261
7.59.3.1 add_element . . . . .	261
7.59.3.2 add_to_front . . . . .	261
7.59.3.3 dequeue_element . . . . .	261
7.59.3.4 get_element . . . . .	261
7.59.3.5 get_index . . . . .	262
7.59.3.6 move_to_end . . . . .	262
7.59.3.7 move_to_front . . . . .	262
7.59.3.8 remove_element . . . . .	262
7.59.3.9 remove_element . . . . .	263
7.59.4 Field Documentation . . . . .	263

---

7.59.4.1	the_list . . . . .	263
7.60	Oscl_Opaque_Type_Alloc Class Reference . . . . .	264
7.60.1	Detailed Description . . . . .	264
7.60.2	Constructor & Destructor Documentation . . . . .	264
7.60.2.1	~Oscl_Opaque_Type_Alloc . . . . .	264
7.60.3	Member Function Documentation . . . . .	264
7.60.3.1	allocate . . . . .	264
7.60.3.2	construct . . . . .	264
7.60.3.3	deallocate . . . . .	264
7.60.3.4	destroy . . . . .	264
7.61	Oscl_Opaque_Type_Alloc_LL Class Reference . . . . .	266
7.61.1	Detailed Description . . . . .	266
7.61.2	Constructor & Destructor Documentation . . . . .	266
7.61.2.1	~Oscl_Opaque_Type_Alloc_LL . . . . .	266
7.61.3	Member Function Documentation . . . . .	266
7.61.3.1	allocate . . . . .	266
7.61.3.2	compare_data . . . . .	266
7.61.3.3	construct . . . . .	266
7.61.3.4	deallocate . . . . .	267
7.61.3.5	destroy . . . . .	267
7.61.3.6	get_data . . . . .	267
7.61.3.7	get_next . . . . .	267
7.61.3.8	set_next . . . . .	267
7.62	Oscl_Opaque_Type_Compare Class Reference . . . . .	268
7.62.1	Detailed Description . . . . .	268
7.62.2	Constructor & Destructor Documentation . . . . .	268
7.62.2.1	~Oscl_Opaque_Type_Compare . . . . .	268
7.62.3	Member Function Documentation . . . . .	268
7.62.3.1	compare_EQ . . . . .	268
7.62.3.2	compare_LT . . . . .	268
7.62.3.3	swap . . . . .	269
7.63	Oscl_Pair< T1, T2 > Struct Template Reference . . . . .	270
7.63.1	Constructor & Destructor Documentation . . . . .	270
7.63.1.1	Oscl_Pair . . . . .	270
7.63.1.2	Oscl_Pair . . . . .	270
7.63.2	Field Documentation . . . . .	270

7.63.2.1	first . . . . .	270
7.63.2.2	second . . . . .	270
7.64	Oscl_Queue< T, Alloc > Class Template Reference . . . . .	271
7.64.1	Detailed Description . . . . .	271
7.64.2	Member Typedef Documentation . . . . .	272
7.64.2.1	const_reference . . . . .	272
7.64.2.2	pointer . . . . .	272
7.64.2.3	reference . . . . .	272
7.64.2.4	size_type . . . . .	272
7.64.2.5	value_type . . . . .	272
7.64.3	Constructor & Destructor Documentation . . . . .	272
7.64.3.1	Oscl_Queue . . . . .	272
7.64.3.2	Oscl_Queue . . . . .	272
7.64.3.3	~Oscl_Queue . . . . .	272
7.64.4	Member Function Documentation . . . . .	272
7.64.4.1	back . . . . .	272
7.64.4.2	back . . . . .	273
7.64.4.3	clear . . . . .	273
7.64.4.4	front . . . . .	273
7.64.4.5	front . . . . .	273
7.64.4.6	pop . . . . .	273
7.64.4.7	push . . . . .	273
7.65	Oscl_Queue_Base Class Reference . . . . .	274
7.65.1	Detailed Description . . . . .	274
7.65.2	Constructor & Destructor Documentation . . . . .	274
7.65.2.1	~Oscl_Queue_Base . . . . .	274
7.65.3	Member Function Documentation . . . . .	275
7.65.3.1	capacity . . . . .	275
7.65.3.2	clear . . . . .	275
7.65.3.3	construct . . . . .	275
7.65.3.4	construct . . . . .	275
7.65.3.5	destroy . . . . .	275
7.65.3.6	empty . . . . .	275
7.65.3.7	pop . . . . .	275
7.65.3.8	push . . . . .	275
7.65.3.9	reserve . . . . .	276

7.65.3.10 <code>size</code>	276
7.65.4 Field Documentation	276
7.65.4.1 <code>bufsize</code>	276
7.65.4.2 <code>elems</code>	276
7.65.4.3 <code>ifront</code>	276
7.65.4.4 <code>irear</code>	276
7.65.4.5 <code>numelems</code>	276
7.65.4.6 <code>sizeof_T</code>	276
7.66 <code>Oscl_Rb_Tree&lt; Key, Value, KeyOfValue, Compare, Alloc &gt;</code> Class Template Reference	277
7.66.1 Member Typedef Documentation	279
7.66.1.1 <code>const_iterator</code>	279
7.66.1.2 <code>const_pointer</code>	279
7.66.1.3 <code>const_reference</code>	279
7.66.1.4 <code>difference_type</code>	279
7.66.1.5 <code>iterator</code>	279
7.66.1.6 <code>key_type</code>	279
7.66.1.7 <code>link_type</code>	279
7.66.1.8 <code>pointer</code>	279
7.66.1.9 <code>reference</code>	279
7.66.1.10 <code>size_type</code>	279
7.66.1.11 <code>value_type</code>	279
7.66.2 Constructor & Destructor Documentation	279
7.66.2.1 <code>Oscl_Rb_Tree</code>	279
7.66.2.2 <code>Oscl_Rb_Tree</code>	279
7.66.2.3 <code>~Oscl_Rb_Tree</code>	279
7.66.3 Member Function Documentation	279
7.66.3.1 <code>begin</code>	279
7.66.3.2 <code>begin</code>	279
7.66.3.3 <code>clear</code>	280
7.66.3.4 <code>count</code>	280
7.66.3.5 <code>empty</code>	280
7.66.3.6 <code>end</code>	280
7.66.3.7 <code>end</code>	280
7.66.3.8 <code>equal_range</code>	280
7.66.3.9 <code>equal_range</code>	280
7.66.3.10 <code>erase</code>	281

7.66.3.11 <code>erase</code>	281
7.66.3.12 <code>erase</code>	281
7.66.3.13 <code>erase</code>	281
7.66.3.14 <code>find</code>	281
7.66.3.15 <code>find</code>	281
7.66.3.16 <code>insert_unique</code>	281
7.66.3.17 <code>insert_unique</code>	281
7.66.3.18 <code>insert_unique</code>	281
7.66.3.19 <code>insert_unique</code>	281
7.66.3.20 <code>lower_bound</code>	282
7.66.3.21 <code>lower_bound</code>	282
7.66.3.22 <code>max_size</code>	282
7.66.3.23 <code>operator=</code>	282
7.66.3.24 <code>size</code>	282
7.66.3.25 <code>upper_bound</code>	282
7.66.3.26 <code>upper_bound</code>	282
7.67 <code>Oscl_Rb_Tree_Base</code> Class Reference	283
7.67.1 Member Typedef Documentation	283
7.67.1.1 <code>base_link_type</code>	283
7.67.2 Member Function Documentation	283
7.67.2.1 <code>rebalance</code>	283
7.67.2.2 <code>rebalance_for_erase</code>	283
7.67.2.3 <code>rotate_left</code>	283
7.67.2.4 <code>rotate_right</code>	283
7.68 <code>Oscl_Rb_Tree_Const_Iterator&lt; Value &gt;</code> Struct Template Reference	284
7.68.1 Member Typedef Documentation	285
7.68.1.1 <code>base_link_type</code>	285
7.68.1.2 <code>const_iterator</code>	285
7.68.1.3 <code>link_type</code>	285
7.68.1.4 <code>pointer</code>	285
7.68.1.5 <code>reference</code>	285
7.68.1.6 <code>self</code>	285
7.68.1.7 <code>value_type</code>	285
7.68.2 Constructor & Destructor Documentation	285
7.68.2.1 <code>Oscl_Rb_Tree_Const_Iterator</code>	285
7.68.2.2 <code>Oscl_Rb_Tree_Const_Iterator</code>	285

---

7.68.2.3 Oscl_Rb_Tree_Const_Iterator . . . . .	285
7.68.3 Member Function Documentation . . . . .	285
7.68.3.1 operator!= . . . . .	285
7.68.3.2 operator* . . . . .	285
7.68.3.3 operator++ . . . . .	286
7.68.3.4 operator++ . . . . .	286
7.68.3.5 operator-- . . . . .	286
7.68.3.6 operator-- . . . . .	286
7.68.3.7 operator-> . . . . .	286
7.68.3.8 operator== . . . . .	286
7.68.4 Field Documentation . . . . .	286
7.68.4.1 node . . . . .	286
7.69 Oscl_Rb_Tree_Iterator< Value > Struct Template Reference . . . . .	287
7.69.1 Member Typedef Documentation . . . . .	288
7.69.1.1 base_link_type . . . . .	288
7.69.1.2 iterator . . . . .	288
7.69.1.3 link_type . . . . .	288
7.69.1.4 pointer . . . . .	288
7.69.1.5 reference . . . . .	288
7.69.1.6 self . . . . .	288
7.69.1.7 value_type . . . . .	288
7.69.2 Constructor & Destructor Documentation . . . . .	288
7.69.2.1 Oscl_Rb_Tree_Iterator . . . . .	288
7.69.2.2 Oscl_Rb_Tree_Iterator . . . . .	288
7.69.2.3 Oscl_Rb_Tree_Iterator . . . . .	288
7.69.3 Member Function Documentation . . . . .	288
7.69.3.1 operator!= . . . . .	288
7.69.3.2 operator* . . . . .	288
7.69.3.3 operator++ . . . . .	289
7.69.3.4 operator++ . . . . .	289
7.69.3.5 operator-- . . . . .	289
7.69.3.6 operator-- . . . . .	289
7.69.3.7 operator-> . . . . .	289
7.69.3.8 operator== . . . . .	289
7.69.4 Field Documentation . . . . .	289
7.69.4.1 node . . . . .	289

7.70 Oscl_Rb_Tree_Node< Value > Struct Template Reference . . . . .	290
7.70.1 Member Typedef Documentation . . . . .	290
7.70.1.1 link_type . . . . .	290
7.70.1.2 value_type . . . . .	290
7.70.2 Field Documentation . . . . .	290
7.70.2.1 value . . . . .	290
7.71 Oscl_Rb_Tree_Node_Base Struct Reference . . . . .	291
7.71.1 Member Typedef Documentation . . . . .	291
7.71.1.1 base_link_type . . . . .	291
7.71.1.2 color_type . . . . .	291
7.71.2 Member Enumeration Documentation . . . . .	291
7.71.2.1 RedBl . . . . .	291
7.71.3 Member Function Documentation . . . . .	292
7.71.3.1 maximum . . . . .	292
7.71.3.2 minimum . . . . .	292
7.71.4 Field Documentation . . . . .	292
7.71.4.1 color . . . . .	292
7.71.4.2 left . . . . .	292
7.71.4.3 parent . . . . .	292
7.71.4.4 right . . . . .	292
7.72 Oscl_Select1st< V, U > Struct Template Reference . . . . .	293
7.72.1 Member Function Documentation . . . . .	293
7.72.1.1 operator() . . . . .	293
7.73 OSCL_StackString< MaxBufSize > Class Template Reference . . . . .	294
7.73.1 Detailed Description . . . . .	295
7.73.2 Member Typedef Documentation . . . . .	295
7.73.2.1 chartype . . . . .	295
7.73.2.2 optype . . . . .	295
7.73.2.3 other_chartype . . . . .	295
7.73.3 Friends And Related Function Documentation . . . . .	295
7.73.3.1 OSCL_String . . . . .	295
7.74 oscl_stat_buf Struct Reference . . . . .	296
7.74.1 Field Documentation . . . . .	296
7.74.1.1 mode . . . . .	296
7.74.1.2 perms . . . . .	296
7.75 OSCL_String Class Reference . . . . .	297

---

7.75.1	Detailed Description	298
7.75.2	Member Typedef Documentation	298
7.75.2.1	chartype	298
7.75.3	Constructor & Destructor Documentation	298
7.75.3.1	OSCL_String	298
7.75.3.2	~OSCL_String	298
7.75.4	Member Function Documentation	298
7.75.4.1	append_rep	298
7.75.4.2	append_rep	298
7.75.4.3	get_cstr	298
7.75.4.4	get_maxsize	299
7.75.4.5	get_size	299
7.75.4.6	get_str	299
7.75.4.7	hash	299
7.75.4.8	is_writable	299
7.75.4.9	operator!=	299
7.75.4.10	operator+=	299
7.75.4.11	operator+=	299
7.75.4.12	operator+=	299
7.75.4.13	operator<	300
7.75.4.14	operator<=	300
7.75.4.15	operator=	300
7.75.4.16	operator=	300
7.75.4.17	operator==	300
7.75.4.18	operator==	300
7.75.4.19	operator>	300
7.75.4.20	operator>=	300
7.75.4.21	operator[]	300
7.75.4.22	read	300
7.75.4.23	set_len	300
7.75.4.24	set_rep	301
7.75.4.25	set_rep	301
7.75.4.26	setrep_to_char	301
7.75.4.27	write	301
7.75.4.28	write	301
7.76	Oscl_Tag< Alloc > Struct Template Reference	302

7.76.1	Constructor & Destructor Documentation . . . . .	302
7.76.1.1	Oscl_Tag . . . . .	302
7.76.1.2	Oscl_Tag . . . . .	302
7.76.1.3	~Oscl_Tag . . . . .	302
7.76.2	Member Function Documentation . . . . .	303
7.76.2.1	operator< . . . . .	303
7.76.3	Field Documentation . . . . .	303
7.76.3.1	tag . . . . .	303
7.76.3.2	tagAllocator . . . . .	303
7.77	Oscl_Tag_Base Struct Reference . . . . .	304
7.77.1	Member Typedef Documentation . . . . .	304
7.77.1.1	size_type . . . . .	304
7.77.1.2	tag_base_type . . . . .	304
7.77.1.3	tag_base_unit . . . . .	304
7.77.2	Member Function Documentation . . . . .	304
7.77.2.1	operator() . . . . .	304
7.77.2.2	tag_ancestor . . . . .	304
7.77.2.3	tag_cmp . . . . .	305
7.77.2.4	tag_copy . . . . .	305
7.77.2.5	tag_depth . . . . .	305
7.77.2.6	tag_len . . . . .	305
7.78	Oscl_TagTree< T, Alloc > Class Template Reference . . . . .	306
7.78.1	Detailed Description . . . . .	306
7.78.2	Member Typedef Documentation . . . . .	307
7.78.2.1	children_type . . . . .	307
7.78.2.2	map_type . . . . .	307
7.78.2.3	node_ptr . . . . .	307
7.78.2.4	node_type . . . . .	307
7.78.2.5	pair_iterator_bool . . . . .	307
7.78.2.6	size_type . . . . .	307
7.78.2.7	tag_base_type . . . . .	307
7.78.2.8	tag_type . . . . .	307
7.78.2.9	value_type . . . . .	307
7.78.3	Constructor & Destructor Documentation . . . . .	307
7.78.3.1	Oscl_TagTree . . . . .	307
7.78.3.2	Oscl_TagTree . . . . .	307

7.78.3.3	<code>~Oscl_TagTree</code>	307
7.78.4	Member Function Documentation	307
7.78.4.1	<code>begin</code>	307
7.78.4.2	<code>begin</code>	308
7.78.4.3	<code>clear</code>	308
7.78.4.4	<code>count</code>	308
7.78.4.5	<code>empty</code>	308
7.78.4.6	<code>end</code>	308
7.78.4.7	<code>end</code>	308
7.78.4.8	<code>erase</code>	308
7.78.4.9	<code>erase</code>	309
7.78.4.10	<code>find</code>	309
7.78.4.11	<code>insert</code>	309
7.78.4.12	<code>operator=</code>	309
7.78.4.13	<code>operator[]</code>	309
7.78.4.14	<code>size</code>	310
7.79	<code>Oscl_TAlloc&lt; T, Alloc &gt;</code> Class Template Reference	311
7.79.1	Member Typedef Documentation	312
7.79.1.1	<code>const_pointer</code>	312
7.79.1.2	<code>const_reference</code>	312
7.79.1.3	<code>pointer</code>	312
7.79.1.4	<code>reference</code>	312
7.79.1.5	<code>size_type</code>	312
7.79.1.6	<code>value_type</code>	312
7.79.2	Constructor & Destructor Documentation	312
7.79.2.1	<code>~Oscl_TAlloc</code>	312
7.79.3	Member Function Documentation	312
7.79.3.1	<code>address</code>	312
7.79.3.2	<code>address</code>	312
7.79.3.3	<code>alloc_and_construct</code>	312
7.79.3.4	<code>alloc_and_construct_fl</code>	312
7.79.3.5	<code>allocate</code>	312
7.79.3.6	<code>allocate_fl</code>	312
7.79.3.7	<code>construct</code>	312
7.79.3.8	<code>deallocate</code>	313
7.79.3.9	<code>deallocate</code>	313

7.79.3.10 <code>destroy</code> . . . . .	313
7.79.3.11 <code>destruct_and_dealloc</code> . . . . .	313
7.80 <code>Oscl_Vector&lt; T, Alloc &gt;</code> Class Template Reference . . . . .	314
7.80.1 Detailed Description . . . . .	315
7.80.2 Member Typedef Documentation . . . . .	315
7.80.2.1 <code>const_iterator</code> . . . . .	315
7.80.2.2 <code>const_reference</code> . . . . .	315
7.80.2.3 <code>iterator</code> . . . . .	315
7.80.2.4 <code>pointer</code> . . . . .	315
7.80.2.5 <code>reference</code> . . . . .	315
7.80.2.6 <code>value_type</code> . . . . .	315
7.80.3 Constructor & Destructor Documentation . . . . .	315
7.80.3.1 <code>Oscl_Vector</code> . . . . .	315
7.80.3.2 <code>Oscl_Vector</code> . . . . .	315
7.80.3.3 <code>Oscl_Vector</code> . . . . .	315
7.80.3.4 <code>~Oscl_Vector</code> . . . . .	316
7.80.4 Member Function Documentation . . . . .	316
7.80.4.1 <code>back</code> . . . . .	316
7.80.4.2 <code>back</code> . . . . .	316
7.80.4.3 <code>begin</code> . . . . .	316
7.80.4.4 <code>clear</code> . . . . .	316
7.80.4.5 <code>destroy</code> . . . . .	316
7.80.4.6 <code>end</code> . . . . .	316
7.80.4.7 <code>erase</code> . . . . .	317
7.80.4.8 <code>erase</code> . . . . .	317
7.80.4.9 <code>front</code> . . . . .	317
7.80.4.10 <code>front</code> . . . . .	317
7.80.4.11 <code>insert</code> . . . . .	317
7.80.4.12 <code>operator=</code> . . . . .	318
7.80.4.13 <code>operator[]</code> . . . . .	318
7.80.4.14 <code>operator[]</code> . . . . .	318
7.80.4.15 <code>pop_back</code> . . . . .	318
7.80.4.16 <code>push_back</code> . . . . .	318
7.80.4.17 <code>push_front</code> . . . . .	318
7.81 <code>Oscl_Vector_Base</code> Class Reference . . . . .	320
7.81.1 Detailed Description . . . . .	321

---

7.81.2	Constructor & Destructor Documentation . . . . .	321
7.81.2.1	~Oscl_Vector_Base . . . . .	321
7.81.3	Member Function Documentation . . . . .	321
7.81.3.1	assign_vector . . . . .	321
7.81.3.2	capacity . . . . .	321
7.81.3.3	construct . . . . .	321
7.81.3.4	construct . . . . .	321
7.81.3.5	construct . . . . .	321
7.81.3.6	destroy . . . . .	321
7.81.3.7	empty . . . . .	322
7.81.3.8	erase . . . . .	322
7.81.3.9	erase . . . . .	322
7.81.3.10	insert . . . . .	322
7.81.3.11	pop_back . . . . .	322
7.81.3.12	push_back . . . . .	323
7.81.3.13	push_front . . . . .	323
7.81.3.14	reserve . . . . .	323
7.81.3.15	size . . . . .	323
7.81.4	Friends And Related Function Documentation . . . . .	324
7.81.4.1	OsclPriorityQueueBase . . . . .	324
7.81.5	Field Documentation . . . . .	324
7.81.5.1	bufsize . . . . .	324
7.81.5.2	elems . . . . .	324
7.81.5.3	numelems . . . . .	324
7.81.5.4	sizeof_T . . . . .	324
7.82	OSCL_wFastString Class Reference . . . . .	325
7.82.1	Detailed Description . . . . .	325
7.82.2	Member Typedef Documentation . . . . .	325
7.82.2.1	chartype . . . . .	325
7.82.2.2	otype . . . . .	326
7.82.2.3	other_chartype . . . . .	326
7.82.3	Constructor & Destructor Documentation . . . . .	326
7.82.3.1	OSCL_wFastString . . . . .	326
7.82.3.2	OSCL_wFastString . . . . .	326
7.82.3.3	OSCL_wFastString . . . . .	326
7.82.3.4	OSCL_wFastString . . . . .	326

7.82.3.5 ~OSCL_wFastString . . . . .	326
7.82.4 Member Function Documentation . . . . .	326
7.82.4.1 get_cstr . . . . .	326
7.82.4.2 get_maxsize . . . . .	326
7.82.4.3 get_size . . . . .	326
7.82.4.4 get_str . . . . .	326
7.82.4.5 operator= . . . . .	326
7.82.4.6 operator= . . . . .	327
7.82.4.7 set . . . . .	327
7.82.4.8 set . . . . .	327
7.82.4.9 set_length . . . . .	327
7.82.5 Friends And Related Function Documentation . . . . .	327
7.82.5.1 OSCL_wString . . . . .	327
7.83 OSCL_wHeapString< Alloc > Class Template Reference . . . . .	328
7.83.1 Detailed Description . . . . .	328
7.83.2 Member Typedef Documentation . . . . .	329
7.83.2.1 chartype . . . . .	329
7.83.2.2 optype . . . . .	329
7.83.2.3 other_chartype . . . . .	329
7.83.3 Friends And Related Function Documentation . . . . .	329
7.83.3.1 OSCL_wString . . . . .	329
7.84 OSCL_wHeapStringA Class Reference . . . . .	330
7.84.1 Detailed Description . . . . .	331
7.84.2 Member Typedef Documentation . . . . .	331
7.84.2.1 chartype . . . . .	331
7.84.2.2 optype . . . . .	331
7.84.2.3 other_chartype . . . . .	331
7.84.3 Constructor & Destructor Documentation . . . . .	331
7.84.3.1 OSCL_wHeapStringA . . . . .	331
7.84.3.2 OSCL_wHeapStringA . . . . .	331
7.84.3.3 OSCL_wHeapStringA . . . . .	331
7.84.3.4 OSCL_wHeapStringA . . . . .	331
7.84.3.5 OSCL_wHeapStringA . . . . .	331
7.84.3.6 OSCL_wHeapStringA . . . . .	331
7.84.3.7 OSCL_wHeapStringA . . . . .	331
7.84.3.8 ~OSCL_wHeapStringA . . . . .	331

7.84.4 Member Function Documentation . . . . .	331
7.84.4.1 get_cstr . . . . .	331
7.84.4.2 get_maxsize . . . . .	331
7.84.4.3 get_size . . . . .	332
7.84.4.4 get_str . . . . .	332
7.84.4.5 operator= . . . . .	332
7.84.4.6 operator= . . . . .	332
7.84.4.7 operator= . . . . .	332
7.84.4.8 set . . . . .	332
7.84.4.9 set . . . . .	332
7.84.4.10 set . . . . .	332
7.84.5 Friends And Related Function Documentation . . . . .	332
7.84.5.1 OSCL_wString . . . . .	332
7.85 OSCL_wStackString< MaxBufSize > Class Template Reference . . . . .	333
7.85.1 Detailed Description . . . . .	333
7.85.2 Member Typedef Documentation . . . . .	334
7.85.2.1 chartype . . . . .	334
7.85.2.2 optype . . . . .	334
7.85.2.3 other_chartype . . . . .	334
7.85.3 Friends And Related Function Documentation . . . . .	334
7.85.3.1 OSCL_wString . . . . .	334
7.86 OSCL_wString Class Reference . . . . .	335
7.86.1 Detailed Description . . . . .	336
7.86.2 Member Typedef Documentation . . . . .	336
7.86.2.1 chartype . . . . .	336
7.86.3 Constructor & Destructor Documentation . . . . .	336
7.86.3.1 OSCL_wString . . . . .	336
7.86.3.2 ~OSCL_wString . . . . .	336
7.86.4 Member Function Documentation . . . . .	336
7.86.4.1 append_rep . . . . .	336
7.86.4.2 append_rep . . . . .	336
7.86.4.3 get_cstr . . . . .	336
7.86.4.4 get_maxsize . . . . .	336
7.86.4.5 get_size . . . . .	336
7.86.4.6 get_str . . . . .	336
7.86.4.7 hash . . . . .	337

7.86.4.8	is_writable	337
7.86.4.9	operator!=	337
7.86.4.10	operator+=	337
7.86.4.11	operator+=	337
7.86.4.12	operator+=	337
7.86.4.13	operator<	337
7.86.4.14	operator<=	337
7.86.4.15	operator=	337
7.86.4.16	operator=	337
7.86.4.17	operator==	337
7.86.4.18	operator==	337
7.86.4.19	operator>	337
7.86.4.20	operator>=	337
7.86.4.21	operator[]	337
7.86.4.22	read	337
7.86.4.23	set_len	337
7.86.4.24	set_rep	337
7.86.4.25	set_rep	337
7.86.4.26	setrep_to_wide_char	338
7.86.4.27	write	338
7.86.4.28	write	338
7.87	OsclAcceptMethod Class Reference	339
7.87.1	Constructor & Destructor Documentation	339
7.87.1.1	~OsclAcceptMethod	339
7.87.2	Member Function Documentation	339
7.87.2.1	Accept	339
7.87.2.2	AcceptRequest	339
7.87.2.3	DiscardAcceptedSocket	340
7.87.2.4	GetAcceptedSocket	340
7.87.2.5	NewL	340
7.88	OsclAcceptRequest Class Reference	341
7.88.1	Constructor & Destructor Documentation	341
7.88.1.1	OsclAcceptRequest	341
7.88.2	Member Function Documentation	341
7.88.2.1	Accept	341
7.89	OsclActiveObject Class Reference	342

---

7.89.1	Detailed Description . . . . .	343
7.89.2	Member Enumeration Documentation . . . . .	343
7.89.2.1	OsclActivePriority . . . . .	343
7.89.3	Constructor & Destructor Documentation . . . . .	343
7.89.3.1	OsclActiveObject . . . . .	343
7.89.3.2	~OsclActiveObject . . . . .	343
7.89.4	Member Function Documentation . . . . .	344
7.89.4.1	AddToScheduler . . . . .	344
7.89.4.2	Cancel . . . . .	344
7.89.4.3	DoCancel . . . . .	344
7.89.4.4	IsBusy . . . . .	344
7.89.4.5	PendComplete . . . . .	344
7.89.4.6	PendForExec . . . . .	344
7.89.4.7	Priority . . . . .	344
7.89.4.8	RemoveFromScheduler . . . . .	345
7.89.4.9	RunError . . . . .	345
7.89.4.10	RunIfNotReady . . . . .	345
7.89.4.11	SetBusy . . . . .	345
7.89.4.12	SetStatus . . . . .	345
7.89.4.13	Status . . . . .	345
7.89.4.14	StatusRef . . . . .	345
7.90	OsclAllocDestructDealloc Class Reference . . . . .	346
7.90.1	Constructor & Destructor Documentation . . . . .	346
7.90.1.1	~OsclAllocDestructDealloc . . . . .	346
7.91	OsclAOStatus Class Reference . . . . .	347
7.91.1	Constructor & Destructor Documentation . . . . .	347
7.91.1.1	OsclAOStatus . . . . .	347
7.91.1.2	OsclAOStatus . . . . .	347
7.91.2	Member Function Documentation . . . . .	347
7.91.2.1	operator!= . . . . .	347
7.91.2.2	operator< . . . . .	347
7.91.2.3	operator<= . . . . .	347
7.91.2.4	operator= . . . . .	347
7.91.2.5	operator== . . . . .	347
7.91.2.6	operator> . . . . .	347
7.91.2.7	operator>= . . . . .	347

7.91.2.8	Value	347
7.92	OsclAsyncFile Class Reference	348
7.92.1	Detailed Description	348
7.92.2	Constructor & Destructor Documentation	349
7.92.2.1	~OsclAsyncFile	349
7.92.3	Member Function Documentation	349
7.92.3.1	Close	349
7.92.3.2	Delete	349
7.92.3.3	EndOfFile	349
7.92.3.4	Flush	349
7.92.3.5	NewL	349
7.92.3.6	Open	349
7.92.3.7	Open	349
7.92.3.8	Read	349
7.92.3.9	Seek	349
7.92.3.10	Size	349
7.92.3.11	Tell	349
7.92.3.12	Write	349
7.92.4	Field Documentation	350
7.92.4.1	iNumOfRun	350
7.92.4.2	iNumOfRunErr	350
7.93	OsclAsyncFileBuffer Class Reference	351
7.93.1	Detailed Description	351
7.93.2	Constructor & Destructor Documentation	352
7.93.2.1	~OsclAsyncFileBuffer	352
7.93.3	Member Function Documentation	352
7.93.3.1	Buffer	352
7.93.3.2	CleanInUse	352
7.93.3.3	HasThisOffset	352
7.93.3.4	Id	352
7.93.3.5	IsInUse	352
7.93.3.6	IsValid	352
7.93.3.7	Length	352
7.93.3.8	NewL	352
7.93.3.9	Offset	352
7.93.3.10	SetInUse	352

7.93.3.11 SetOffset . . . . .	352
7.93.3.12 StartAsyncRead . . . . .	352
7.93.3.13 UpdateData . . . . .	352
7.94 OsclAuditCB Class Reference . . . . .	353
7.95 OsclBindMethod Class Reference . . . . .	354
7.95.1 Constructor & Destructor Documentation . . . . .	354
7.95.1.1 ~OsclBindMethod . . . . .	354
7.95.2 Member Function Documentation . . . . .	354
7.95.2.1 Bind . . . . .	354
7.95.2.2 BindRequest . . . . .	354
7.95.2.3 NewL . . . . .	354
7.96 OsclBindRequest Class Reference . . . . .	355
7.96.1 Detailed Description . . . . .	355
7.96.2 Constructor & Destructor Documentation . . . . .	355
7.96.2.1 OsclBindRequest . . . . .	355
7.96.3 Member Function Documentation . . . . .	355
7.96.3.1 Bind . . . . .	355
7.97 OsclBinIStream Class Reference . . . . .	356
7.97.1 Constructor & Destructor Documentation . . . . .	356
7.97.1.1 OsclBinIStream . . . . .	356
7.97.1.2 ~OsclBinIStream . . . . .	356
7.97.2 Member Function Documentation . . . . .	356
7.97.2.1 get . . . . .	356
7.97.2.2 Read_uint8 . . . . .	356
7.98 OsclBinIStreamBigEndian Class Reference . . . . .	358
7.98.1 Constructor & Destructor Documentation . . . . .	359
7.98.1.1 OsclBinIStreamBigEndian . . . . .	359
7.98.2 Member Function Documentation . . . . .	359
7.98.2.1 operator>> . . . . .	359
7.98.2.2 operator>> . . . . .	359
7.98.2.3 operator>> . . . . .	359
7.98.2.4 operator>> . . . . .	359
7.98.2.5 operator>> . . . . .	359
7.98.2.6 operator>> . . . . .	359
7.98.2.7 Read . . . . .	359
7.98.2.8 Read . . . . .	359

7.98.2.9 Read . . . . .	359
7.98.2.10 Read . . . . .	359
7.98.2.11 Read . . . . .	359
7.98.2.12 Read . . . . .	359
7.98.2.13 Read_uint16 . . . . .	359
7.98.2.14 Read_uint32 . . . . .	360
7.99 OsclBinIStreamLittleEndian Class Reference . . . . .	361
7.99.1 Constructor & Destructor Documentation . . . . .	362
7.99.1.1 OsclBinIStreamLittleEndian . . . . .	362
7.99.2 Member Function Documentation . . . . .	362
7.99.2.1 operator>> . . . . .	362
7.99.2.2 operator>> . . . . .	362
7.99.2.3 operator>> . . . . .	362
7.99.2.4 operator>> . . . . .	362
7.99.2.5 operator>> . . . . .	362
7.99.2.6 operator>> . . . . .	362
7.99.2.7 Read_uint16 . . . . .	362
7.99.2.8 Read_uint32 . . . . .	362
7.100 OsclBinOStream Class Reference . . . . .	363
7.100.1 Detailed Description . . . . .	363
7.100.2 Constructor & Destructor Documentation . . . . .	363
7.100.2.1 OsclBinOStream . . . . .	363
7.100.2.2 ~OsclBinOStream . . . . .	363
7.100.3 Member Function Documentation . . . . .	363
7.100.3.1 write . . . . .	363
7.101 OsclBinOStreamBigEndian Class Reference . . . . .	364
7.101.1 Detailed Description . . . . .	364
7.101.2 Constructor & Destructor Documentation . . . . .	365
7.101.2.1 OsclBinOStreamBigEndian . . . . .	365
7.101.3 Member Function Documentation . . . . .	365
7.101.3.1 operator<< . . . . .	365
7.101.3.2 operator<< . . . . .	365
7.101.3.3 operator<< . . . . .	365
7.101.3.4 operator<< . . . . .	365
7.101.3.5 operator<< . . . . .	365
7.101.3.6 operator<< . . . . .	365

---

7.101.3.7 WriteUnsignedLong . . . . .	365
7.101.3.8 WriteUnsignedShort . . . . .	365
7.102 OsclBinOStreamLittleEndian Class Reference . . . . .	366
7.102.1 Detailed Description . . . . .	366
7.102.2 Constructor & Destructor Documentation . . . . .	367
7.102.2.1 OsclBinOStreamLittleEndian . . . . .	367
7.102.3 Member Function Documentation . . . . .	367
7.102.3.1 operator<< . . . . .	367
7.102.3.2 operator<< . . . . .	367
7.102.3.3 operator<< . . . . .	367
7.102.3.4 operator<< . . . . .	367
7.102.3.5 operator<< . . . . .	367
7.102.3.6 operator<< . . . . .	367
7.102.3.7 WriteUnsignedLong . . . . .	367
7.102.3.8 WriteUnsignedShort . . . . .	367
7.103 OsclBinStream Class Reference . . . . .	368
7.103.1 Member Enumeration Documentation . . . . .	369
7.103.1.1 state_t . . . . .	369
7.103.2 Constructor & Destructor Documentation . . . . .	369
7.103.2.1 OsclBinStream . . . . .	369
7.103.3 Member Function Documentation . . . . .	369
7.103.3.1 Attach . . . . .	369
7.103.3.2 Attach . . . . .	369
7.103.3.3 eof . . . . .	370
7.103.3.4 fail . . . . .	370
7.103.3.5 good . . . . .	370
7.103.3.6 HaveRoomInCurrentBlock . . . . .	370
7.103.3.7 PositionInBlock . . . . .	370
7.103.3.8 ReserveSpace . . . . .	370
7.103.3.9 Seek . . . . .	370
7.103.3.10 seekFromCurrentPosition . . . . .	371
7.103.3.11 tellg . . . . .	371
7.103.4 Field Documentation . . . . .	371
7.103.4.1 firstFragPtr . . . . .	371
7.103.4.2 fragsLeft . . . . .	371
7.103.4.3 length . . . . .	371

7.103.4.4 nextFragPtr . . . . .	371
7.103.4.5 numFrags . . . . .	371
7.103.4.6 pBasePosition . . . . .	371
7.103.4.7 pPosition . . . . .	371
7.103.4.8 specialFragBuffer . . . . .	371
7.103.4.9 state . . . . .	371
7.104 OsclBuf Class Reference . . . . .	372
7.104.1 Constructor & Destructor Documentation . . . . .	372
7.104.1.1 OsclBuf . . . . .	372
7.104.2 Member Function Documentation . . . . .	372
7.104.2.1 Delete . . . . .	372
7.104.2.2 Des . . . . .	372
7.104.2.3 DesC . . . . .	373
7.104.2.4 Length . . . . .	373
7.104.2.5 NewL . . . . .	373
7.104.3 Field Documentation . . . . .	373
7.104.3.1 iBuffer . . . . .	373
7.104.3.2 iLength . . . . .	373
7.104.3.3 iMaxLength . . . . .	373
7.105 Oscl_File::OsclCacheObserver Class Reference . . . . .	374
7.105.1 Detailed Description . . . . .	374
7.105.2 Constructor & Destructor Documentation . . . . .	374
7.105.2.1 ~OsclCacheObserver . . . . .	374
7.105.3 Member Function Documentation . . . . .	374
7.105.3.1 ChooseCurCache . . . . .	374
7.106 OsclCompareLess< T > Class Template Reference . . . . .	375
7.106.1 Member Function Documentation . . . . .	375
7.106.1.1 compare . . . . .	375
7.107 OsclComponentRegistry Class Reference . . . . .	376
7.107.1 Detailed Description . . . . .	376
7.107.2 Constructor & Destructor Documentation . . . . .	377
7.107.2.1 OsclComponentRegistry . . . . .	377
7.107.2.2 ~OsclComponentRegistry . . . . .	377
7.107.3 Member Function Documentation . . . . .	377
7.107.3.1 CloseSession . . . . .	377
7.107.3.2 FindExact . . . . .	377

---

7.107.3.3 FindHierarchical . . . . .	377
7.107.3.4 OpenSession . . . . .	377
7.107.3.5 Register . . . . .	377
7.107.3.6 Unregister . . . . .	377
7.107.3.7 Unregister . . . . .	377
7.107.4 Field Documentation . . . . .	377
7.107.4.1 iComponentIdCounter . . . . .	377
7.107.4.2 iData . . . . .	377
7.107.4.3 iMutex . . . . .	377
7.107.4.4 iNumSessions . . . . .	377
7.108 OsclComponentRegistryData Class Reference . . . . .	378
7.108.1 Detailed Description . . . . .	378
7.108.2 Member Function Documentation . . . . .	378
7.108.2.1 Find . . . . .	378
7.108.3 Field Documentation . . . . .	378
7.108.3.1 iVec . . . . .	378
7.109 OsclComponentRegistryElement Class Reference . . . . .	379
7.109.1 Detailed Description . . . . .	379
7.109.2 Constructor & Destructor Documentation . . . . .	379
7.109.2.1 OsclComponentRegistryElement . . . . .	379
7.109.2.2 OsclComponentRegistryElement . . . . .	379
7.109.2.3 ~OsclComponentRegistryElement . . . . .	379
7.109.3 Member Function Documentation . . . . .	379
7.109.3.1 Match . . . . .	379
7.109.3.2 operator= . . . . .	379
7.109.4 Field Documentation . . . . .	379
7.109.4.1 iComponentId . . . . .	379
7.109.4.2 iFactory . . . . .	379
7.109.4.3 iId . . . . .	379
7.110 OsclConnectMethod Class Reference . . . . .	381
7.110.1 Constructor & Destructor Documentation . . . . .	381
7.110.1.1 ~OsclConnectMethod . . . . .	381
7.110.2 Member Function Documentation . . . . .	381
7.110.2.1 Connect . . . . .	381
7.110.2.2 ConnectRequest . . . . .	381
7.110.2.3 NewL . . . . .	381

7.111 OsclConnectRequest Class Reference . . . . .	383
7.111.1 Detailed Description . . . . .	383
7.111.2 Constructor & Destructor Documentation . . . . .	383
7.111.2.1 OsclConnectRequest . . . . .	383
7.111.3 Member Function Documentation . . . . .	383
7.111.3.1 Connect . . . . .	383
7.112 OsclDestructDealloc Class Reference . . . . .	384
7.112.1 Constructor & Destructor Documentation . . . . .	384
7.112.1.1 ~OsclDestructDealloc . . . . .	384
7.112.2 Member Function Documentation . . . . .	384
7.112.2.1 destruct_and_dealloc . . . . .	384
7.113 OsclDNS Class Reference . . . . .	385
7.113.1 Detailed Description . . . . .	385
7.113.2 Member Function Documentation . . . . .	385
7.113.2.1 NewL . . . . .	385
7.114 OsclDNSI Class Reference . . . . .	387
7.114.1 Detailed Description . . . . .	387
7.114.2 Constructor & Destructor Documentation . . . . .	387
7.114.2.1 ~OsclDNSI . . . . .	387
7.114.3 Member Function Documentation . . . . .	387
7.114.3.1 Close . . . . .	387
7.114.3.2 GetHostByName . . . . .	388
7.114.3.3 GetHostByNameResponseContainsAliasInfo . . . . .	388
7.114.3.4 GetHostByNameSuccess . . . . .	388
7.114.3.5 GetNextHost . . . . .	388
7.114.3.6 GetNextHostSuccess . . . . .	388
7.114.3.7 NewL . . . . .	388
7.114.3.8 Open . . . . .	388
7.114.4 Friends And Related Function Documentation . . . . .	388
7.114.4.1 DNSRequestParam . . . . .	388
7.114.4.2 OsclDNSRequest . . . . .	388
7.114.4.3 OsclGetHostByNameRequest . . . . .	388
7.115 OsclDNSIBase Class Reference . . . . .	389
7.115.1 Detailed Description . . . . .	389
7.115.2 Constructor & Destructor Documentation . . . . .	390
7.115.2.1 ~OsclDNSIBase . . . . .	390

---

7.115.2.2 OsclDNSIBase . . . . .	390
7.115.3 Member Function Documentation . . . . .	390
7.115.3.1 CancelFxn . . . . .	390
7.115.3.2 CancelGetHostName . . . . .	390
7.115.3.3 Close . . . . .	390
7.115.3.4 GetHostName . . . . .	390
7.115.3.5 GetHostNameResponseContainsAliasInfo . . . . .	390
7.115.3.6 GetHostNameSuccess . . . . .	390
7.115.3.7 GetNextHost . . . . .	390
7.115.3.8 GetNextHostSuccess . . . . .	390
7.115.3.9 IsReady . . . . .	390
7.115.3.10 Open . . . . .	390
7.115.4 Friends And Related Function Documentation . . . . .	391
7.115.4.1 OsclDNSRequest . . . . .	391
7.115.4.2 OsclGetHostNameRequest . . . . .	391
7.115.5 Field Documentation . . . . .	391
7.115.5.1 iAlloc . . . . .	391
7.115.5.2 iSocketServ . . . . .	391
7.116 OsclDNSMethod Class Reference . . . . .	392
7.116.1 Detailed Description . . . . .	392
7.116.2 Constructor & Destructor Documentation . . . . .	393
7.116.2.1 OsclDNSMethod . . . . .	393
7.116.3 Member Function Documentation . . . . .	393
7.116.3.1 Abort . . . . .	393
7.116.3.2 AbortAll . . . . .	393
7.116.3.3 CancelMethod . . . . .	393
7.116.3.4 ConstructL . . . . .	393
7.116.3.5 MethodDone . . . . .	393
7.116.3.6 Run . . . . .	393
7.116.3.7 StartMethod . . . . .	394
7.116.4 Field Documentation . . . . .	394
7.116.4.1 iAlloc . . . . .	394
7.116.4.2 iDNSFxn . . . . .	394
7.116.4.3 iDNSObserver . . . . .	394
7.116.4.4 iDNSRequestAO . . . . .	394
7.116.4.5 iId . . . . .	394

7.116.4.6 iLogger . . . . .	394
7.117 OsclDNSObserver Class Reference . . . . .	395
7.117.1 Detailed Description . . . . .	395
7.117.2 Member Function Documentation . . . . .	395
7.117.2.1 HandleDNSEvent . . . . .	395
7.118 OsclDNSRequestAO Class Reference . . . . .	396
7.118.1 Detailed Description . . . . .	397
7.118.2 Constructor & Destructor Documentation . . . . .	397
7.118.2.1 OsclDNSRequestAO . . . . .	397
7.118.3 Member Function Documentation . . . . .	397
7.118.3.1 Abort . . . . .	397
7.118.3.2 Cancelled . . . . .	397
7.118.3.3 ConstructL . . . . .	397
7.118.3.4 DoCancel . . . . .	397
7.118.3.5 Failure . . . . .	397
7.118.3.6 GetSocketError . . . . .	397
7.118.3.7 NewRequest . . . . .	397
7.118.3.8 RequestDone . . . . .	397
7.118.3.9 Run . . . . .	397
7.118.3.10 Serv . . . . .	398
7.118.3.11 Success . . . . .	398
7.118.4 Friends And Related Function Documentation . . . . .	398
7.118.4.1 GetHostByNameParam . . . . .	398
7.118.4.2 OsclDNSI . . . . .	398
7.118.4.3 OsclDNSMethod . . . . .	398
7.118.4.4 OsclDNSRequest . . . . .	398
7.118.5 Field Documentation . . . . .	398
7.118.5.1 iDNSI . . . . .	398
7.118.5.2 iDNSMethod . . . . .	398
7.118.5.3 iLogger . . . . .	398
7.118.5.4 iSocketError . . . . .	398
7.119 OsclDoubleLink Class Reference . . . . .	399
7.119.1 Constructor & Destructor Documentation . . . . .	399
7.119.1.1 OsclDoubleLink . . . . .	399
7.119.2 Member Function Documentation . . . . .	399
7.119.2.1 InsertAfter . . . . .	399

---

7.119.2.2 InsertBefore . . . . .	399
7.119.2.3 Remove . . . . .	399
7.119.3 Field Documentation . . . . .	399
7.119.3.1 iNext . . . . .	399
7.119.3.2 iPrev . . . . .	399
7.120 OsclDoubleList< T > Class Template Reference . . . . .	400
7.120.1 Constructor & Destructor Documentation . . . . .	400
7.120.1.1 OsclDoubleList . . . . .	400
7.120.1.2 OsclDoubleList . . . . .	400
7.120.2 Member Function Documentation . . . . .	400
7.120.2.1 Head . . . . .	400
7.120.2.2 InsertHead . . . . .	400
7.120.2.3 InsertTail . . . . .	400
7.120.2.4 IsHead . . . . .	400
7.120.2.5 IsTail . . . . .	400
7.120.2.6 Tail . . . . .	400
7.121 OsclDoubleListBase Class Reference . . . . .	401
7.121.1 Constructor & Destructor Documentation . . . . .	401
7.121.1.1 OsclDoubleListBase . . . . .	401
7.121.1.2 OsclDoubleListBase . . . . .	401
7.121.2 Member Function Documentation . . . . .	401
7.121.2.1 getHead . . . . .	401
7.121.2.2 getOffset . . . . .	402
7.121.2.3 Insert . . . . .	402
7.121.2.4 InsertHead . . . . .	402
7.121.2.5 InsertTail . . . . .	402
7.121.2.6 IsEmpty . . . . .	402
7.121.2.7 Reset . . . . .	402
7.121.2.8 SetOffset . . . . .	402
7.121.3 Field Documentation . . . . .	402
7.121.3.1 iHead . . . . .	402
7.121.3.2 iOffset . . . . .	402
7.122 OsclDoubleRunner< T > Class Template Reference . . . . .	403
7.122.1 Constructor & Destructor Documentation . . . . .	403
7.122.1.1 OsclDoubleRunner . . . . .	403
7.122.2 Member Function Documentation . . . . .	403

7.122.2.1 operator T * . . . . .	403
7.122.2.2 operator++ . . . . .	403
7.122.2.3 operator-- . . . . .	403
7.122.2.4 Set . . . . .	403
7.122.2.5 SetToHead . . . . .	404
7.122.2.6 SetToTail . . . . .	404
7.122.3 Field Documentation . . . . .	404
7.122.3.1 iHead . . . . .	404
7.122.3.2 iNext . . . . .	404
7.122.3.3 iOffset . . . . .	404
7.123 OsclError Class Reference . . . . .	405
7.123.1 Detailed Description . . . . .	405
7.123.2 Member Function Documentation . . . . .	405
7.123.2.1 Leave . . . . .	405
7.123.2.2 LeaveIfError . . . . .	405
7.123.2.3 LeaveIfNull . . . . .	405
7.123.2.4 Pop . . . . .	405
7.123.2.5 Pop . . . . .	406
7.123.2.6 PopDealloc . . . . .	406
7.123.2.7 PopDealloc . . . . .	406
7.123.2.8 PushL . . . . .	406
7.123.2.9 PushL . . . . .	406
7.123.2.10 PushL . . . . .	406
7.124 OsclErrorAllocator Class Reference . . . . .	407
7.124.1 Detailed Description . . . . .	407
7.124.2 Constructor & Destructor Documentation . . . . .	407
7.124.2.1 OsclErrorAllocator . . . . .	407
7.124.3 Member Function Documentation . . . . .	407
7.124.3.1 allocate . . . . .	407
7.124.3.2 deallocate . . . . .	408
7.124.3.3 operator delete . . . . .	408
7.124.3.4 operator new . . . . .	408
7.125 OsclErrorTrap Class Reference . . . . .	409
7.125.1 Member Function Documentation . . . . .	409
7.125.1.1 Cleanup . . . . .	409
7.125.1.2 GetErrorTrapImp . . . . .	409

---

7.125.1.3 Init . . . . .	409
7.126 OsclErrorTrapImp Class Reference . . . . .	410
7.126.1 Detailed Description . . . . .	410
7.126.2 Member Function Documentation . . . . .	410
7.126.2.1 Trap . . . . .	410
7.126.2.2 TrapNoTls . . . . .	410
7.126.2.3 UnTrap . . . . .	410
7.126.3 Friends And Related Function Documentation . . . . .	411
7.126.3.1 CPVInterfaceProxy . . . . .	411
7.126.3.2 OsclError . . . . .	411
7.126.3.3 OsclErrorTrap . . . . .	411
7.126.3.4 OsclExecScheduler . . . . .	411
7.126.3.5 OsclExecSchedulerCommonBase . . . . .	411
7.126.3.6 OsclJump . . . . .	411
7.126.3.7 OsclJumpMark . . . . .	411
7.126.3.8 OsclScheduler . . . . .	411
7.126.3.9 OsclTrapStack . . . . .	411
7.126.4 Field Documentation . . . . .	411
7.126.4.1 iJumpData . . . . .	411
7.126.4.2 iLeave . . . . .	411
7.126.4.3 iTrapStack . . . . .	411
7.127 OsclException< LeaveCode > Class Template Reference . . . . .	412
7.127.1 Detailed Description . . . . .	412
7.127.2 Constructor & Destructor Documentation . . . . .	412
7.127.2.1 OsclException . . . . .	412
7.127.3 Member Function Documentation . . . . .	412
7.127.3.1 getLeaveCode . . . . .	412
7.128 OsclExclusiveArrayPtr< T > Class Template Reference . . . . .	413
7.128.1 Detailed Description . . . . .	413
7.128.2 Constructor & Destructor Documentation . . . . .	414
7.128.2.1 OsclExclusiveArrayPtr . . . . .	414
7.128.2.2 OsclExclusiveArrayPtr . . . . .	414
7.128.2.3 ~OsclExclusiveArrayPtr . . . . .	414
7.128.3 Member Function Documentation . . . . .	414
7.128.3.1 get . . . . .	414
7.128.3.2 operator* . . . . .	414

7.128.3.3 operator-> . . . . .	414
7.128.3.4 operator= . . . . .	415
7.128.3.5 release . . . . .	415
7.128.3.6 set . . . . .	415
7.128.4 Field Documentation . . . . .	415
7.128.4.1 _Ptr . . . . .	415
7.129 OsclExclusivePtr< T > Class Template Reference . . . . .	416
7.129.1 Detailed Description . . . . .	416
7.129.2 Constructor & Destructor Documentation . . . . .	417
7.129.2.1 OsclExclusivePtr . . . . .	417
7.129.2.2 OsclExclusivePtr . . . . .	417
7.129.2.3 ~OsclExclusivePtr . . . . .	417
7.129.3 Member Function Documentation . . . . .	417
7.129.3.1 get . . . . .	417
7.129.3.2 operator* . . . . .	417
7.129.3.3 operator-> . . . . .	417
7.129.3.4 operator= . . . . .	418
7.129.3.5 release . . . . .	418
7.129.3.6 set . . . . .	418
7.129.4 Field Documentation . . . . .	418
7.129.4.1 _Ptr . . . . .	418
7.130 OsclExclusivePtrA< T, Alloc > Class Template Reference . . . . .	419
7.130.1 Detailed Description . . . . .	419
7.130.2 Constructor & Destructor Documentation . . . . .	420
7.130.2.1 OsclExclusivePtrA . . . . .	420
7.130.2.2 OsclExclusivePtrA . . . . .	420
7.130.2.3 ~OsclExclusivePtrA . . . . .	420
7.130.3 Member Function Documentation . . . . .	420
7.130.3.1 get . . . . .	420
7.130.3.2 operator* . . . . .	420
7.130.3.3 operator-> . . . . .	421
7.130.3.4 operator= . . . . .	421
7.130.3.5 release . . . . .	421
7.130.3.6 set . . . . .	421
7.130.4 Field Documentation . . . . .	421
7.130.4.1 _Ptr . . . . .	421

7.131 OsclExecScheduler Class Reference . . . . .	423
7.131.1 Member Function Documentation . . . . .	423
7.131.1.1 Current . . . . .	423
7.131.1.2 RegisterForCallback . . . . .	423
7.131.1.3 RunSchedulerNonBlocking . . . . .	423
7.131.2 Friends And Related Function Documentation . . . . .	424
7.131.2.1 OsclScheduler . . . . .	424
7.132 OsclExecSchedulerBase Class Reference . . . . .	425
7.132.1 Detailed Description . . . . .	425
7.132.2 Friends And Related Function Documentation . . . . .	425
7.132.2.1 OsclCoeActiveScheduler . . . . .	425
7.132.2.2 OsclExecScheduler . . . . .	425
7.132.2.3 PVActiveBase . . . . .	425
7.133 OsclExecSchedulerCommonBase Class Reference . . . . .	426
7.133.1 Constructor & Destructor Documentation . . . . .	428
7.133.1.1 ~OsclExecSchedulerCommonBase . . . . .	428
7.133.1.2 OsclExecSchedulerCommonBase . . . . .	428
7.133.2 Member Function Documentation . . . . .	428
7.133.2.1 AddToExecTimerQ . . . . .	428
7.133.2.2 BeginScheduling . . . . .	428
7.133.2.3 BlockingLoopL . . . . .	428
7.133.2.4 CallRunExec . . . . .	428
7.133.2.5 CleanupExecQ . . . . .	428
7.133.2.6 ConstructL . . . . .	428
7.133.2.7 EndScheduling . . . . .	428
7.133.2.8 Error . . . . .	428
7.133.2.9 FindPVBase . . . . .	428
7.133.2.10 GetId . . . . .	428
7.133.2.11 GetName . . . . .	428
7.133.2.12 GetScheduler . . . . .	429
7.133.2.13 IncLogPerf . . . . .	429
7.133.2.14 InitExecQ . . . . .	429
7.133.2.15 InstallScheduler . . . . .	429
7.133.2.16 IsInstalled . . . . .	429
7.133.2.17 IsStarted . . . . .	429
7.133.2.18 PendComplete . . . . .	429

7.133.2.19	RequestCanceled	429
7.133.2.20	ResetLogPerf	429
7.133.2.21	ResumeScheduler	429
7.133.2.22	SetScheduler	429
7.133.2.23	StartNativeScheduler	429
7.133.2.24	StartScheduler	429
7.133.2.25	StopScheduler	429
7.133.2.26	SuspendScheduler	430
7.133.2.27	UninstallScheduler	430
7.133.2.28	UpdateTimers	430
7.133.2.29	UpdateTimersMsec	430
7.133.2.30	WaitForReadyAO	430
7.133.3	FRIENDS AND RELATED FUNCTION DOCUMENTATION	430
7.133.3.1	OsclActiveObject	430
7.133.3.2	OsclCoeActiveScheduler	430
7.133.3.3	OsclError	430
7.133.3.4	OsclExecScheduler	430
7.133.3.5	OsclReadyQ	430
7.133.3.6	OsclScheduler	430
7.133.3.7	OsclTimerCompare	432
7.133.3.8	OsclTimerObject	432
7.133.3.9	PVActiveBase	432
7.133.3.10	PVSchedulerStopper	432
7.133.3.11	IPVThreadContext	432
7.133.4	FIELD DOCUMENTATION	432
7.133.4.1	iAlloc	432
7.133.4.2	iBlockingMode	432
7.133.4.3	iDebugLogger	432
7.133.4.4	iDefAlloc	432
7.133.4.5	iDoStop	432
7.133.4.6	iDoSuspend	432
7.133.4.7	iErrorTrapImp	432
7.133.4.8	iExecTimerQ	432
7.133.4.9	iLogger	432
7.133.4.10	LogPerfIndentStr	432
7.133.4.11	LogPerfIndentStrLen	432

7.133.4.12 LogPerfTotal . . . . .	432
7.133.4.13 iName . . . . .	432
7.133.4.14 NativeMode . . . . .	432
7.133.4.15 iNumAOAdded . . . . .	432
7.133.4.16 ReadyQ . . . . .	432
7.133.4.17 iResumeSem . . . . .	432
7.133.4.18 Stopper . . . . .	432
7.133.4.19 StopperCrit . . . . .	432
7.133.4.20 iSuspended . . . . .	432
7.133.4.21 iThreadContext . . . . .	432
7.133.4.22 iTimeCompareThreshold . . . . .	432
7.134 OsclFileCache Class Reference . . . . .	434
7.134.1 Constructor & Destructor Documentation . . . . .	435
7.134.1.1 OsclFileCache . . . . .	435
7.134.1.2 ~OsclFileCache . . . . .	435
7.134.2 Member Function Documentation . . . . .	435
7.134.2.1 AddFixedCache . . . . .	435
7.134.2.2 Close . . . . .	435
7.134.2.3 EndOfFile . . . . .	435
7.134.2.4 FileSize . . . . .	435
7.134.2.5 Flush . . . . .	435
7.134.2.6 Open . . . . .	435
7.134.2.7 Read . . . . .	435
7.134.2.8 Seek . . . . .	435
7.134.2.9 Tell . . . . .	435
7.134.2.10 Write . . . . .	435
7.134.3 Friends And Related Function Documentation . . . . .	435
7.134.3.1 OsclFileCacheBuffer . . . . .	435
7.134.4 Field Documentation . . . . .	435
7.134.4.1 _fixedCaches . . . . .	435
7.134.4.2 _movableCache . . . . .	435
7.135 OsclFileCacheBuffer Class Reference . . . . .	436
7.135.1 Constructor & Destructor Documentation . . . . .	436
7.135.1.1 OsclFileCacheBuffer . . . . .	436
7.135.2 Member Function Documentation . . . . .	436
7.135.2.1 Contains . . . . .	436

7.135.2.2 FillFromFile . . . . .	436
7.135.2.3 IsUpdated . . . . .	436
7.135.2.4 Preceeds . . . . .	436
7.135.2.5 PrepRead . . . . .	437
7.135.2.6 PrepWrite . . . . .	437
7.135.2.7 SetPosition . . . . .	437
7.135.2.8 WriteUpdatesToFile . . . . .	437
7.135.3 Field Documentation . . . . .	437
7.135.3.1 capacity . . . . .	437
7.135.3.2 currentPos . . . . .	437
7.135.3.3 endPos . . . . .	437
7.135.3.4 filePosition . . . . .	437
7.135.3.5 iContainer . . . . .	437
7.135.3.6 isFixed . . . . .	437
7.135.3.7 pBuffer . . . . .	437
7.135.3.8 updateEnd . . . . .	437
7.135.3.9 updateStart . . . . .	437
7.135.3.10 usableSize . . . . .	437
7.136 OsclFileHandle Class Reference . . . . .	438
7.136.1 Detailed Description . . . . .	438
7.136.2 Constructor & Destructor Documentation . . . . .	438
7.136.2.1 OsclFileHandle . . . . .	438
7.136.2.2 OsclFileHandle . . . . .	438
7.136.3 Member Function Documentation . . . . .	438
7.136.3.1 Handle . . . . .	438
7.136.4 Friends And Related Function Documentation . . . . .	438
7.136.4.1 Oscl_File . . . . .	438
7.137 OsclFileManager Class Reference . . . . .	439
7.137.1 Member Enumeration Documentation . . . . .	439
7.137.1.1 OSCL_FILE_ATTRIBUTE_TYPE . . . . .	439
7.138 OsclFileStats Class Reference . . . . .	441
7.138.1 Constructor & Destructor Documentation . . . . .	441
7.138.1.1 OsclFileStats . . . . .	441
7.138.2 Member Function Documentation . . . . .	441
7.138.2.1 End . . . . .	441
7.138.2.2 Log . . . . .	441

---

7.138.2.3 LogAll . . . . .	441
7.138.2.4 Start . . . . .	441
7.139 OsclFileStatsItem Class Reference . . . . .	442
7.139.1 Field Documentation . . . . .	442
7.139.1.1 iOpCount . . . . .	442
7.139.1.2 iParam . . . . .	442
7.139.1.3 iParam2 . . . . .	442
7.139.1.4 iStartTick . . . . .	442
7.139.1.5 iTotalsTicks . . . . .	442
7.140 Oscl_File::OsclFixedCacheParam Class Reference . . . . .	443
7.140.1 Detailed Description . . . . .	443
7.140.2 Member Function Documentation . . . . .	443
7.140.2.1 Contains . . . . .	443
7.140.3 Field Documentation . . . . .	443
7.140.3.1 iFilePath . . . . .	443
7.140.3.2 iSize . . . . .	443
7.141 OsclGetHostByNameMethod Class Reference . . . . .	444
7.141.1 Constructor & Destructor Documentation . . . . .	444
7.141.1.1 ~OsclGetHostByNameMethod . . . . .	444
7.141.2 Member Function Documentation . . . . .	444
7.141.2.1 GetHostByName . . . . .	444
7.141.2.2 NewL . . . . .	444
7.142 OsclGetHostByNameRequest Class Reference . . . . .	445
7.142.1 Friends And Related Function Documentation . . . . .	445
7.142.1.1 OsclGetHostByNameMethod . . . . .	445
7.143 OsclInit Class Reference . . . . .	446
7.143.1 Detailed Description . . . . .	446
7.143.2 Member Function Documentation . . . . .	446
7.143.2.1 Cleanup . . . . .	446
7.143.2.2 Init . . . . .	446
7.144 OsclInteger64Transport Struct Reference . . . . .	447
7.144.1 Detailed Description . . . . .	447
7.144.2 Field Documentation . . . . .	447
7.144.2.1 iHigh . . . . .	447
7.144.2.2 iLow . . . . .	447
7.145 OsclIpMReq Class Reference . . . . .	448

7.145.1 Constructor & Destructor Documentation . . . . .	448
7.145.1.1 OsclIpMReq . . . . .	448
7.145.2 Field Documentation . . . . .	448
7.145.2.1 interfaceAddr . . . . .	448
7.145.2.2 multicastAddr . . . . .	448
7.146 OsclIPSocketI Class Reference . . . . .	449
7.146.1 Constructor & Destructor Documentation . . . . .	450
7.146.1.1 ~OsclIPSocketI . . . . .	450
7.146.1.2 OsclIPSocketI . . . . .	450
7.146.2 Member Function Documentation . . . . .	450
7.146.2.1 Alloc . . . . .	450
7.146.2.2 Bind . . . . .	450
7.146.2.3 Close . . . . .	450
7.146.2.4 ConstructL . . . . .	450
7.146.2.5 GetPeerName . . . . .	450
7.146.2.6 GetRecvData . . . . .	450
7.146.2.7 GetSendData . . . . .	450
7.146.2.8 Join . . . . .	450
7.146.2.9 SetOptionToReuseAddress . . . . .	450
7.146.2.10 SetRecvBufferSize . . . . .	450
7.146.2.11 SetTOS . . . . .	450
7.146.2.12 SocketServ . . . . .	450
7.146.2.13 ThreadLogoff . . . . .	450
7.146.2.14 ThreadLogon . . . . .	451
7.146.3 Friends And Related Function Documentation . . . . .	451
7.146.3.1 OsclSocketMethod . . . . .	451
7.146.3.2 OsclSocketRequestAO . . . . .	451
7.146.4 Field Documentation . . . . .	451
7.146.4.1 iAddress . . . . .	451
7.146.4.2 iAlloc . . . . .	451
7.146.4.3 iId . . . . .	451
7.146.4.4 iLogger . . . . .	451
7.146.4.5 iObserver . . . . .	451
7.146.4.6 iSocket . . . . .	451
7.146.4.7 iSocketServ . . . . .	451
7.147 OsclJump Class Reference . . . . .	452

---

7.147.1 Constructor & Destructor Documentation . . . . .	452
7.147.1.1 ~OsclJump . . . . .	452
7.147.2 Member Function Documentation . . . . .	452
7.147.2.1 Jump . . . . .	452
7.147.2.2 StaticJump . . . . .	452
7.147.2.3 Top . . . . .	452
7.147.3 Friends And Related Function Documentation . . . . .	452
7.147.3.1 OsclErrorTrapImp . . . . .	452
7.148 OsclListenMethod Class Reference . . . . .	453
7.148.1 Constructor & Destructor Documentation . . . . .	453
7.148.1.1 ~OsclListenMethod . . . . .	453
7.148.2 Member Function Documentation . . . . .	453
7.148.2.1 Listen . . . . .	453
7.148.2.2 ListenRequest . . . . .	453
7.148.2.3 NewL . . . . .	453
7.149 OsclListenRequest Class Reference . . . . .	454
7.149.1 Detailed Description . . . . .	454
7.149.2 Constructor & Destructor Documentation . . . . .	454
7.149.2.1 OsclListenRequest . . . . .	454
7.149.3 Member Function Documentation . . . . .	454
7.149.3.1 Listen . . . . .	454
7.150 OsclLockBase Class Reference . . . . .	455
7.150.1 Constructor & Destructor Documentation . . . . .	455
7.150.1.1 ~OsclLockBase . . . . .	455
7.150.2 Member Function Documentation . . . . .	455
7.150.2.1 Lock . . . . .	455
7.150.2.2 Unlock . . . . .	455
7.151 OsclMem Class Reference . . . . .	456
7.151.1 Member Function Documentation . . . . .	456
7.151.1.1 Cleanup . . . . .	456
7.151.1.2 Init . . . . .	456
7.152 OsclMemAllocator Class Reference . . . . .	457
7.152.1 Detailed Description . . . . .	457
7.152.2 Member Function Documentation . . . . .	457
7.152.2.1 allocate . . . . .	457
7.152.2.2 deallocate . . . . .	457

7.153 OsclMemAllocDestructDealloc< T > Class Template Reference . . . . .	458
7.153.1 Detailed Description . . . . .	458
7.153.2 Member Function Documentation . . . . .	458
7.153.2.1 allocate . . . . .	458
7.153.2.2 deallocate . . . . .	458
7.153.2.3 destruct_and_dealloc . . . . .	458
7.154 OsclMemAudit Class Reference . . . . .	460
7.155 OSCLMemAutoPtr< T, _Allocator > Class Template Reference . . . . .	461
7.155.1 Detailed Description . . . . .	462
7.155.2 Constructor & Destructor Documentation . . . . .	462
7.155.2.1 OSCLMemAutoPtr . . . . .	462
7.155.2.2 OSCLMemAutoPtr . . . . .	462
7.155.2.3 ~OSCLMemAutoPtr . . . . .	462
7.155.3 Member Function Documentation . . . . .	462
7.155.3.1 allocate . . . . .	462
7.155.3.2 deallocate . . . . .	463
7.155.3.3 get . . . . .	463
7.155.3.4 operator* . . . . .	463
7.155.3.5 operator-> . . . . .	463
7.155.3.6 operator= . . . . .	463
7.155.3.7 release . . . . .	463
7.155.3.8 setWithoutOwnership . . . . .	464
7.155.3.9 takeOwnership . . . . .	464
7.155.4 Field Documentation . . . . .	464
7.155.4.1 _Ownership . . . . .	464
7.156 OsclMemBasicAllocator Class Reference . . . . .	465
7.156.1 Detailed Description . . . . .	465
7.156.2 Member Function Documentation . . . . .	465
7.156.2.1 allocate . . . . .	465
7.156.2.2 deallocate . . . . .	465
7.157 OsclMemBasicAllocDestructDealloc< T > Class Template Reference . . . . .	466
7.157.1 Detailed Description . . . . .	466
7.157.2 Member Function Documentation . . . . .	466
7.157.2.1 allocate . . . . .	466
7.157.2.2 deallocate . . . . .	466
7.157.2.3 destruct_and_dealloc . . . . .	466

7.158OsclMemGlobalAuditObject Class Reference . . . . .	468
7.158.1 Member Typedef Documentation . . . . .	468
7.158.1.1 audit_type . . . . .	468
7.158.2 Member Function Documentation . . . . .	468
7.158.2.1 getGlobalMemAuditObject . . . . .	468
7.158.3 Friends And Related Function Documentation . . . . .	468
7.158.3.1 OsclMem . . . . .	468
7.159OsclMemoryFragment Struct Reference . . . . .	469
7.159.1 Field Documentation . . . . .	469
7.159.1.1 len . . . . .	469
7.159.1.2 ptr . . . . .	469
7.160OsclMemPoolFixedChunkAllocator Class Reference . . . . .	470
7.160.1 Constructor & Destructor Documentation . . . . .	471
7.160.1.1 OsclMemPoolFixedChunkAllocator . . . . .	471
7.160.1.2 ~OsclMemPoolFixedChunkAllocator . . . . .	471
7.160.2 Member Function Documentation . . . . .	471
7.160.2.1 addRef . . . . .	471
7.160.2.2 allocate . . . . .	471
7.160.2.3 CancelFreeChunkAvailableCallback . . . . .	471
7.160.2.4 createmempool . . . . .	472
7.160.2.5 deallocate . . . . .	472
7.160.2.6 destroymempool . . . . .	472
7.160.2.7 enablenullpointerreturn . . . . .	472
7.160.2.8 notifyfreechunkavailable . . . . .	472
7.160.2.9 removeRef . . . . .	472
7.160.3 Field Documentation . . . . .	473
7.160.3.1 iCheckNextAvailableFreeChunk . . . . .	473
7.160.3.2 iChunkAlignment . . . . .	473
7.160.3.3 iChunkSize . . . . .	473
7.160.3.4 iChunkSizeMemAligned . . . . .	473
7.160.3.5 iEnableNullPtrReturn . . . . .	473
7.160.3.6 iFreeMemChunkList . . . . .	473
7.160.3.7 iMemPool . . . . .	473
7.160.3.8 iMemPoolAligned . . . . .	473
7.160.3.9 iMemPoolAllocator . . . . .	473
7.160.3.10 iNextAvailableContextData . . . . .	473

7.160.3.1 <code>liNumChunk</code>	473
7.160.3.12 <code>Observer</code>	473
7.160.3.13 <code>RefCount</code>	473
7.161 <code>OsclMemPoolFixedChunkAllocatorObserver</code> Class Reference	474
7.161.1 Constructor & Destructor Documentation	474
7.161.1.1 <code>~OsclMemPoolFixedChunkAllocatorObserver</code>	474
7.161.2 Member Function Documentation	474
7.161.2.1 <code>freechunkavailable</code>	474
7.162 <code>OsclMemPoolResizableAllocator</code> Class Reference	475
7.162.1 Constructor & Destructor Documentation	476
7.162.1.1 <code>OsclMemPoolResizableAllocator</code>	476
7.162.1.2 <code>~OsclMemPoolResizableAllocator</code>	477
7.162.2 Member Function Documentation	477
7.162.2.1 <code>addnewmempoolbuffer</code>	477
7.162.2.2 <code>addRef</code>	477
7.162.2.3 <code>allocate</code>	477
7.162.2.4 <code>allocateblock</code>	477
7.162.2.5 <code>CancelFreeChunkAvailableCallback</code>	477
7.162.2.6 <code>CancelFreeMemoryAvailableCallback</code>	477
7.162.2.7 <code>deallocate</code>	477
7.162.2.8 <code>deallocateblock</code>	478
7.162.2.9 <code>destroyallmempoolbuffers</code>	478
7.162.2.10 <code>enablenullpointerreturn</code>	478
7.162.2.11 <code>findfreeblock</code>	478
7.162.2.12 <code>getAllocatedSize</code>	478
7.162.2.13 <code>getAvailableSize</code>	478
7.162.2.14 <code>getBufferSize</code>	478
7.162.2.15 <code>getLargestContiguousFreeBlockSize</code>	478
7.162.2.16 <code>getMemPoolBufferAllocatedSize</code>	479
7.162.2.17 <code>getMemPoolBufferSize</code>	479
7.162.2.18 <code>memoryPoolBufferMgmtOverhead</code>	479
7.162.2.19 <code>notifyfreeblockavailable</code>	479
7.162.2.20 <code>notifyfreememoryavailable</code>	479
7.162.2.21 <code>removeRef</code>	479
7.162.2.22 <code>setMaxSzForNewMemPoolBuffer</code>	479
7.162.2.23 <code>trim</code>	479

7.162.2.24validateblock . . . . .	482
7.162.3 Field Documentation . . . . .	482
7.162.3.1 iBlockInfoAlignedSize . . . . .	482
7.162.3.2 iBufferInfoAlignedSize . . . . .	482
7.162.3.3 iCheckFreeMemoryAvailable . . . . .	482
7.162.3.4 iCheckNextAvailable . . . . .	482
7.162.3.5 iDebugLogger . . . . .	482
7.162.3.6 iEnableNullPtrReturn . . . . .	482
7.162.3.7 iExpectedNumBlocksPerBuffer . . . . .	482
7.162.3.8 iFreeMemContextData . . . . .	482
7.162.3.9 iFreeMemPoolObserver . . . . .	482
7.162.3.10 iMaxNewMemPoolBufferSz . . . . .	482
7.162.3.11 iMemPoolBufferAllocator . . . . .	482
7.162.3.12 iMemPoolBufferList . . . . .	482
7.162.3.13 iMemPoolBufferNumLimit . . . . .	482
7.162.3.14 iMemPoolBufferSize . . . . .	482
7.162.3.15 iMemPoolPrevAllocBufferIndex . . . . .	482
7.162.3.16 iNextAvailableContextData . . . . .	482
7.162.3.17 iObserver . . . . .	482
7.162.3.18 iRefCount . . . . .	482
7.162.3.19 iRequestedAvailableFreeMemSize . . . . .	482
7.162.3.20 iRequestedNextAvailableSize . . . . .	482
7.163 OsclMemPoolResizableAllocatorMemoryObserver Class Reference . . . . .	484
7.163.1 Constructor & Destructor Documentation . . . . .	484
7.163.1.1 ~OsclMemPoolResizableAllocatorMemoryObserver . . . . .	484
7.163.2 Member Function Documentation . . . . .	484
7.163.2.1 freememoryavailable . . . . .	484
7.164 OsclMemPoolResizableAllocatorObserver Class Reference . . . . .	485
7.164.1 Constructor & Destructor Documentation . . . . .	485
7.164.1.1 ~OsclMemPoolResizableAllocatorObserver . . . . .	485
7.164.2 Member Function Documentation . . . . .	485
7.164.2.1 freeblockavailable . . . . .	485
7.165 OsclMemStatsNode Class Reference . . . . .	486
7.165.1 Constructor & Destructor Documentation . . . . .	486
7.165.1.1 OsclMemStatsNode . . . . .	486
7.165.1.2 ~OsclMemStatsNode . . . . .	486

7.165.2 Member Function Documentation . . . . .	486
7.165.2.1 operator delete . . . . .	486
7.165.2.2 operator new . . . . .	486
7.165.2.3 operator new . . . . .	486
7.165.2.4 reset . . . . .	486
7.165.3 Field Documentation . . . . .	487
7.165.3.1 pMMFIParam . . . . .	487
7.165.3.2 pMMStats . . . . .	487
7.165.3.3 tag . . . . .	487
7.166 OsclMutex Class Reference . . . . .	488
7.166.1 Detailed Description . . . . .	488
7.166.2 Constructor & Destructor Documentation . . . . .	488
7.166.2.1 OsclMutex . . . . .	488
7.166.2.2 ~OsclMutex . . . . .	488
7.166.3 Member Function Documentation . . . . .	488
7.166.3.1 Close . . . . .	488
7.166.3.2 Create . . . . .	489
7.166.3.3 Lock . . . . .	489
7.166.3.4 TryLock . . . . .	489
7.166.3.5 Unlock . . . . .	489
7.167 OsclNameString< __len > Class Template Reference . . . . .	490
7.167.1 Detailed Description . . . . .	490
7.167.2 Constructor & Destructor Documentation . . . . .	490
7.167.2.1 OsclNameString . . . . .	490
7.167.2.2 OsclNameString . . . . .	490
7.167.2.3 OsclNameString . . . . .	490
7.167.3 Member Function Documentation . . . . .	490
7.167.3.1 MaxLen . . . . .	490
7.167.3.2 Set . . . . .	490
7.167.3.3 Set . . . . .	490
7.167.3.4 Str . . . . .	491
7.168 OsclNativeFile Class Reference . . . . .	492
7.168.1 Constructor & Destructor Documentation . . . . .	493
7.168.1.1 OsclNativeFile . . . . .	493
7.168.1.2 ~OsclNativeFile . . . . .	493
7.168.2 Member Function Documentation . . . . .	493

---

7.168.2.1 Close . . . . .	493
7.168.2.2 EndOfFile . . . . .	493
7.168.2.3 Flush . . . . .	493
7.168.2.4 GetError . . . . .	493
7.168.2.5 GetReadAsyncNumElements . . . . .	493
7.168.2.6 HasAsyncRead . . . . .	493
7.168.2.7 Mode . . . . .	493
7.168.2.8 Open . . . . .	493
7.168.2.9 Open . . . . .	493
7.168.2.10Open . . . . .	493
7.168.2.11Read . . . . .	493
7.168.2.12ReadAsync . . . . .	493
7.168.2.13ReadAsyncCancel . . . . .	494
7.168.2.14Seek . . . . .	494
7.168.2.15SetSize . . . . .	494
7.168.2.16Size . . . . .	494
7.168.2.17Tell . . . . .	494
7.168.2.18Write . . . . .	494
7.169OsclNativeFileParams Class Reference . . . . .	495
7.169.1 Constructor & Destructor Documentation . . . . .	495
7.169.1.1 OsclNativeFileParams . . . . .	495
7.169.2 Field Documentation . . . . .	495
7.169.2.1 iAsyncReadBufferSize . . . . .	495
7.169.2.2 iNativeAccessMode . . . . .	495
7.169.2.3 iNativeBufferSize . . . . .	495
7.170OsclNetworkAddress Class Reference . . . . .	496
7.170.1 Constructor & Destructor Documentation . . . . .	496
7.170.1.1 OsclNetworkAddress . . . . .	496
7.170.1.2 OsclNetworkAddress . . . . .	496
7.170.2 Member Function Documentation . . . . .	496
7.170.2.1 operator== . . . . .	496
7.170.3 Field Documentation . . . . .	496
7.170.3.1 ipAddr . . . . .	496
7.170.3.2 port . . . . .	496
7.171OsclNullLock Class Reference . . . . .	497
7.171.1 Constructor & Destructor Documentation . . . . .	497

7.171.1.1 ~OsclNullLock . . . . .	497
7.171.2 Member Function Documentation . . . . .	497
7.171.2.1 Lock . . . . .	497
7.171.2.2 Unlock . . . . .	497
7.172 OsclPriorityLink Class Reference . . . . .	498
7.172.1 Field Documentation . . . . .	498
7.172.1.1 iPriority . . . . .	498
7.173 OsclPriorityList< T > Class Template Reference . . . . .	499
7.173.1 Constructor & Destructor Documentation . . . . .	499
7.173.1.1 OsclPriorityList . . . . .	499
7.173.1.2 OsclPriorityList . . . . .	499
7.173.2 Member Function Documentation . . . . .	499
7.173.2.1 Head . . . . .	499
7.173.2.2 Insert . . . . .	499
7.173.2.3 IsHead . . . . .	499
7.173.2.4 IsTail . . . . .	499
7.173.2.5 Tail . . . . .	499
7.174 OsclPriorityQueue< Qelem, Alloc, Container, Compare > Class Template Reference . . . . .	500
7.174.1 Member Typedef Documentation . . . . .	501
7.174.1.1 const_reference . . . . .	501
7.174.1.2 container_type . . . . .	501
7.174.1.3 iterator . . . . .	501
7.174.1.4 value_type . . . . .	501
7.174.2 Constructor & Destructor Documentation . . . . .	501
7.174.2.1 OsclPriorityQueue . . . . .	501
7.174.2.2 ~OsclPriorityQueue . . . . .	501
7.174.3 Member Function Documentation . . . . .	501
7.174.3.1 compare_EQ . . . . .	501
7.174.3.2 compare_LT . . . . .	501
7.174.3.3 empty . . . . .	502
7.174.3.4 find_heap . . . . .	502
7.174.3.5 pop . . . . .	502
7.174.3.6 pop_heap . . . . .	502
7.174.3.7 push . . . . .	502
7.174.3.8 push_heap . . . . .	502
7.174.3.9 remove . . . . .	502

---

7.174.3.10reserve . . . . .	503
7.174.3.1 lsize . . . . .	503
7.174.3.12swap . . . . .	503
7.174.3.13top . . . . .	503
7.174.3.14validate . . . . .	503
7.174.3.15vec . . . . .	503
7.174.4 Friends And Related Function Documentation . . . . .	503
7.174.4.1 oscl_priqueue_test . . . . .	503
7.174.5 Field Documentation . . . . .	503
7.174.5.1 c . . . . .	503
7.174.5.2 comp . . . . .	504
7.175 OsclPriorityQueueBase Class Reference . . . . .	505
7.175.1 Detailed Description . . . . .	505
7.175.2 Constructor & Destructor Documentation . . . . .	505
7.175.2.1 ~OsclPriorityQueueBase . . . . .	505
7.175.3 Member Function Documentation . . . . .	505
7.175.3.1 construct . . . . .	505
7.175.3.2 find_heap . . . . .	505
7.175.3.3 pop_heap . . . . .	505
7.175.3.4 push_heap . . . . .	505
7.175.3.5 remove . . . . .	505
7.176 OsclProcStatus Class Reference . . . . .	507
7.176.1 Detailed Description . . . . .	507
7.176.2 Member Enumeration Documentation . . . . .	507
7.176.2.1 eOsclProcError . . . . .	507
7.177 OsclPtr Class Reference . . . . .	509
7.177.1 Constructor & Destructor Documentation . . . . .	509
7.177.1.1 OsclPtr . . . . .	509
7.177.1.2 OsclPtr . . . . .	509
7.177.2 Member Function Documentation . . . . .	509
7.177.2.1 Append . . . . .	509
7.177.2.2 Length . . . . .	509
7.177.2.3 Ptr . . . . .	509
7.177.2.4 Set . . . . .	509
7.177.2.5 Set . . . . .	509
7.177.2.6 SetLength . . . . .	509

7.177.2.7 Zero . . . . .	510
7.178 OsclPtrC Class Reference . . . . .	511
7.178.1 Constructor & Destructor Documentation . . . . .	511
7.178.1.1 OsclPtrC . . . . .	511
7.178.1.2 OsclPtrC . . . . .	511
7.178.2 Member Function Documentation . . . . .	511
7.178.2.1 Left . . . . .	511
7.178.2.2 Length . . . . .	511
7.178.2.3 Ptr . . . . .	511
7.178.2.4 Right . . . . .	512
7.178.2.5 Set . . . . .	512
7.178.2.6 Set . . . . .	512
7.178.2.7 SetLength . . . . .	512
7.178.2.8 Zero . . . . .	512
7.179 OsclRand Class Reference . . . . .	513
7.179.1 Member Function Documentation . . . . .	513
7.179.1.1 Rand . . . . .	513
7.179.1.2 Seed . . . . .	513
7.180 OsclReadyAlloc Class Reference . . . . .	514
7.180.1 Member Function Documentation . . . . .	514
7.180.1.1 allocate . . . . .	514
7.180.1.2 allocate_fl . . . . .	514
7.180.1.3 deallocate . . . . .	514
7.181 OsclReadyCompare Class Reference . . . . .	515
7.181.1 Member Function Documentation . . . . .	515
7.181.1.1 compare . . . . .	515
7.182 OsclReadyQ Class Reference . . . . .	516
7.182.1 Member Function Documentation . . . . .	516
7.182.1.1 Callback . . . . .	516
7.182.1.2 Construct . . . . .	516
7.182.1.3 Depth . . . . .	516
7.182.1.4 IsIn . . . . .	517
7.182.1.5 PendComplete . . . . .	517
7.182.1.6 PopTop . . . . .	517
7.182.1.7 RegisterForCallback . . . . .	517
7.182.1.8 Remove . . . . .	517

---

7.182.1.9 ThreadLogoff . . . . .	517
7.182.1.10 ThreadLogon . . . . .	517
7.182.1.11 ITimerCallback . . . . .	517
7.182.1.12 Top . . . . .	517
7.182.1.13 WaitAndPopTop . . . . .	517
7.182.1.14 WaitAndPopTop . . . . .	517
7.182.1.15 WaitForRequestComplete . . . . .	517
7.183 OsclRecvFromMethod Class Reference . . . . .	518
7.183.1 Constructor & Destructor Documentation . . . . .	518
7.183.1.1 ~OsclRecvFromMethod . . . . .	518
7.183.2 Member Function Documentation . . . . .	518
7.183.2.1 GetRecvData . . . . .	518
7.183.2.2 NewL . . . . .	519
7.183.2.3 RecvFrom . . . . .	519
7.183.2.4 RecvFromRequest . . . . .	519
7.184 OsclRecvFromRequest Class Reference . . . . .	520
7.184.1 Detailed Description . . . . .	520
7.184.2 Constructor & Destructor Documentation . . . . .	520
7.184.2.1 OsclRecvFromRequest . . . . .	520
7.184.3 Member Function Documentation . . . . .	520
7.184.3.1 GetRecvData . . . . .	520
7.184.3.2 RecvFrom . . . . .	520
7.184.3.3 Success . . . . .	520
7.185 OsclRecvMethod Class Reference . . . . .	522
7.185.1 Constructor & Destructor Documentation . . . . .	522
7.185.1.1 ~OsclRecvMethod . . . . .	522
7.185.2 Member Function Documentation . . . . .	522
7.185.2.1 GetRecvData . . . . .	522
7.185.2.2 NewL . . . . .	522
7.185.2.3 Recv . . . . .	522
7.185.2.4 RecvRequest . . . . .	523
7.186 OsclRecvRequest Class Reference . . . . .	524
7.186.1 Detailed Description . . . . .	524
7.186.2 Constructor & Destructor Documentation . . . . .	524
7.186.2.1 OsclRecvRequest . . . . .	524
7.186.3 Member Function Documentation . . . . .	524

7.186.3.1 GetRecvData . . . . .	524
7.186.3.2 Recv . . . . .	524
7.186.3.3 Success . . . . .	524
7.187 OsclRefCounter Class Reference . . . . .	525
7.187.1 Detailed Description . . . . .	525
7.187.2 Constructor & Destructor Documentation . . . . .	525
7.187.2.1 ~OsclRefCounter . . . . .	525
7.187.3 Member Function Documentation . . . . .	525
7.187.3.1 addRef . . . . .	525
7.187.3.2 getCount . . . . .	525
7.187.3.3 removeRef . . . . .	525
7.188 OsclRefCounterDA Class Reference . . . . .	527
7.188.1 Detailed Description . . . . .	527
7.188.2 Constructor & Destructor Documentation . . . . .	527
7.188.2.1 OsclRefCounterDA . . . . .	527
7.188.2.2 ~OsclRefCounterDA . . . . .	527
7.188.3 Member Function Documentation . . . . .	528
7.188.3.1 addRef . . . . .	528
7.188.3.2 getCount . . . . .	528
7.188.3.3 removeRef . . . . .	528
7.189 OsclRefCounterMemFrag Class Reference . . . . .	529
7.189.1 Detailed Description . . . . .	529
7.189.2 Constructor & Destructor Documentation . . . . .	529
7.189.2.1 OsclRefCounterMemFrag . . . . .	529
7.189.2.2 OsclRefCounterMemFrag . . . . .	529
7.189.2.3 OsclRefCounterMemFrag . . . . .	529
7.189.2.4 ~OsclRefCounterMemFrag . . . . .	530
7.189.3 Member Function Documentation . . . . .	530
7.189.3.1 getCapacity . . . . .	530
7.189.3.2 getCount . . . . .	530
7.189.3.3 getMemFrag . . . . .	530
7.189.3.4 getMemFragPtr . . . . .	530
7.189.3.5 getMemFragSize . . . . .	530
7.189.3.6 getRefCounter . . . . .	530
7.189.3.7 operator= . . . . .	530
7.190 OsclRefCounterMTDA< LockType > Class Template Reference . . . . .	531

---

7.190.1 Detailed Description . . . . .	531
7.190.2 Constructor & Destructor Documentation . . . . .	531
7.190.2.1 OsclRefCounterMTDA . . . . .	531
7.190.2.2 ~OsclRefCounterMTDA . . . . .	532
7.190.3 Member Function Documentation . . . . .	532
7.190.3.1 addRef . . . . .	532
7.190.3.2 getCount . . . . .	532
7.190.3.3 removeRef . . . . .	532
7.191 OsclRefCounterMTSA< DeallocType, LockType > Class Template Reference . . . . .	533
7.191.1 Detailed Description . . . . .	533
7.191.2 Constructor & Destructor Documentation . . . . .	533
7.191.2.1 OsclRefCounterMTSA . . . . .	533
7.191.2.2 ~OsclRefCounterMTSA . . . . .	533
7.191.3 Member Function Documentation . . . . .	534
7.191.3.1 addRef . . . . .	534
7.191.3.2 getCount . . . . .	534
7.191.3.3 removeRef . . . . .	534
7.192 OsclRefCounterSA< DeallocType > Class Template Reference . . . . .	535
7.192.1 Detailed Description . . . . .	535
7.192.2 Constructor & Destructor Documentation . . . . .	535
7.192.2.1 OsclRefCounterSA . . . . .	535
7.192.2.2 ~OsclRefCounterSA . . . . .	535
7.192.3 Member Function Documentation . . . . .	536
7.192.3.1 addRef . . . . .	536
7.192.3.2 getCount . . . . .	536
7.192.3.3 removeRef . . . . .	536
7.193 OsclRegistryAccessClient Class Reference . . . . .	537
7.193.1 Constructor & Destructor Documentation . . . . .	537
7.193.1.1 OsclRegistryAccessClient . . . . .	537
7.193.1.2 ~OsclRegistryAccessClient . . . . .	537
7.193.2 Member Function Documentation . . . . .	537
7.193.2.1 Close . . . . .	537
7.193.2.2 Connect . . . . .	537
7.193.2.3 GetFactories . . . . .	537
7.193.2.4 GetFactory . . . . .	537
7.194 OsclRegistryAccessClientImpl Class Reference . . . . .	539

7.195 OsclRegistryAccessClientTlsImpl Class Reference . . . . .	540
7.196 OsclRegistryAccessElement Class Reference . . . . .	541
7.196.1 Detailed Description . . . . .	541
7.196.2 Field Documentation . . . . .	541
7.196.2.1 iFactory . . . . .	541
7.196.2.2 iMimeType . . . . .	541
7.197 OsclRegistryClient Class Reference . . . . .	542
7.197.1 Constructor & Destructor Documentation . . . . .	542
7.197.1.1 OsclRegistryClient . . . . .	542
7.197.1.2 ~OsclRegistryClient . . . . .	542
7.197.2 Member Function Documentation . . . . .	542
7.197.2.1 Close . . . . .	542
7.197.2.2 Connect . . . . .	542
7.197.2.3 Register . . . . .	543
7.197.2.4 UnRegister . . . . .	543
7.198 OsclRegistryClientImpl Class Reference . . . . .	544
7.198.1 Member Function Documentation . . . . .	544
7.198.1.1 Close . . . . .	544
7.198.1.2 Connect . . . . .	544
7.198.1.3 GetFactories . . . . .	544
7.198.1.4 GetFactory . . . . .	544
7.198.1.5 Register . . . . .	544
7.198.1.6 UnRegister . . . . .	545
7.198.2 Friends And Related Function Documentation . . . . .	545
7.198.2.1 OsclRegistryAccessClient . . . . .	545
7.198.2.2 OsclRegistryClient . . . . .	545
7.199 OsclRegistryClientTlsImpl Class Reference . . . . .	546
7.200 OsclRegistryServTlsImpl Class Reference . . . . .	547
7.200.1 Constructor & Destructor Documentation . . . . .	548
7.200.1.1 OsclRegistryServTlsImpl . . . . .	548
7.200.1.2 ~OsclRegistryServTlsImpl . . . . .	548
7.200.2 Member Function Documentation . . . . .	548
7.200.2.1 Close . . . . .	548
7.200.2.2 Connect . . . . .	548
7.200.2.3 GetFactories . . . . .	548
7.200.2.4 GetFactory . . . . .	548

7.200.2.5 Register . . . . .	548
7.200.2.6 UnRegister . . . . .	548
7.200.3 Friends And Related Function Documentation . . . . .	548
7.200.3.1 OsclRegistryAccessClient . . . . .	548
7.200.3.2 OsclRegistryClient . . . . .	548
7.201 OsclScheduler Class Reference . . . . .	549
7.201.1 Detailed Description . . . . .	549
7.201.2 Member Function Documentation . . . . .	549
7.201.2.1 Cleanup . . . . .	549
7.201.2.2 Init . . . . .	549
7.202 OsclSchedulerObserver Class Reference . . . . .	550
7.202.1 Detailed Description . . . . .	550
7.202.2 Constructor & Destructor Documentation . . . . .	550
7.202.2.1 ~OsclSchedulerObserver . . . . .	550
7.202.3 Member Function Documentation . . . . .	550
7.202.3.1 OsclSchedulerReadyCallback . . . . .	550
7.202.3.2 OsclSchedulerTimerCallback . . . . .	550
7.203 OsclScopedLock< LockClass > Class Template Reference . . . . .	551
7.203.1 Detailed Description . . . . .	551
7.203.2 Constructor & Destructor Documentation . . . . .	551
7.203.2.1 OsclScopedLock . . . . .	551
7.203.2.2 ~OsclScopedLock . . . . .	551
7.204 OsclSelect Class Reference . . . . .	552
7.204.1 Detailed Description . . . . .	552
7.204.2 Constructor & Destructor Documentation . . . . .	553
7.204.2.1 OsclSelect . . . . .	553
7.204.2.2 OsclSelect . . . . .	553
7.204.3 Field Documentation . . . . .	553
7.204.3.1 iErrAlloc . . . . .	553
7.204.3.2 iHeapCheck . . . . .	553
7.204.3.3 iOsclBase . . . . .	553
7.204.3.4 iOsclErrorTrap . . . . .	553
7.204.3.5 iOsclLogger . . . . .	553
7.204.3.6 iOsclMemory . . . . .	553
7.204.3.7 iOsclScheduler . . . . .	553
7.204.3.8 iOutputFile . . . . .	553

7.204.3.9 iSchedulerAlloc . . . . .	553
7.204.3.10 iSchedulerName . . . . .	553
7.204.3.11 iSchedulerReserve . . . . .	553
7.205 OsclSemaphore Class Reference . . . . .	554
7.205.1 Detailed Description . . . . .	554
7.205.2 Constructor & Destructor Documentation . . . . .	554
7.205.2.1 OsclSemaphore . . . . .	554
7.205.2.2 ~OsclSemaphore . . . . .	554
7.205.3 Member Function Documentation . . . . .	554
7.205.3.1 Close . . . . .	554
7.205.3.2 Create . . . . .	554
7.205.3.3 Signal . . . . .	555
7.205.3.4 TryWait . . . . .	555
7.205.3.5 Wait . . . . .	555
7.205.3.6 Wait . . . . .	556
7.206 OsclSendMethod Class Reference . . . . .	557
7.206.1 Constructor & Destructor Documentation . . . . .	557
7.206.1.1 ~OsclSendMethod . . . . .	557
7.206.2 Member Function Documentation . . . . .	557
7.206.2.1 GetSendData . . . . .	557
7.206.2.2 NewL . . . . .	557
7.206.2.3 Send . . . . .	557
7.206.2.4 SendRequest . . . . .	558
7.207 OsclSendRequest Class Reference . . . . .	559
7.207.1 Constructor & Destructor Documentation . . . . .	559
7.207.1.1 OsclSendRequest . . . . .	559
7.207.2 Member Function Documentation . . . . .	559
7.207.2.1 GetSendData . . . . .	559
7.207.2.2 Send . . . . .	559
7.207.2.3 Success . . . . .	559
7.208 OsclSendToMethod Class Reference . . . . .	560
7.208.1 Constructor & Destructor Documentation . . . . .	560
7.208.1.1 ~OsclSendToMethod . . . . .	560
7.208.2 Member Function Documentation . . . . .	560
7.208.2.1 GetSendData . . . . .	560
7.208.2.2 NewL . . . . .	560

---

7.208.2.3 SendTo . . . . .	560
7.208.2.4 SendToRequest . . . . .	561
7.209 OsclSendToRequest Class Reference . . . . .	562
7.209.1 Detailed Description . . . . .	562
7.209.2 Constructor & Destructor Documentation . . . . .	562
7.209.2.1 OsclSendToRequest . . . . .	562
7.209.3 Member Function Documentation . . . . .	562
7.209.3.1 GetSendData . . . . .	562
7.209.3.2 SendTo . . . . .	562
7.209.3.3 Success . . . . .	562
7.210 OsclSharedPtr< TheClass > Class Template Reference . . . . .	563
7.210.1 Detailed Description . . . . .	564
7.210.2 Constructor & Destructor Documentation . . . . .	564
7.210.2.1 OsclSharedPtr . . . . .	564
7.211 OsclShutdownMethod Class Reference . . . . .	565
7.211.1 Constructor & Destructor Documentation . . . . .	565
7.211.1.1 ~OsclShutdownMethod . . . . .	565
7.211.2 Member Function Documentation . . . . .	565
7.211.2.1 NewL . . . . .	565
7.211.2.2 Shutdown . . . . .	565
7.211.2.3 ShutdownRequest . . . . .	565
7.212 OsclShutdownRequest Class Reference . . . . .	566
7.212.1 Detailed Description . . . . .	566
7.212.2 Constructor & Destructor Documentation . . . . .	566
7.212.2.1 OsclShutdownRequest . . . . .	566
7.212.3 Member Function Documentation . . . . .	566
7.212.3.1 Shutdown . . . . .	566
7.213 OsclSingletonEx< T, ID, Registry > Class Template Reference . . . . .	567
7.213.1 Constructor & Destructor Documentation . . . . .	567
7.213.1.1 OsclSingletonEx . . . . .	567
7.213.1.2 ~OsclSingletonEx . . . . .	567
7.213.2 Member Function Documentation . . . . .	567
7.213.2.1 operator* . . . . .	567
7.213.2.2 operator-> . . . . .	567
7.213.2.3 set . . . . .	568
7.213.3 Field Documentation . . . . .	568

7.213.3.1 <code>_Ptr</code> . . . . .	568
7.214 <code>OsclSingletonRegistryEx</code> Class Reference . . . . .	569
7.214.1 Member Function Documentation . . . . .	569
7.214.1.1 <code>getInstance</code> . . . . .	569
7.214.1.2 <code>lockAndGetInstance</code> . . . . .	569
7.214.1.3 <code>registerInstance</code> . . . . .	569
7.214.1.4 <code>registerInstanceAndUnlock</code> . . . . .	569
7.215 <code>OsclSocketI</code> Class Reference . . . . .	570
7.215.1 Detailed Description . . . . .	571
7.215.2 Constructor & Destructor Documentation . . . . .	571
7.215.2.1 <code>~OsclSocketI</code> . . . . .	571
7.215.3 Member Function Documentation . . . . .	571
7.215.3.1 <code>Accept</code> . . . . .	571
7.215.3.2 <code>Bind</code> . . . . .	571
7.215.3.3 <code>Close</code> . . . . .	571
7.215.3.4 <code>Connect</code> . . . . .	571
7.215.3.5 <code>GetPeerName</code> . . . . .	572
7.215.3.6 <code>Join</code> . . . . .	572
7.215.3.7 <code>Listen</code> . . . . .	572
7.215.3.8 <code>Logger</code> . . . . .	572
7.215.3.9 <code>MakeAddr</code> . . . . .	572
7.215.3.10 <code>MakeAddr</code> . . . . .	572
7.215.3.11 <code>IMakeMulticastGroupInformation</code> . . . . .	572
7.215.3.12 <code>IMakeMulticastGroupInformation</code> . . . . .	572
7.215.3.13 <code>NewL</code> . . . . .	572
7.215.3.14 <code>Open</code> . . . . .	572
7.215.3.15 <code>Open</code> . . . . .	572
7.215.3.16 <code>ProcessAccept</code> . . . . .	573
7.215.3.17 <code>ProcessConnect</code> . . . . .	573
7.215.3.18 <code>ProcessRecv</code> . . . . .	573
7.215.3.19 <code>ProcessRecvFrom</code> . . . . .	573
7.215.3.20 <code>ProcessSend</code> . . . . .	573
7.215.3.21 <code>ProcessSendTo</code> . . . . .	573
7.215.3.22 <code>ProcessShutdown</code> . . . . .	573
7.215.3.23 <code>Recv</code> . . . . .	573
7.215.3.24 <code>RecvFrom</code> . . . . .	573

7.215.3.25RecvFromSuccess . . . . .	573
7.215.3.26RecvSuccess . . . . .	573
7.215.3.27Send . . . . .	573
7.215.3.28SendSuccess . . . . .	573
7.215.3.29SendTo . . . . .	573
7.215.3.30SendToSuccess . . . . .	573
7.215.3.31SetRecvBufferSize . . . . .	574
7.215.3.32SetSockOpt . . . . .	574
7.215.3.33Shutdown . . . . .	574
7.215.3.34Socket . . . . .	574
7.215.3.35ThreadLogoff . . . . .	574
7.215.3.36ThreadLogon . . . . .	574
7.215.4 Friends And Related Function Documentation . . . . .	574
7.215.4.1 OsclAcceptRequest . . . . .	574
7.215.4.2 OsclConnectRequest . . . . .	574
7.215.4.3 OsclRecvFromRequest . . . . .	574
7.215.4.4 OsclRecvRequest . . . . .	574
7.215.4.5 OsclSendRequest . . . . .	574
7.215.4.6 OsclSendToRequest . . . . .	574
7.215.4.7 OsclShutdownRequest . . . . .	574
7.215.4.8 OsclTCPSocket . . . . .	574
7.215.4.9 OsclUDPSocket . . . . .	574
7.216 OsclSocketIBase Class Reference . . . . .	575
7.216.1 Detailed Description . . . . .	576
7.216.2 Constructor & Destructor Documentation . . . . .	576
7.216.2.1 ~OsclSocketIBase . . . . .	576
7.216.2.2 OsclSocketIBase . . . . .	576
7.216.3 Member Function Documentation . . . . .	576
7.216.3.1 Accept . . . . .	576
7.216.3.2 Bind . . . . .	576
7.216.3.3 BindAsync . . . . .	577
7.216.3.4 CancelAccept . . . . .	577
7.216.3.5 CancelBind . . . . .	577
7.216.3.6 CancelConnect . . . . .	577
7.216.3.7 CancelFxn . . . . .	577
7.216.3.8 CancelListen . . . . .	577

7.216.3.9 CancelRecv . . . . .	577
7.216.3.10 CancelRecvFrom . . . . .	577
7.216.3.11 lCancelSend . . . . .	577
7.216.3.12 CancelSendTo . . . . .	577
7.216.3.13 CancelShutdown . . . . .	577
7.216.3.14 Close . . . . .	577
7.216.3.15 Connect . . . . .	577
7.216.3.16 GetShutdown . . . . .	577
7.216.3.17 HasAsyncBind . . . . .	577
7.216.3.18 HasAsyncListen . . . . .	577
7.216.3.19 lOpen . . . . .	578
7.216.3.20 lJoin . . . . .	578
7.216.3.21 lListen . . . . .	578
7.216.3.22 ListenAsync . . . . .	578
7.216.3.23 Open . . . . .	578
7.216.3.24 Open . . . . .	578
7.216.3.25 Recv . . . . .	578
7.216.3.26 RecvFrom . . . . .	578
7.216.3.27 RecvFromSuccess . . . . .	578
7.216.3.28 RecvSuccess . . . . .	578
7.216.3.29 Send . . . . .	578
7.216.3.30 SendSuccess . . . . .	579
7.216.3.31 lSendTo . . . . .	579
7.216.3.32 SendToSuccess . . . . .	579
7.216.3.33 Shutdown . . . . .	579
7.216.4 Friends And Related Function Documentation . . . . .	579
7.216.4.1 OsclSocketMethod . . . . .	579
7.216.4.2 OsclSocketRequest . . . . .	579
7.216.4.3 OsclSocketRequestAO . . . . .	579
7.216.4.4 OsclTCPSocket . . . . .	579
7.216.4.5 OsclUDPSocket . . . . .	579
7.216.5 Field Documentation . . . . .	579
7.216.5.1 iAlloc . . . . .	579
7.216.5.2 iSocketServ . . . . .	579
7.217 OsclSocketMethod Class Reference . . . . .	580
7.217.1 Detailed Description . . . . .	581

---

7.217.2 Constructor & Destructor Documentation . . . . .	581
7.217.2.1 OsclSocketMethod . . . . .	581
7.217.2.2 ~OsclSocketMethod . . . . .	581
7.217.3 Member Function Documentation . . . . .	581
7.217.3.1 Abort . . . . .	581
7.217.3.2 AbortAll . . . . .	581
7.217.3.3 Alloc . . . . .	581
7.217.3.4 CancelMethod . . . . .	581
7.217.3.5 ConstructL . . . . .	582
7.217.3.6 MethodDone . . . . .	582
7.217.3.7 Run . . . . .	582
7.217.3.8 StartMethod . . . . .	582
7.217.3.9 ThreadLogoff . . . . .	582
7.217.3.10 ThreadLogon . . . . .	582
7.217.4 Field Documentation . . . . .	582
7.217.4.1 iContainer . . . . .	582
7.217.4.2 iSocketFxn . . . . .	582
7.217.4.3 iSocketRequestAO . . . . .	582
7.218 OsclSocketObserver Class Reference . . . . .	584
7.218.1 Detailed Description . . . . .	584
7.218.2 Constructor & Destructor Documentation . . . . .	584
7.218.2.1 ~OsclSocketObserver . . . . .	584
7.218.3 Member Function Documentation . . . . .	584
7.218.3.1 HandleSocketEvent . . . . .	584
7.219 OsclSocketRequestAO Class Reference . . . . .	585
7.219.1 Detailed Description . . . . .	586
7.219.2 Constructor & Destructor Documentation . . . . .	586
7.219.2.1 OsclSocketRequestAO . . . . .	586
7.219.2.2 ~OsclSocketRequestAO . . . . .	586
7.219.3 Member Function Documentation . . . . .	586
7.219.3.1 Abort . . . . .	586
7.219.3.2 Alloc . . . . .	586
7.219.3.3 CleanupParam . . . . .	586
7.219.3.4 ConstructL . . . . .	587
7.219.3.5 DoCancel . . . . .	587
7.219.3.6 GetSocketError . . . . .	587

7.219.3.7 Id . . . . .	587
7.219.3.8 NewRequest . . . . .	587
7.219.3.9 RequestDone . . . . .	587
7.219.3.10 Run . . . . .	587
7.219.3.11 ISocketI . . . . .	587
7.219.3.12 ISocketObserver . . . . .	588
7.219.3.13 Success . . . . .	588
7.219.4 Friends And Related Function Documentation . . . . .	588
7.219.4.1 OsclSocketI . . . . .	588
7.219.4.2 OsclSocketMethod . . . . .	588
7.219.4.3 OsclSocketRequest . . . . .	588
7.219.5 Field Documentation . . . . .	588
7.219.5.1 iContainer . . . . .	588
7.219.5.2 iParam . . . . .	588
7.219.5.3 iParamSize . . . . .	588
7.219.5.4 iSocketError . . . . .	588
7.220 OsclSocketServ Class Reference . . . . .	589
7.220.1 Member Function Documentation . . . . .	589
7.220.1.1 NewL . . . . .	589
7.221 OsclSocketServI Class Reference . . . . .	590
7.221.1 Detailed Description . . . . .	590
7.221.2 Member Function Documentation . . . . .	590
7.221.2.1 Close . . . . .	590
7.221.2.2 Connect . . . . .	591
7.221.2.3 IsServerThread . . . . .	591
7.221.2.4 NewL . . . . .	591
7.221.3 Friends And Related Function Documentation . . . . .	591
7.221.3.1 LoopbackSocket . . . . .	591
7.221.3.2 OsclDNSI . . . . .	591
7.221.3.3 OsclSocketI . . . . .	591
7.221.3.4 OsclSocketRequest . . . . .	591
7.221.3.5 OsclSocketServ . . . . .	591
7.221.3.6 OsclSocketServRequestList . . . . .	591
7.221.3.7 OsclTCPSocketI . . . . .	591
7.221.3.8 OsclUDPSocketI . . . . .	591
7.222 OsclSocketServIBase Class Reference . . . . .	592

---

7.222.1 Detailed Description . . . . .	592
7.222.2 Member Enumeration Documentation . . . . .	592
7.222.2.1 TSocketServState . . . . .	592
7.222.3 Constructor & Destructor Documentation . . . . .	593
7.222.3.1 ~OsclSocketServIBase . . . . .	593
7.222.3.2 OsclSocketServIBase . . . . .	593
7.222.4 Member Function Documentation . . . . .	593
7.222.4.1 Close . . . . .	593
7.222.4.2 Connect . . . . .	593
7.222.4.3 IsServConnected . . . . .	593
7.222.4.4 State . . . . .	593
7.222.5 Field Documentation . . . . .	593
7.222.5.1 iAlloc . . . . .	593
7.222.5.2 iLogger . . . . .	593
7.222.5.3 iServerError . . . . .	593
7.222.5.4 iServState . . . . .	593
7.223 OsclSocketServRequestList Class Reference . . . . .	595
7.223.1 Detailed Description . . . . .	595
7.223.2 Constructor & Destructor Documentation . . . . .	595
7.223.2.1 OsclSocketServRequestList . . . . .	595
7.223.3 Member Function Documentation . . . . .	595
7.223.3.1 Add . . . . .	595
7.223.3.2 Close . . . . .	595
7.223.3.3 Open . . . . .	595
7.223.3.4 Remove . . . . .	595
7.223.3.5 StartCancel . . . . .	596
7.223.3.6 WaitOnRequests . . . . .	596
7.223.3.7 Wakeup . . . . .	596
7.223.4 Friends And Related Function Documentation . . . . .	596
7.223.4.1 OsclSocketServI . . . . .	596
7.224 OsclSocketServRequestQElem Class Reference . . . . .	597
7.224.1 Constructor & Destructor Documentation . . . . .	597
7.224.1.1 OsclSocketServRequestQElem . . . . .	597
7.224.2 Field Documentation . . . . .	597
7.224.2.1 iCancel . . . . .	597
7.224.2.2 iSelect . . . . .	597

7.224.2.3 iSocketRequest . . . . .	597
7.225 OsclSocketTOS Class Reference . . . . .	598
7.225.1 Member Enumeration Documentation . . . . .	598
7.225.1.1 TPVServicePrecedence . . . . .	598
7.225.1.2 TPVServicePriority . . . . .	598
7.225.2 Constructor & Destructor Documentation . . . . .	599
7.225.2.1 OsclSocketTOS . . . . .	599
7.225.3 Member Function Documentation . . . . .	599
7.225.3.1 ClearTOS . . . . .	599
7.225.3.2 GetTOS . . . . .	599
7.225.3.3 SetPrecedence . . . . .	599
7.225.3.4 SetPriority . . . . .	599
7.226 OsclTCPSocket Class Reference . . . . .	600
7.226.1 Detailed Description . . . . .	601
7.226.2 Member Function Documentation . . . . .	601
7.226.2.1 NewL . . . . .	601
7.227 OsclTCPSocketI Class Reference . . . . .	602
7.227.1 Detailed Description . . . . .	602
7.227.2 Constructor & Destructor Documentation . . . . .	603
7.227.2.1 ~OsclTCPSocketI . . . . .	603
7.227.3 Member Function Documentation . . . . .	603
7.227.3.1 Accept . . . . .	603
7.227.3.2 BindAsync . . . . .	603
7.227.3.3 CancelAccept . . . . .	603
7.227.3.4 CancelBind . . . . .	603
7.227.3.5 CancelConnect . . . . .	603
7.227.3.6 CancelListen . . . . .	603
7.227.3.7 CancelRecv . . . . .	603
7.227.3.8 CancelSend . . . . .	603
7.227.3.9 CancelShutdown . . . . .	603
7.227.3.10 Close . . . . .	604
7.227.3.11 Connect . . . . .	604
7.227.3.12 GetAcceptedSocketL . . . . .	604
7.227.3.13 GetRecvData . . . . .	604
7.227.3.14 GetSendData . . . . .	604
7.227.3.15 Listen . . . . .	604

7.227.3.16 ListenAsync . . . . .	604
7.227.3.17 NewL . . . . .	604
7.227.3.18 Recv . . . . .	604
7.227.3.19 Send . . . . .	604
7.227.3.20 Shutdown . . . . .	605
7.227.3.21 ThreadLogoff . . . . .	605
7.227.3.22 ThreadLogon . . . . .	605
7.228 OsclThread Class Reference . . . . .	606
7.228.1 Detailed Description . . . . .	606
7.228.2 Constructor & Destructor Documentation . . . . .	606
7.228.2.1 OsclThread . . . . .	606
7.228.2.2 ~OsclThread . . . . .	606
7.228.3 Member Function Documentation . . . . .	606
7.228.3.1 CanTerminate . . . . .	606
7.228.3.2 CompareId . . . . .	607
7.228.3.3 Create . . . . .	607
7.228.3.4 Exit . . . . .	607
7.228.3.5 GetId . . . . .	608
7.228.3.6 GetPriority . . . . .	608
7.228.3.7 Resume . . . . .	608
7.228.3.8 SetPriority . . . . .	608
7.228.3.9 SleepMillisecc . . . . .	609
7.228.3.10 Suspend . . . . .	609
7.228.3.11 Terminate . . . . .	609
7.229 OsclThreadLock Class Reference . . . . .	610
7.229.1 Detailed Description . . . . .	610
7.229.2 Constructor & Destructor Documentation . . . . .	610
7.229.2.1 OsclThreadLock . . . . .	610
7.229.2.2 ~OsclThreadLock . . . . .	610
7.229.3 Member Function Documentation . . . . .	610
7.229.3.1 Lock . . . . .	610
7.229.3.2 Unlock . . . . .	610
7.230 OsclTickCount Class Reference . . . . .	611
7.230.1 Detailed Description . . . . .	611
7.230.2 Member Function Documentation . . . . .	611
7.230.2.1 MsecToTicks . . . . .	611

7.230.2.2 TickCount . . . . .	611
7.230.2.3 TickCountFrequency . . . . .	611
7.230.2.4 TickCountPeriod . . . . .	612
7.230.2.5 TicksToMsec . . . . .	612
7.231 OsclTimer< Alloc > Class Template Reference . . . . .	613
7.231.1 Member Typedef Documentation . . . . .	614
7.231.1.1 callback_timer_type . . . . .	614
7.231.2 Constructor & Destructor Documentation . . . . .	614
7.231.2.1 OsclTimer . . . . .	614
7.231.2.2 ~OsclTimer . . . . .	614
7.231.3 Member Function Documentation . . . . .	614
7.231.3.1 Cancel . . . . .	614
7.231.3.2 Clear . . . . .	614
7.231.3.3 Request . . . . .	615
7.231.3.4 SetExactFrequency . . . . .	615
7.231.3.5 SetFrequency . . . . .	615
7.231.3.6 SetObserver . . . . .	615
7.231.3.7 TimerBaseElapsed . . . . .	616
7.231.4 Friends And Related Function Documentation . . . . .	616
7.231.4.1 CallbackTimer< Alloc > . . . . .	616
7.232 OsclTimerCompare Class Reference . . . . .	617
7.232.1 Member Function Documentation . . . . .	617
7.232.1.1 compare . . . . .	617
7.233 OsclTimerObject Class Reference . . . . .	618
7.233.1 Detailed Description . . . . .	619
7.233.2 Constructor & Destructor Documentation . . . . .	619
7.233.2.1 OsclTimerObject . . . . .	619
7.233.2.2 ~OsclTimerObject . . . . .	619
7.233.3 Member Function Documentation . . . . .	619
7.233.3.1 AddToScheduler . . . . .	619
7.233.3.2 After . . . . .	619
7.233.3.3 Cancel . . . . .	619
7.233.3.4 DoCancel . . . . .	620
7.233.3.5 IsBusy . . . . .	620
7.233.3.6 Priority . . . . .	620
7.233.3.7 RemoveFromScheduler . . . . .	620

---

7.233.3.8 RunError . . . . .	620
7.233.3.9 RunIfNotReady . . . . .	620
7.233.3.10 SetBusy . . . . .	621
7.233.3.11 SetStatus . . . . .	621
7.233.3.12 Status . . . . .	621
7.233.3.13 StatusRef . . . . .	621
7.234 OsclTimerObserver Class Reference . . . . .	622
7.234.1 Detailed Description . . . . .	622
7.234.2 Constructor & Destructor Documentation . . . . .	622
7.234.2.1 ~OsclTimerObserver . . . . .	622
7.234.3 Member Function Documentation . . . . .	622
7.234.3.1 TimeoutOccurred . . . . .	622
7.235 OsclTimerQ Class Reference . . . . .	623
7.235.1 Member Function Documentation . . . . .	623
7.235.1.1 Add . . . . .	623
7.235.1.2 Construct . . . . .	623
7.235.1.3 IsIn . . . . .	623
7.235.1.4 Pop . . . . .	623
7.235.1.5 PopTop . . . . .	623
7.235.1.6 Remove . . . . .	623
7.235.1.7 Top . . . . .	623
7.236 OsclTLS< T, ID, Registry > Class Template Reference . . . . .	624
7.236.1 Constructor & Destructor Documentation . . . . .	624
7.236.1.1 OsclTLS . . . . .	624
7.236.1.2 ~OsclTLS . . . . .	624
7.236.2 Member Function Documentation . . . . .	624
7.236.2.1 operator* . . . . .	624
7.236.2.2 operator-> . . . . .	624
7.236.2.3 set . . . . .	625
7.236.3 Field Documentation . . . . .	625
7.236.3.1 _Ptr . . . . .	625
7.237 OsclTLSEEx< T, ID, Registry > Class Template Reference . . . . .	626
7.237.1 Constructor & Destructor Documentation . . . . .	626
7.237.1.1 OsclTLSEEx . . . . .	626
7.237.1.2 ~OsclTLSEEx . . . . .	626
7.237.2 Member Function Documentation . . . . .	626

7.237.2.1 operator* . . . . .	626
7.237.2.2 operator-> . . . . .	626
7.237.2.3 set . . . . .	627
7.237.3 Field Documentation . . . . .	627
7.237.3.1 _Ptr . . . . .	627
7.238 OsclTLSRegistry Class Reference . . . . .	628
7.238.1 Member Function Documentation . . . . .	628
7.238.1.1 getInstance . . . . .	628
7.238.1.2 registerInstance . . . . .	628
7.238.2 Friends And Related Function Documentation . . . . .	628
7.238.2.1 OsclBase . . . . .	628
7.239 OsclTLSRegistryEx Class Reference . . . . .	629
7.239.1 Member Function Documentation . . . . .	629
7.239.1.1 getInstance . . . . .	629
7.239.1.2 registerInstance . . . . .	629
7.240 OsclTrapItem Class Reference . . . . .	630
7.240.1 Constructor & Destructor Documentation . . . . .	630
7.240.1.1 OsclTrapItem . . . . .	630
7.240.1.2 OsclTrapItem . . . . .	630
7.240.2 Friends And Related Function Documentation . . . . .	630
7.240.2.1 OsclTrapStack . . . . .	630
7.240.2.2 OsclTrapStackItem . . . . .	630
7.241 OsclTrapStack Class Reference . . . . .	631
7.241.1 Detailed Description . . . . .	631
7.241.2 Friends And Related Function Documentation . . . . .	631
7.241.2.1 OsclError . . . . .	631
7.241.2.2 OsclErrorTrap . . . . .	631
7.241.2.3 OsclErrorTrapImp . . . . .	631
7.242 OsclTrapStackItem Class Reference . . . . .	632
7.242.1 Detailed Description . . . . .	632
7.242.2 Constructor & Destructor Documentation . . . . .	632
7.242.2.1 OsclTrapStackItem . . . . .	632
7.242.2.2 OsclTrapStackItem . . . . .	632
7.242.2.3 OsclTrapStackItem . . . . .	632
7.242.2.4 OsclTrapStackItem . . . . .	632
7.242.3 Field Documentation . . . . .	632

---

7.242.3.1 iCBase . . . . .	632
7.242.3.2 iNext . . . . .	632
7.242.3.3 iTAny . . . . .	633
7.242.3.4 iTrapOperation . . . . .	633
7.243 OsclUDPSocket Class Reference . . . . .	634
7.243.1 Detailed Description . . . . .	635
7.243.2 Member Function Documentation . . . . .	635
7.243.2.1 NewL . . . . .	635
7.244 OsclUDPSocketI Class Reference . . . . .	636
7.244.1 Detailed Description . . . . .	636
7.244.2 Constructor & Destructor Documentation . . . . .	637
7.244.2.1 ~OsclUDPSocketI . . . . .	637
7.244.3 Member Function Documentation . . . . .	637
7.244.3.1 BindAsync . . . . .	637
7.244.3.2 CancelBind . . . . .	637
7.244.3.3 CancelRecvFrom . . . . .	637
7.244.3.4 CancelSendTo . . . . .	637
7.244.3.5 Close . . . . .	637
7.244.3.6 GetRecvData . . . . .	637
7.244.3.7 GetSendData . . . . .	637
7.244.3.8 JoinMulticastGroup . . . . .	638
7.244.3.9 NewL . . . . .	638
7.244.3.10 RecvFrom . . . . .	638
7.244.3.11 SendTo . . . . .	638
7.244.3.12 SetMulticastTTL . . . . .	638
7.244.3.13 ThreadLogoff . . . . .	638
7.244.3.14 ThreadLogon . . . . .	638
7.245 OsclUuid Struct Reference . . . . .	639
7.245.1 Detailed Description . . . . .	639
7.245.2 Constructor & Destructor Documentation . . . . .	639
7.245.2.1 OsclUuid . . . . .	639
7.245.2.2 OsclUuid . . . . .	639
7.245.2.3 OsclUuid . . . . .	639
7.245.3 Member Function Documentation . . . . .	639
7.245.3.1 operator!= . . . . .	639
7.245.3.2 operator= . . . . .	639

7.245.3.3 operator== . . . . .	640
7.245.4 Field Documentation . . . . .	640
7.245.4.1 data1 . . . . .	640
7.245.4.2 data2 . . . . .	640
7.245.4.3 data3 . . . . .	640
7.245.4.4 data4 . . . . .	640
7.246 PVActiveBase Class Reference . . . . .	641
7.246.1 Detailed Description . . . . .	642
7.246.2 Constructor & Destructor Documentation . . . . .	642
7.246.2.1 PVActiveBase . . . . .	642
7.246.2.2 ~PVActiveBase . . . . .	642
7.246.3 Member Function Documentation . . . . .	642
7.246.3.1 Activate . . . . .	642
7.246.3.2 AddToScheduler . . . . .	642
7.246.3.3 Cancel . . . . .	642
7.246.3.4 Destroy . . . . .	642
7.246.3.5 DoCancel . . . . .	642
7.246.3.6 IsAdded . . . . .	642
7.246.3.7 IsInAnyQ . . . . .	642
7.246.3.8 RemoveFromScheduler . . . . .	643
7.246.3.9 Run . . . . .	643
7.246.3.10 RunError . . . . .	643
7.246.4 Friends And Related Function Documentation . . . . .	644
7.246.4.1 OsclActiveObject . . . . .	644
7.246.4.2 OsclExecScheduler . . . . .	644
7.246.4.3 OsclReadyCompare . . . . .	644
7.246.4.4 OsclReadyQ . . . . .	644
7.246.4.5 OsclReadySetPosition . . . . .	644
7.246.4.6 OsclSchedulerCommonBase . . . . .	644
7.246.4.7 OsclTimerObject . . . . .	644
7.246.5 Field Documentation . . . . .	644
7.246.5.1 iAddedNum . . . . .	644
7.246.5.2 iBusy . . . . .	644
7.246.5.3 iName . . . . .	644
7.246.5.4 iPVReadyQLink . . . . .	644
7.246.5.5 iStatus . . . . .	644

---

7.246.5.6 iThreadContext . . . . .	644
7.247PVLogger Class Reference . . . . .	645
7.247.1 Member Typedef Documentation . . . . .	646
7.247.1.1 alloc_type . . . . .	646
7.247.1.2 filter_status_type . . . . .	646
7.247.1.3 log_level_type . . . . .	646
7.247.1.4 message_id_type . . . . .	646
7.247.2 Constructor & Destructor Documentation . . . . .	646
7.247.2.1 PVLogger . . . . .	646
7.247.2.2 ~PVLogger . . . . .	646
7.247.3 Member Function Documentation . . . . .	646
7.247.3.1 AddAppender . . . . .	646
7.247.3.2 AddFilter . . . . .	647
7.247.3.3 Cleanup . . . . .	647
7.247.3.4 DisableAppenderInheritance . . . . .	647
7.247.3.5 GetLoggerObject . . . . .	647
7.247.3.6 GetLogLevel . . . . .	648
7.247.3.7 GetNumAppenders . . . . .	648
7.247.3.8 GetParent . . . . .	648
7.247.3.9 Init . . . . .	648
7.247.3.10 IsActive . . . . .	648
7.247.3.11 LogMsgBuffers . . . . .	648
7.247.3.12 LogMsgBuffersV . . . . .	649
7.247.3.13 LogMsgString . . . . .	649
7.247.3.14 LogMsgStringV . . . . .	649
7.247.3.15 RemoveAppender . . . . .	650
7.247.3.16 SetLogLevel . . . . .	650
7.247.3.17 SetLogLevelAndPropagate . . . . .	650
7.247.3.18 SetParent . . . . .	651
7.247.4 Friends And Related Function Documentation . . . . .	651
7.247.4.1 PVLoggerRegistry . . . . .	651
7.248PVLoggerAppender Class Reference . . . . .	652
7.248.1 Detailed Description . . . . .	652
7.248.2 Member Typedef Documentation . . . . .	652
7.248.2.1 message_id_type . . . . .	652
7.248.3 Constructor & Destructor Documentation . . . . .	652

7.248.3.1 ~PVLoggerAppender . . . . .	652
7.248.4 Member Function Documentation . . . . .	652
7.248.4.1 AppendBuffers . . . . .	652
7.248.4.2 AppendString . . . . .	652
7.249PVLoggerFilter Class Reference . . . . .	653
7.249.1 Detailed Description . . . . .	653
7.249.2 Member Typedef Documentation . . . . .	653
7.249.2.1 filter_status_type . . . . .	653
7.249.2.2 log_level_type . . . . .	653
7.249.2.3 message_id_type . . . . .	653
7.249.3 Constructor & Destructor Documentation . . . . .	654
7.249.3.1 ~PVLoggerFilter . . . . .	654
7.249.4 Member Function Documentation . . . . .	654
7.249.4.1 FilterOpaqueMessge . . . . .	654
7.249.4.2 FilterString . . . . .	654
7.250PVLoggerLayout Class Reference . . . . .	655
7.250.1 Detailed Description . . . . .	655
7.250.2 Member Typedef Documentation . . . . .	655
7.250.2.1 message_id_type . . . . .	655
7.250.3 Constructor & Destructor Documentation . . . . .	655
7.250.3.1 ~PVLoggerLayout . . . . .	655
7.250.4 Member Function Documentation . . . . .	655
7.250.4.1 FormatOpaqueMessage . . . . .	655
7.250.4.2 FormatString . . . . .	655
7.251PVLoggerRegistry Class Reference . . . . .	657
7.251.1 Detailed Description . . . . .	657
7.251.2 Member Typedef Documentation . . . . .	657
7.251.2.1 alloc_type . . . . .	657
7.251.2.2 log_level_type . . . . .	657
7.251.3 Constructor & Destructor Documentation . . . . .	657
7.251.3.1 PVLoggerRegistry . . . . .	657
7.251.3.2 ~PVLoggerRegistry . . . . .	657
7.251.4 Member Function Documentation . . . . .	658
7.251.4.1 CreatePVLogger . . . . .	658
7.251.4.2 GetPVLoggerObject . . . . .	658
7.251.4.3 GetPVLoggerRegistry . . . . .	658

---

7.251.4.4 SetNodeLogLevelExplicit . . . . .	658
7.251.4.5 SetNodeLogLevelExplicit . . . . .	659
7.252 PVSchedulerStopper Class Reference . . . . .	660
7.252.1 Detailed Description . . . . .	660
7.252.2 Constructor & Destructor Documentation . . . . .	660
7.252.2.1 PVSchedulerStopper . . . . .	660
7.252.2.2 ~PVSchedulerStopper . . . . .	660
7.253 PVSockBufRecv Class Reference . . . . .	661
7.253.1 Constructor & Destructor Documentation . . . . .	661
7.253.1.1 PVSockBufRecv . . . . .	661
7.253.1.2 PVSockBufRecv . . . . .	661
7.253.1.3 PVSockBufRecv . . . . .	661
7.253.2 Field Documentation . . . . .	661
7.253.2.1 iLen . . . . .	661
7.253.2.2 iMaxLen . . . . .	661
7.253.2.3 iPtr . . . . .	661
7.254 PVSockBufSend Class Reference . . . . .	662
7.254.1 Constructor & Destructor Documentation . . . . .	662
7.254.1.1 PVSockBufSend . . . . .	662
7.254.1.2 PVSockBufSend . . . . .	662
7.254.1.3 PVSockBufSend . . . . .	662
7.254.2 Field Documentation . . . . .	662
7.254.2.1 iLen . . . . .	662
7.254.2.2 iPtr . . . . .	662
7.255 PVThreadContext Class Reference . . . . .	663
7.255.1 Constructor & Destructor Documentation . . . . .	663
7.255.1.1 PVThreadContext . . . . .	663
7.255.1.2 ~PVThreadContext . . . . .	663
7.255.2 Member Function Documentation . . . . .	663
7.255.2.1 EnterThreadContext . . . . .	663
7.255.2.2 ExitThreadContext . . . . .	663
7.255.2.3 Id . . . . .	663
7.255.2.4 IsSameThreadContext . . . . .	664
7.255.2.5 ThreadHasScheduler . . . . .	664
7.255.3 Friends And Related Function Documentation . . . . .	664
7.255.3.1 OsclActiveObject . . . . .	664

7.255.3.2 OsclCoeActiveScheduler . . . . .	664
7.255.3.3 OsclCoeActiveSchedulerBase . . . . .	664
7.255.3.4 OsclExecScheduler . . . . .	664
7.255.3.5 OsclExecSchedulerBase . . . . .	664
7.255.3.6 OsclExecSchedulerCommonBase . . . . .	664
7.255.3.7 OsclTimerObject . . . . .	664
7.255.3.8 PVActiveBase . . . . .	664
7.256Oscl_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference . . . . .	665
7.256.1 Member Typedef Documentation . . . . .	665
7.256.1.1 other . . . . .	665
7.257RecvFromParam Class Reference . . . . .	666
7.257.1 Constructor & Destructor Documentation . . . . .	666
7.257.1.1 RecvFromParam . . . . .	666
7.257.2 Field Documentation . . . . .	666
7.257.2.1 iAddr . . . . .	666
7.257.2.2 iBufRecv . . . . .	666
7.257.2.3 iFlags . . . . .	666
7.257.2.4 iMultiMaxLen . . . . .	666
7.257.2.5 iPacketLen . . . . .	666
7.257.2.6 iPacketSource . . . . .	666
7.258RecvParam Class Reference . . . . .	668
7.258.1 Constructor & Destructor Documentation . . . . .	668
7.258.1.1 RecvParam . . . . .	668
7.258.2 Field Documentation . . . . .	668
7.258.2.1 iBufRecv . . . . .	668
7.258.2.2 iFlags . . . . .	668
7.259SendParam Class Reference . . . . .	669
7.259.1 Detailed Description . . . . .	669
7.259.2 Constructor & Destructor Documentation . . . . .	669
7.259.2.1 SendParam . . . . .	669
7.259.3 Field Documentation . . . . .	669
7.259.3.1 iBufSend . . . . .	669
7.259.3.2 iFlags . . . . .	669
7.259.3.3 iXferLen . . . . .	669
7.260SendToParam Class Reference . . . . .	670
7.260.1 Constructor & Destructor Documentation . . . . .	670

---

7.260.1.1 SendToParam . . . . .	670
7.260.1.2 ~SendToParam . . . . .	670
7.260.2 Field Documentation . . . . .	670
7.260.2.1 iAddr . . . . .	670
7.260.2.2 iBufSend . . . . .	670
7.260.2.3 iFlags . . . . .	670
7.260.2.4 iXferLen . . . . .	670
7.261 ShutdownParam Class Reference . . . . .	671
7.261.1 Constructor & Destructor Documentation . . . . .	671
7.261.1.1 ShutdownParam . . . . .	671
7.261.2 Field Documentation . . . . .	671
7.261.2.1 iHow . . . . .	671
7.262 SocketRequestParam Class Reference . . . . .	672
7.262.1 Detailed Description . . . . .	672
7.262.2 Constructor & Destructor Documentation . . . . .	673
7.262.2.1 SocketRequestParam . . . . .	673
7.262.3 Field Documentation . . . . .	673
7.262.3.1 iFxn . . . . .	673
7.263 StrCSumPtrLen Struct Reference . . . . .	674
7.263.1 Detailed Description . . . . .	674
7.263.2 Member Typedef Documentation . . . . .	675
7.263.2.1 CheckSumType . . . . .	675
7.263.3 Constructor & Destructor Documentation . . . . .	675
7.263.3.1 StrCSumPtrLen . . . . .	675
7.263.3.2 StrCSumPtrLen . . . . .	675
7.263.3.3 StrCSumPtrLen . . . . .	675
7.263.3.4 StrCSumPtrLen . . . . .	675
7.263.3.5 StrCSumPtrLen . . . . .	675
7.263.4 Member Function Documentation . . . . .	675
7.263.4.1 getCheckSum . . . . .	675
7.263.4.2 isCIEquivalentTo . . . . .	675
7.263.4.3 operator!= . . . . .	675
7.263.4.4 operator= . . . . .	675
7.263.4.5 operator= . . . . .	675
7.263.4.6 operator= . . . . .	676
7.263.4.7 operator== . . . . .	676

7.263.4.8 setCheckSum . . . . .	676
7.263.4.9 setPtrLen . . . . .	676
7.263.5 Field Documentation . . . . .	676
7.263.5.1 checkSum . . . . .	676
7.264 StrPtrLen Struct Reference . . . . .	677
7.264.1 Detailed Description . . . . .	677
7.264.2 Constructor & Destructor Documentation . . . . .	677
7.264.2.1 StrPtrLen . . . . .	677
7.264.2.2 StrPtrLen . . . . .	678
7.264.2.3 StrPtrLen . . . . .	678
7.264.2.4 StrPtrLen . . . . .	678
7.264.3 Member Function Documentation . . . . .	678
7.264.3.1 c_str . . . . .	678
7.264.3.2 isCIEquivalentTo . . . . .	678
7.264.3.3 isCIPrefixOf . . . . .	678
7.264.3.4 isLetter . . . . .	678
7.264.3.5 length . . . . .	678
7.264.3.6 operator!= . . . . .	678
7.264.3.7 operator= . . . . .	678
7.264.3.8 operator= . . . . .	678
7.264.3.9 operator== . . . . .	678
7.264.3.10 setPtrLen . . . . .	679
7.264.3.11 lsize . . . . .	679
7.264.4 Field Documentation . . . . .	679
7.264.4.1 len . . . . .	679
7.264.4.2 ptr . . . . .	679
7.265 TimeValue Class Reference . . . . .	680
7.265.1 Detailed Description . . . . .	682
7.265.2 Constructor & Destructor Documentation . . . . .	682
7.265.2.1 TimeValue . . . . .	682
7.265.2.2 TimeValue . . . . .	682
7.265.2.3 TimeValue . . . . .	682
7.265.2.4 TimeValue . . . . .	682
7.265.2.5 TimeValue . . . . .	682
7.265.2.6 TimeValue . . . . .	682
7.265.2.7 TimeValue . . . . .	683

---

7.265.3 Member Function Documentation . . . . .	683
7.265.3.1 get_ISO8601_str_time . . . . .	683
7.265.3.2 get_local_time . . . . .	683
7.265.3.3 get_pv8601_str_time . . . . .	683
7.265.3.4 get_rfc822_gmtime_str . . . . .	684
7.265.3.5 get_sec . . . . .	684
7.265.3.6 get_str_ctime . . . . .	684
7.265.3.7 get_timeval_ptr . . . . .	685
7.265.3.8 get_timevalue_in_usec . . . . .	685
7.265.3.9 get_usec . . . . .	685
7.265.3.10 is_zero . . . . .	685
7.265.3.11 is_zulu . . . . .	685
7.265.3.12 operator*= 7.265.3.13 operator+= . . . . .	685
7.265.3.14 operator+= . . . . .	685
7.265.3.15 operator-= . . . . .	685
7.265.3.16 operator-= . . . . .	685
7.265.3.17 operator= . . . . .	686
7.265.3.18 set_from_ntp_time . . . . .	686
7.265.3.19 set_to_current_time . . . . .	686
7.265.3.20 set_to_zero . . . . .	686
7.265.3.21 set_zulu . . . . .	686
7.265.3.22 to_msec . . . . .	686
7.265.4 Friends And Related Function Documentation . . . . .	686
7.265.4.1 NTPTime . . . . .	686
7.265.4.2 operator!= . . . . .	686
7.265.4.3 operator< . . . . .	686
7.265.4.4 operator<= . . . . .	686
7.265.4.5 operator== . . . . .	686
7.265.4.6 operator> . . . . .	686
7.265.4.7 operator>= . . . . .	686
7.266 TLSStorageOps Class Reference . . . . .	687
7.266.1 Member Function Documentation . . . . .	687
7.266.1.1 get_registry . . . . .	687
7.266.1.2 save_registry . . . . .	687
7.267 TReadyQueLink Class Reference . . . . .	688

7.267.1 Detailed Description . . . . .	688
7.267.2 Constructor & Destructor Documentation . . . . .	688
7.267.2.1 TReadyQueLink . . . . .	688
7.267.3 Field Documentation . . . . .	688
7.267.3.1 iAOPriority . . . . .	688
7.267.3.2 iIsIn . . . . .	688
7.267.3.3 iSeqNum . . . . .	688
7.267.3.4 iTimeQueuedTicks . . . . .	688
7.267.3.5 iTimeToRunTicks . . . . .	688
7.268Oscl_Map< Key, T, Alloc, Compare >::value_compare Class Reference . . . . .	690
7.268.1 Constructor & Destructor Documentation . . . . .	690
7.268.1.1 value_compare . . . . .	690
7.268.2 Member Function Documentation . . . . .	690
7.268.2.1 operator() . . . . .	690
7.268.3 Friends And Related Function Documentation . . . . .	691
7.268.3.1 Oscl_Map< Key, T, Alloc, Compare > . . . . .	691
7.268.4 Field Documentation . . . . .	691
7.268.4.1 comp . . . . .	691
7.269WStrPtrLen Struct Reference . . . . .	692
7.269.1 Detailed Description . . . . .	692
7.269.2 Constructor & Destructor Documentation . . . . .	692
7.269.2.1 WStrPtrLen . . . . .	692
7.269.2.2 WStrPtrLen . . . . .	692
7.269.2.3 WStrPtrLen . . . . .	692
7.269.2.4 WStrPtrLen . . . . .	692
7.269.3 Member Function Documentation . . . . .	692
7.269.3.1 c_str . . . . .	692
7.269.3.2 isCIEquivalentTo . . . . .	693
7.269.3.3 length . . . . .	693
7.269.3.4 operator!= . . . . .	693
7.269.3.5 operator= . . . . .	693
7.269.3.6 operator= . . . . .	693
7.269.3.7 operator== . . . . .	693
7.269.3.8 setPtrLen . . . . .	693
7.269.3.9 size . . . . .	693
7.269.4 Field Documentation . . . . .	693

---

7.269.4.1 len . . . . .	693
7.269.4.2 ptr . . . . .	693
<b>8 File Documentation</b>	<b>695</b>
8.1 oscl_aostatus.h File Reference . . . . .	695
8.1.1 Detailed Description . . . . .	695
8.2 oscl_assert.h File Reference . . . . .	696
8.2.1 Detailed Description . . . . .	696
8.3 oscl_base.h File Reference . . . . .	697
8.3.1 Detailed Description . . . . .	697
8.4 oscl_base_alloc.h File Reference . . . . .	698
8.4.1 Detailed Description . . . . .	698
8.5 oscl_base_macros.h File Reference . . . . .	699
8.5.1 Detailed Description . . . . .	699
8.6 oscl_bin_stream.h File Reference . . . . .	700
8.6.1 Detailed Description . . . . .	700
8.7 oscl_byte_order.h File Reference . . . . .	701
8.7.1 Detailed Description . . . . .	701
8.8 oscl_defalloc.h File Reference . . . . .	702
8.8.1 Detailed Description . . . . .	702
8.9 oscl_dll.h File Reference . . . . .	703
8.9.1 Detailed Description . . . . .	703
8.10 oscl_dns.h File Reference . . . . .	704
8.10.1 Detailed Description . . . . .	704
8.11 oscl_dns_gethostbyname.h File Reference . . . . .	705
8.12 oscl_dns_imp.h File Reference . . . . .	706
8.13 oscl_dns_imp_base.h File Reference . . . . .	707
8.14 oscl_dns_imp_pv.h File Reference . . . . .	708
8.15 oscl_dns_method.h File Reference . . . . .	709
8.16 oscl_dns_param.h File Reference . . . . .	710
8.16.1 Typedef Documentation . . . . .	710
8.16.1.1 TDNSRequestParamAllocator . . . . .	710
8.17 oscl_dns_request.h File Reference . . . . .	711
8.18 oscl_dns_tuneables.h File Reference . . . . .	712
8.19 oscl_double_list.h File Reference . . . . .	713
8.19.1 Detailed Description . . . . .	713
8.20 oscl_errno.h File Reference . . . . .	714

8.20.1	Detailed Description	714
8.21	oscl_error.h File Reference	715
8.21.1	Detailed Description	715
8.22	oscl_error_allocator.h File Reference	716
8.22.1	Detailed Description	716
8.23	oscl_error_codes.h File Reference	717
8.23.1	Detailed Description	717
8.24	oscl_error_imp.h File Reference	718
8.24.1	Detailed Description	718
8.25	oscl_error_imp_cppexceptions.h File Reference	719
8.25.1	Detailed Description	719
8.26	oscl_error_imp_fatalerror.h File Reference	720
8.26.1	Detailed Description	720
8.27	oscl_error_imp_jumps.h File Reference	721
8.27.1	Detailed Description	721
8.28	oscl_error_trapcleanup.h File Reference	722
8.28.1	Detailed Description	722
8.29	oscl_exception.h File Reference	723
8.29.1	Detailed Description	723
8.30	oscl_exclusive_ptr.h File Reference	724
8.30.1	Detailed Description	724
8.31	oscl_file_async_read.h File Reference	725
8.32	oscl_file_cache.h File Reference	726
8.32.1	Detailed Description	726
8.33	oscl_file_dir_utils.h File Reference	727
8.33.1	Detailed Description	728
8.34	oscl_file_find.h File Reference	729
8.34.1	Detailed Description	729
8.35	oscl_file_handle.h File Reference	730
8.35.1	Detailed Description	730
8.36	oscl_file_io.h File Reference	731
8.36.1	Detailed Description	731
8.37	oscl_file_manager.h File Reference	732
8.37.1	Detailed Description	732
8.38	oscl_file_native.h File Reference	733
8.38.1	Detailed Description	733

8.39 oscl_file_server.h File Reference . . . . .	734
8.39.1 Detailed Description . . . . .	734
8.40 oscl_file_stats.h File Reference . . . . .	735
8.40.1 Detailed Description . . . . .	735
8.41 oscl_file_types.h File Reference . . . . .	736
8.41.1 Detailed Description . . . . .	736
8.42 oscl_heapbase.h File Reference . . . . .	737
8.42.1 Detailed Description . . . . .	737
8.43 oscl_init.h File Reference . . . . .	738
8.43.1 Detailed Description . . . . .	738
8.44 oscl_int64_utils.h File Reference . . . . .	739
8.44.1 Typedef Documentation . . . . .	739
8.44.1.1 _OsclInteger64Transport . . . . .	739
8.45 oscl_ip_socket.h File Reference . . . . .	740
8.46 oscl_linked_list.h File Reference . . . . .	741
8.46.1 Detailed Description . . . . .	741
8.47 oscl_lock_base.h File Reference . . . . .	742
8.47.1 Detailed Description . . . . .	742
8.48 oscl_map.h File Reference . . . . .	743
8.48.1 Detailed Description . . . . .	743
8.49 oscl_math.h File Reference . . . . .	744
8.49.1 Detailed Description . . . . .	744
8.50 oscl_media_data.h File Reference . . . . .	745
8.50.1 Detailed Description . . . . .	745
8.51 oscl_media_status.h File Reference . . . . .	746
8.51.1 Detailed Description . . . . .	746
8.52 oscl_mem.h File Reference . . . . .	747
8.52.1 Detailed Description . . . . .	748
8.53 oscl_mem_audit.h File Reference . . . . .	749
8.53.1 Detailed Description . . . . .	750
8.54 oscl_mem_audit_internals.h File Reference . . . . .	751
8.54.1 Detailed Description . . . . .	751
8.55 oscl_mem_auto_ptr.h File Reference . . . . .	752
8.55.1 Detailed Description . . . . .	752
8.56 oscl_mem_basic_functions.h File Reference . . . . .	753
8.56.1 Detailed Description . . . . .	753

8.57 oscl_mem_inst.h File Reference . . . . .	754
8.57.1 Detailed Description . . . . .	754
8.58 oscl_mem_mempool.h File Reference . . . . .	755
8.58.1 Detailed Description . . . . .	755
8.59 oscl_mutex.h File Reference . . . . .	756
8.59.1 Detailed Description . . . . .	756
8.59.2 Typedef Documentation . . . . .	756
8.59.2.1 OsclNoYieldMutex . . . . .	756
8.60 oscl_namestring.h File Reference . . . . .	757
8.60.1 Detailed Description . . . . .	757
8.61 oscl_opaque_type.h File Reference . . . . .	758
8.61.1 Detailed Description . . . . .	758
8.62 oscl_pqueue.h File Reference . . . . .	759
8.62.1 Detailed Description . . . . .	759
8.63 oscl_procstatus.h File Reference . . . . .	760
8.64 oscl_queue.h File Reference . . . . .	761
8.64.1 Detailed Description . . . . .	761
8.65 oscl_rand.h File Reference . . . . .	762
8.65.1 Detailed Description . . . . .	762
8.66 oscl_refcounter.h File Reference . . . . .	763
8.66.1 Detailed Description . . . . .	763
8.67 oscl_refcounter_memfrag.h File Reference . . . . .	764
8.67.1 Detailed Description . . . . .	764
8.68 oscl_registry_access_client.h File Reference . . . . .	765
8.68.1 Detailed Description . . . . .	765
8.69 oscl_registry_client.h File Reference . . . . .	766
8.69.1 Detailed Description . . . . .	766
8.70 oscl_registry_client_impl.h File Reference . . . . .	767
8.70.1 Detailed Description . . . . .	767
8.71 oscl_registry_serv_impl.h File Reference . . . . .	768
8.71.1 Detailed Description . . . . .	768
8.72 oscl_registry_serv_impl_global.h File Reference . . . . .	769
8.73 oscl_registry_serv_impl_tls.h File Reference . . . . .	770
8.74 oscl_registry_types.h File Reference . . . . .	771
8.74.1 Detailed Description . . . . .	771
8.75 oscl_scheduler.h File Reference . . . . .	772

8.76	oscl_scheduler_ao.h File Reference . . . . .	773
8.76.1	Detailed Description . . . . .	773
8.77	oscl_scheduler_aobase.h File Reference . . . . .	774
8.77.1	Detailed Description . . . . .	774
8.78	oscl_scheduler_readyq.h File Reference . . . . .	775
8.78.1	Detailed Description . . . . .	775
8.79	oscl_scheduler_threadcontext.h File Reference . . . . .	776
8.79.1	Detailed Description . . . . .	776
8.80	oscl_scheduler_tuneables.h File Reference . . . . .	777
8.80.1	Detailed Description . . . . .	777
8.81	oscl_scheduler_types.h File Reference . . . . .	778
8.81.1	Detailed Description . . . . .	778
8.82	oscl_semaphore.h File Reference . . . . .	779
8.82.1	Detailed Description . . . . .	779
8.83	oscl_shared_ptr.h File Reference . . . . .	780
8.83.1	Detailed Description . . . . .	780
8.84	oscl_singleton.h File Reference . . . . .	781
8.84.1	Detailed Description . . . . .	781
8.85	oscl_snprintf.h File Reference . . . . .	782
8.85.1	Detailed Description . . . . .	782
8.86	oscl_socket.h File Reference . . . . .	783
8.86.1	Detailed Description . . . . .	783
8.87	oscl_socket_accept.h File Reference . . . . .	784
8.88	oscl_socket_bind.h File Reference . . . . .	785
8.89	oscl_socket_connect.h File Reference . . . . .	786
8.90	oscl_socket_imp.h File Reference . . . . .	787
8.91	oscl_socket_imp_base.h File Reference . . . . .	788
8.92	oscl_socket_imp_pv.h File Reference . . . . .	789
8.92.1	Define Documentation . . . . .	789
8.92.1.1	PVSOCK_ERR_BAD_PARAM . . . . .	789
8.92.1.2	PVSOCK_ERR_NOT_IMPLEMENTED . . . . .	789
8.92.1.3	PVSOCK_ERR_NOT_SUPPORTED . . . . .	789
8.92.1.4	PVSOCK_ERR_SERV_NOT_CONNECTED . . . . .	789
8.92.1.5	PVSOCK_ERR SOCK_NO_SERV . . . . .	789
8.92.1.6	PVSOCK_ERR SOCK_NOT_CONNECTED . . . . .	789
8.92.1.7	PVSOCK_ERR SOCK_NOT_OPEN . . . . .	789

8.93 oscl_socket_listen.h File Reference . . . . .	790
8.93.1 Define Documentation . . . . .	790
8.93.1.1 OSCL_SOCKET_LISTEN_H_INCLUDEDd . . . . .	790
8.94 oscl_socket_method.h File Reference . . . . .	791
8.94.1 Define Documentation . . . . .	791
8.94.1.1 MSEC_TO_MICROSEC . . . . .	791
8.95 oscl_socket_recv.h File Reference . . . . .	792
8.96 oscl_socket_recv_from.h File Reference . . . . .	793
8.97 oscl_socket_request.h File Reference . . . . .	794
8.98 oscl_socket_send.h File Reference . . . . .	795
8.99 oscl_socket_send_to.h File Reference . . . . .	796
8.100 oscl_socket_serv_imp.h File Reference . . . . .	797
8.101 oscl_socket_serv_imp_base.h File Reference . . . . .	798
8.102 oscl_socket_serv_imp_pv.h File Reference . . . . .	799
8.102.1 Define Documentation . . . . .	799
8.102.1.1 OSCL_EXCEPTSET_FLAG . . . . .	799
8.102.1.2 OSCL_READSET_FLAG . . . . .	799
8.102.1.3 OSCL_WRITESET_FLAG . . . . .	799
8.103 oscl_socket_serv_imp_reqlist.h File Reference . . . . .	800
8.104 oscl_socket_shutdown.h File Reference . . . . .	801
8.105 oscl_socket_stats.h File Reference . . . . .	802
8.105.1 Enumeration Type Documentation . . . . .	802
8.105.1.1 TOsclSocketServStatEvent . . . . .	802
8.105.1.2 TOsclSocketStatEvent . . . . .	802
8.106 oscl_socket_tuneables.h File Reference . . . . .	804
8.106.1 Define Documentation . . . . .	804
8.106.1.1 PV_OSCL_SOCKET_STATS_LOGGING . . . . .	804
8.106.1.2 PV_SOCKET_SERVER . . . . .	804
8.107 oscl_socket_types.h File Reference . . . . .	805
8.107.1 Define Documentation . . . . .	805
8.107.1.1 PVNETWORKADDRESS_LEN . . . . .	805
8.107.2 Enumeration Type Documentation . . . . .	805
8.107.2.1 TPVSocketEvent . . . . .	805
8.107.2.2 TPVSocketFxn . . . . .	806
8.107.2.3 TPVSocketOptionLevel . . . . .	806
8.107.2.4 TPVSocketOptionName . . . . .	806

8.107.2.5 TPVSocketShutdown . . . . .	806
8.108oscl_stdstring.h File Reference . . . . .	807
8.108.1 Detailed Description . . . . .	808
8.109oscl_str_ptr_len.h File Reference . . . . .	809
8.109.1 Detailed Description . . . . .	809
8.110oscl_string.h File Reference . . . . .	810
8.110.1 Detailed Description . . . . .	810
8.111oscl_string_containers.h File Reference . . . . .	811
8.111.1 Detailed Description . . . . .	811
8.112oscl_string_rep.h File Reference . . . . .	812
8.112.1 Detailed Description . . . . .	812
8.113oscl_string_uri.h File Reference . . . . .	813
8.113.1 Detailed Description . . . . .	813
8.114oscl_string_utf8.h File Reference . . . . .	814
8.114.1 Detailed Description . . . . .	814
8.115oscl_string_utils.h File Reference . . . . .	815
8.115.1 Detailed Description . . . . .	815
8.116oscl_string_xml.h File Reference . . . . .	816
8.116.1 Detailed Description . . . . .	816
8.117oscl_tagtree.h File Reference . . . . .	817
8.117.1 Detailed Description . . . . .	817
8.118oscl_tcp_socket.h File Reference . . . . .	818
8.119oscl_thread.h File Reference . . . . .	819
8.119.1 Detailed Description . . . . .	819
8.119.2 Typedef Documentation . . . . .	819
8.119.2.1 TOsclThreadFuncPtr . . . . .	819
8.119.3 Enumeration Type Documentation . . . . .	819
8.119.3.1 OsclThread_State . . . . .	819
8.119.3.2 OsclThreadPriority . . . . .	820
8.119.3.3 TOsclThreadTerminate . . . . .	820
8.120oscl_tickcount.h File Reference . . . . .	821
8.120.1 Detailed Description . . . . .	821
8.121oscl_time.h File Reference . . . . .	822
8.121.1 Detailed Description . . . . .	823
8.122oscl_timer.h File Reference . . . . .	824
8.123oscl_tls.h File Reference . . . . .	825

8.124oscl_tree.h File Reference . . . . .	826
8.124.1 Detailed Description . . . . .	826
8.125oscl_types.h File Reference . . . . .	827
8.125.1 Detailed Description . . . . .	827
8.126oscl_udp_socket.h File Reference . . . . .	828
8.127oscl_utf8conv.h File Reference . . . . .	829
8.127.1 Detailed Description . . . . .	829
8.128oscl_uuid.h File Reference . . . . .	830
8.128.1 Detailed Description . . . . .	830
8.128.2 Define Documentation . . . . .	830
8.128.2.1 BYTES_IN_UUID_ARRAY . . . . .	830
8.128.2.2 EMPTY_UUID . . . . .	830
8.128.3 Typedef Documentation . . . . .	830
8.128.3.1 OsclUid32 . . . . .	830
8.129oscl_uuid_utils.h File Reference . . . . .	831
8.129.1 Detailed Description . . . . .	831
8.129.2 Variable Documentation . . . . .	831
8.129.2.1 PV_CHAR_CLOSE_BRACKET . . . . .	831
8.129.2.2 PV_CHAR_COMMA . . . . .	831
8.130oscl_vector.h File Reference . . . . .	832
8.130.1 Detailed Description . . . . .	832
8.131osclconfig.h File Reference . . . . .	833
8.131.1 Detailed Description . . . . .	833
8.131.2 Define Documentation . . . . .	834
8.131.2.1 __TFS__ . . . . .	834
8.131.2.2 OSCL_ASSERT_ALWAYS . . . . .	834
8.131.2.3 OSCL_EXPORT_REF . . . . .	834
8.131.2.4 OSCL_HAS_ANDROID_FILE_IO_SUPPORT . . . . .	834
8.131.2.5 OSCL_HAS_ANDROID_SUPPORT . . . . .	834
8.131.2.6 OSCL_HAS_PACKED_STRUCT . . . . .	834
8.131.2.7 OSCL_HAS_PRAGMA_PACK . . . . .	834
8.131.2.8 OSCL_IMPORT_REF . . . . .	834
8.131.2.9 OSCL_NATIVE_UINT64_TYPE . . . . .	834
8.131.2.10 OSCL_PACKED_STRUCT_BEGIN . . . . .	834
8.131.2.11 OSCL_PACKED_STRUCT_END . . . . .	834
8.131.2.12 OSCL_PACKED_VAR . . . . .	834

8.131.2.13OSCL_RELEASE_BUILD . . . . .	834
8.131.2.14OSCL_TEMPLATED_DESTRUCTOR_CALL . . . . .	834
8.131.2.15OSCL_UNSIGNED_CONST . . . . .	834
8.132osclconfig_ansi_memory.h File Reference . . . . .	835
8.132.1 Detailed Description . . . . .	835
8.132.2 Define Documentation . . . . .	835
8.132.2.1 OSCL_HAS_ANSI_MEMORY_FUNCS . . . . .	835
8.132.3 Typedef Documentation . . . . .	835
8.132.3.1 oscl_memsize_t . . . . .	835
8.133osclconfig_check.h File Reference . . . . .	836
8.134osclconfig_compiler_warnings.h File Reference . . . . .	837
8.134.1 Detailed Description . . . . .	837
8.134.2 Define Documentation . . . . .	837
8.134.2.1 OSCL_FUNCTION_PTR . . . . .	837
8.135osclconfig_error.h File Reference . . . . .	838
8.135.1 Detailed Description . . . . .	838
8.135.2 Define Documentation . . . . .	838
8.135.2.1 OSCL_HAS_ERRNO_H . . . . .	838
8.135.2.2 OSCL_HAS_EXCEPTIONS . . . . .	838
8.135.2.3 OSCL_HAS_SETJMP_H . . . . .	838
8.135.2.4 OSCL_HAS_SYMBIAN_ERRORTRAP . . . . .	838
8.136osclconfig_error_check.h File Reference . . . . .	839
8.137osclconfig_global_new_delete.h File Reference . . . . .	840
8.138osclconfig_global_placement_new.h File Reference . . . . .	841
8.138.1 Function Documentation . . . . .	841
8.138.1.1 operator new . . . . .	841
8.139osclconfig_io.h File Reference . . . . .	842
8.139.1 Detailed Description . . . . .	844
8.139.2 Define Documentation . . . . .	846
8.139.2.1 MAX_TOSCLFILEOFFSET_VALUE . . . . .	846
8.139.2.2 OSCL_AF_INET . . . . .	846
8.139.2.3 OSCL_FILE_BUFFER_MAX_SIZE . . . . .	846
8.139.2.4 OSCL_HAS_ANSI_64BIT_FILE_IO_SUPPORT . . . . .	846
8.139.2.5 OSCL_HAS_ANSI_FILE_IO_SUPPORT . . . . .	846
8.139.2.6 OSCL_HAS_BERKELEY_SOCKETS . . . . .	846
8.139.2.7 OSCL_HAS_GLOB . . . . .	846

8.139.2.8 OSCL_HAS_LARGE_FILE_SUPPORT . . . . .	846
8.139.2.9 OSCL_HAS_MSWIN_FILE_IO_SUPPORT . . . . .	846
8.139.2.10 OSCL_HAS_NATIVE_FILE_CACHE_ENABLE . . . . .	846
8.139.2.11 OSCL_HAS_PV_FILE_CACHE . . . . .	846
8.139.2.12 OSCL_HAS_SOCKET_SUPPORT . . . . .	846
8.139.2.13 OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION . . . . .	846
8.139.2.14 OSCL_HAS_SYMBIAN_DNS_SERVER . . . . .	846
8.139.2.15 OSCL_HAS_SYMBIAN_SOCKET_SERVER . . . . .	846
8.139.2.16 OSCL_IPPROTO_IP . . . . .	846
8.139.2.17 OSCL_IPPROTO_TCP . . . . .	846
8.139.2.18 OSCL_IPPROTO_UDP . . . . .	846
8.139.2.19 OSCL_SD_BOTH . . . . .	846
8.139.2.20 OSCL_SD_RECEIVE . . . . .	846
8.139.2.21 OSCL_SD_SEND . . . . .	846
8.139.2.22 OSCL SOCK_DGRAM . . . . .	846
8.139.2.23 OSCL SOCK_STREAM . . . . .	846
8.139.2.24 OSCL_SOCKOPT_IP_ADDMEMBERSHIP . . . . .	846
8.139.2.25 OSCL_SOCKOPT_IP_MULTICAST_TTL . . . . .	846
8.139.2.26 OSCL_SOCKOPT_IP_TOS . . . . .	846
8.139.2.27 OSCL_SOCKOPT_SOL_REUSEADDR . . . . .	846
8.139.2.28 OSCL_SOL_IP . . . . .	846
8.139.2.29 OSCL_SOL_SOCKET . . . . .	846
8.139.2.30 OSCL_SOL_TCP . . . . .	846
8.139.2.31 OSCL_SOL_UDP . . . . .	846
8.139.2.32 OsclAccept . . . . .	846
8.139.2.33 OsclBind . . . . .	847
8.139.2.34 OsclCloseSocket . . . . .	847
8.139.2.35 OsclConnect . . . . .	847
8.139.2.36 OsclConnectComplete . . . . .	847
8.139.2.37 OsclGetAsyncSockErr . . . . .	847
8.139.2.38 OsclGetDottedAddr . . . . .	848
8.139.2.39 OsclGetDottedAddrVector . . . . .	848
8.139.2.40 OsclGetHostname . . . . .	848
8.139.2.41 OsclGetPeerName . . . . .	848
8.139.2.42 OsclJoin . . . . .	848
8.139.2.43 OsclListen . . . . .	849

---

8.139.2.44 OsclMakeInAddr . . . . .	849
8.139.2.45 OsclMakeSockAddr . . . . .	849
8.139.2.46 OsclPipe . . . . .	849
8.139.2.47 OsclReadFD . . . . .	849
8.139.2.48 OsclRecv . . . . .	849
8.139.2.49 OsclRecvFrom . . . . .	849
8.139.2.50 OsclSend . . . . .	850
8.139.2.51 OsclSendTo . . . . .	850
8.139.2.52 OsclSetNonBlocking . . . . .	850
8.139.2.53 OsclSetRecvBufferSize . . . . .	850
8.139.2.54 OsclSetSockOpt . . . . .	850
8.139.2.55 OsclShutdown . . . . .	850
8.139.2.56 OsclSocket . . . . .	851
8.139.2.57 OsclSocketCleanup . . . . .	851
8.139.2.58 OsclSocketSelect . . . . .	851
8.139.2.59 OsclSocketStartup . . . . .	851
8.139.2.60 OsclUnMakeInAddr . . . . .	851
8.139.2.61 OsclUnMakeSockAddr . . . . .	851
8.139.2.62 OsclValidInetAddr . . . . .	851
8.139.2.63 OsclWriteFD . . . . .	851
8.139.3 Typedef Documentation . . . . .	851
8.139.3.1 TIpMReq . . . . .	851
8.139.3.2 TOsclFileOffset . . . . .	851
8.139.3.3 TOsclHostent . . . . .	851
8.139.3.4 TOsclSockAddr . . . . .	851
8.139.3.5 TOsclSockAddrLen . . . . .	851
8.139.3.6 TOsclSocket . . . . .	851
8.140 osclconfig_io_check.h File Reference . . . . .	852
8.140.1 Typedef Documentation . . . . .	852
8.140.1.1 __verify__TOsclFileOffset__defined__ . . . . .	852
8.141 osclconfig_ix86.h File Reference . . . . .	853
8.141.1 Detailed Description . . . . .	853
8.141.2 Define Documentation . . . . .	853
8.141.2.1 OSCL_BYTE_ORDER_BIG_ENDIAN . . . . .	853
8.141.2.2 OSCL_BYTE_ORDER_LITTLE_ENDIAN . . . . .	853
8.141.2.3 OSCL_INTEGERS_WORD_ALIGNED . . . . .	853

8.142osclconfig_lib.h File Reference . . . . .	854
8.142.1 Detailed Description . . . . .	854
8.142.2 Define Documentation . . . . .	854
8.142.2.1 OSCL_HAS_RUNTIME_LIB_LOADING_SUPPORT . . . . .	854
8.142.2.2 PV_DYNAMIC_LOADING_CONFIG_FILE_PATH . . . . .	854
8.142.2.3 PV_RUNTIME_LIB_FILENAME_EXTENSION . . . . .	854
8.143osclconfig_lib_check.h File Reference . . . . .	855
8.144osclconfig_limits_typedefs.h File Reference . . . . .	856
8.144.1 Detailed Description . . . . .	856
8.144.2 Define Documentation . . . . .	856
8.144.2.1 OSCL_CHAR_IS_SIGNED . . . . .	856
8.144.2.2 OSCL_CHAR_IS_UNSIGNED . . . . .	856
8.145osclconfig_memory.h File Reference . . . . .	857
8.145.1 Define Documentation . . . . .	857
8.145.1.1 OSCL_BYPASS_MEMMGT . . . . .	857
8.145.1.2 OSCL_HAS_GLOBAL_NEW_DELETE . . . . .	857
8.145.1.3 OSCL_HAS_HEAP_BASE_SUPPORT . . . . .	857
8.145.1.4 OSCL_HAS_SYMBIAN_MEMORY_FUNCS . . . . .	857
8.145.1.5 PVMEM_INST_LEVEL . . . . .	857
8.146osclconfig_memory_check.h File Reference . . . . .	858
8.147osclconfig_no_os.h File Reference . . . . .	859
8.148osclconfig_proc.h File Reference . . . . .	860
8.148.1 Detailed Description . . . . .	860
8.149osclconfig_proc_check.h File Reference . . . . .	861
8.149.1 Typedef Documentation . . . . .	861
8.149.1.1 __verify__TOsclConditionObject_defined_ . . . . .	861
8.149.1.2 __verify__TOsclMutexObject_defined_ . . . . .	861
8.149.1.3 __verify__TOsclSemaphoreObject_defined_ . . . . .	861
8.149.1.4 __verify__TOsclThreadFuncArg_defined_ . . . . .	861
8.149.1.5 __verify__TOsclThreadFuncRet_defined_ . . . . .	861
8.149.1.6 __verify__TOsclThreadId_defined_ . . . . .	861
8.149.1.7 __verify__TOsclThreadObject_defined_ . . . . .	862
8.150osclconfig_proc_unix_android.h File Reference . . . . .	863
8.150.1 Define Documentation . . . . .	864
8.150.1.1 OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT . . . . .	864
8.150.1.2 OSCL_HAS_PTHREAD_SUPPORT . . . . .	864

---

8.150.1.3 OSCL_HAS_SEM_TIMEDWAIT_SUPPORT . . . . .	864
8.150.1.4 OSCL_HAS_SYMBIAN_SCHEDULER . . . . .	864
8.150.1.5 OSCL_HAS_THREAD_SUPPORT . . . . .	864
8.150.1.6 OSCL_THREAD_DECL . . . . .	864
8.150.2 Typedef Documentation . . . . .	864
8.150.2.1 TOsclConditionObject . . . . .	864
8.150.2.2 TOsclMutexObject . . . . .	864
8.150.2.3 TOsclSemaphoreObject . . . . .	864
8.150.2.4 TOsclThreadFuncArg . . . . .	864
8.150.2.5 TOsclThreadFuncRet . . . . .	864
8.150.2.6 TOsclThreadId . . . . .	864
8.150.2.7 TOsclThreadObject . . . . .	864
8.151 osclconfig_proc_unix_common.h File Reference . . . . .	865
8.151.1 Define Documentation . . . . .	866
8.151.1.1 OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT . . . . .	866
8.151.1.2 OSCL_HAS_PTHREAD_SUPPORT . . . . .	866
8.151.1.3 OSCL_HAS_SEM_TIMEDWAIT_SUPPORT . . . . .	866
8.151.1.4 OSCL_HAS_SYMBIAN_SCHEDULER . . . . .	866
8.151.1.5 OSCL_HAS_THREAD_SUPPORT . . . . .	866
8.151.1.6 OSCL_THREAD_DECL . . . . .	866
8.151.2 Typedef Documentation . . . . .	866
8.151.2.1 TOsclConditionObject . . . . .	866
8.151.2.2 TOsclMutexObject . . . . .	866
8.151.2.3 TOsclSemaphoreObject . . . . .	866
8.151.2.4 TOsclThreadFuncArg . . . . .	866
8.151.2.5 TOsclThreadFuncRet . . . . .	866
8.151.2.6 TOsclThreadId . . . . .	866
8.151.2.7 TOsclThreadObject . . . . .	866
8.152 osclconfig_time.h File Reference . . . . .	867
8.152.1 Define Documentation . . . . .	867
8.152.1.1 OSCL_HAS_UNIX_TIME_FUNCS . . . . .	867
8.152.2 Typedef Documentation . . . . .	867
8.152.2.1 OsclBasicDateTimeStruct . . . . .	867
8.152.2.2 OsclBasicTimeStruct . . . . .	867
8.153 osclconfig_time_check.h File Reference . . . . .	868
8.153.1 Typedef Documentation . . . . .	868

8.153.1.1 <code>__Validate__BasicTimeDateStruct</code>	868
8.153.1.2 <code>__Validate__BasicTimeStruct</code>	868
8.154 <code>osclconfig_unix_android.h</code> File Reference	869
8.154.1 Define Documentation	872
8.154.1.1 <code>_STRLIT</code>	872
8.154.1.2 <code>_STRLIT_CHAR</code>	872
8.154.1.3 <code>_STRLIT_WCHAR</code>	872
8.154.1.4 <code>INT64</code>	872
8.154.1.5 <code>INT64_HILO</code>	872
8.154.1.6 <code>OSCL_DISABLE_INLINES</code>	872
8.154.1.7 <code>OSCL_HAS_ANSI_MATH_SUPPORT</code>	872
8.154.1.8 <code>OSCL_HAS_ANSI_STDIO_SUPPORT</code>	872
8.154.1.9 <code>OSCL_HAS_ANSI_STDLIB_SUPPORT</code>	872
8.154.1.10 <code>OSCL_HAS_ANSI_STRING_SUPPORT</code>	872
8.154.1.11 <code>OSCL_HAS_ANSI_WIDE_STRING_SUPPORT</code>	872
8.154.1.12 <code>OSCL_HAS_BASIC_LOCK</code>	872
8.154.1.13 <code>OSCL_HAS_GLOBAL_VARIABLE_SUPPORT</code>	872
8.154.1.14 <code>OSCL_HAS_IPHONE_SUPPORT</code>	872
8.154.1.15 <code>OSCL_HAS_MSWIN_PARTIAL_SUPPORT</code>	872
8.154.1.16 <code>OSCL_HAS_MSWIN_SUPPORT</code>	872
8.154.1.17 <code>OSCL_HAS_SYMBIAN_SUPPORT</code>	872
8.154.1.18 <code>OSCL_HAS_TLS_SUPPORT</code>	872
8.154.1.19 <code>OSCL_HAS_UNICODE_SUPPORT</code>	872
8.154.1.20 <code>OSCL_HAS_UNIX_SUPPORT</code>	872
8.154.1.21 <code>OSCL_MEMFRAG_PTR_BEFORE_LEN</code>	872
8.154.1.22 <code>OSCL_NATIVE_INT64_TYPE</code>	872
8.154.1.23 <code>OSCL_NATIVE_UINT64_TYPE</code>	872
8.154.1.24 <code>OSCL_NATIVE_WCHAR_TYPE</code>	872
8.154.1.25 <code>OSCL_TLS_GET_FUNC</code>	872
8.154.1.26 <code>OSCL_TLS_IS_KEYED</code>	872
8.154.1.27 <code>OSCL_TLS_KEY_CREATE_FUNC</code>	872
8.154.1.28 <code>OSCL_TLS_KEY_DELETE_FUNC</code>	872
8.154.1.29 <code>OSCL_TLS_STORE_FUNC</code>	872
8.154.1.30 <code>UINT64</code>	872
8.154.1.31 <code>UINT64_HILO</code>	872
8.154.2 Typedef Documentation	872

8.154.2.1 TOsclBasicLockObject . . . . .	872
8.154.2.2 TOsclTlsKey . . . . .	872
8.155 osclconfig_unix_common.h File Reference . . . . .	873
8.155.1 Define Documentation . . . . .	876
8.155.1.1 _STRLIT . . . . .	876
8.155.1.2 _STRLIT_CHAR . . . . .	876
8.155.1.3 _STRLIT_WCHAR . . . . .	876
8.155.1.4 INT64 . . . . .	876
8.155.1.5 INT64_HILO . . . . .	876
8.155.1.6 OSCL_DISABLE_INLINES . . . . .	876
8.155.1.7 OSCL_HAS_ANSI_MATH_SUPPORT . . . . .	876
8.155.1.8 OSCL_HAS_ANSI_STDIO_SUPPORT . . . . .	876
8.155.1.9 OSCL_HAS_ANSI_STDLIB_SUPPORT . . . . .	876
8.155.1.10 OSCL_HAS_ANSI_STRING_SUPPORT . . . . .	876
8.155.1.11 OSCL_HAS_ANSI_WIDE_STRING_SUPPORT . . . . .	876
8.155.1.12 OSCL_HAS_BASIC_LOCK . . . . .	876
8.155.1.13 OSCL_HAS_GLOBAL_VARIABLE_SUPPORT . . . . .	876
8.155.1.14 OSCL_HAS_MSWIN_PARTIAL_SUPPORT . . . . .	876
8.155.1.15 OSCL_HAS_MSWIN_SUPPORT . . . . .	876
8.155.1.16 OSCL_HAS_SYMBIAN_SUPPORT . . . . .	876
8.155.1.17 OSCL_HAS_TLS_SUPPORT . . . . .	876
8.155.1.18 OSCL_HAS_UNICODE_SUPPORT . . . . .	876
8.155.1.19 OSCL_HAS_UNIX_SUPPORT . . . . .	876
8.155.1.20 OSCL_MEMFRAG_PTR_BEFORE_LEN . . . . .	876
8.155.1.21 OSCL_NATIVE_INT64_TYPE . . . . .	876
8.155.1.22 OSCL_NATIVE_UINT64_TYPE . . . . .	876
8.155.1.23 OSCL_NATIVE_WCHAR_TYPE . . . . .	876
8.155.1.24 OSCL_TLS_GET_FUNC . . . . .	876
8.155.1.25 OSCL_TLS_IS_KEYED . . . . .	876
8.155.1.26 OSCL_TLS_KEY_CREATE_FUNC . . . . .	876
8.155.1.27 OSCL_TLS_KEY_DELETE_FUNC . . . . .	876
8.155.1.28 OSCL_TLS_STORE_FUNC . . . . .	876
8.155.1.29 UINT64 . . . . .	876
8.155.1.30 UINT64_HILO . . . . .	876
8.155.2 Typedef Documentation . . . . .	876
8.155.2.1 TOsclBasicLockObject . . . . .	876

8.155.2.2 TOsclTlsKey . . . . .	876
8.156osclconfig_util.h File Reference . . . . .	877
8.156.1 Define Documentation . . . . .	877
8.156.1.1 OSCL_CLOCK_HAS_DRIFT_CORRECTION . . . . .	877
8.156.1.2 OSCL_HAS_SNPRINTF_LONGLONG_SUPPORT . . . . .	877
8.156.1.3 OSCL_HAS_SYMBIAN_MATH . . . . .	877
8.156.1.4 OSCL_HAS_SYMBIAN_TIMERS . . . . .	877
8.156.1.5 OSCL_RAND_MAX . . . . .	877
8.156.1.6 SLEEP_ONE_SEC . . . . .	877
8.157osclconfig_util_check.h File Reference . . . . .	878
8.158pvlogger.h File Reference . . . . .	879
8.158.1 Detailed Description . . . . .	881
8.158.2 Define Documentation . . . . .	881
8.158.2.1 _PVLOGGER_LOGBIN . . . . .	881
8.158.2.2 _PVLOGGER_LOGBIN_V . . . . .	881
8.158.2.3 _PVLOGGER_LOGMSG . . . . .	881
8.158.2.4 _PVLOGGER_LOGMSG_V . . . . .	881
8.158.2.5 PVLOGGER_ENABLE . . . . .	882
8.158.2.6 PVLOGGER_INST_LEVEL . . . . .	882
8.158.2.7 PVLOGGER_INST_LEVEL_SUPPORT . . . . .	882
8.158.2.8 PVLOGGER_LOG_USE_ONLY . . . . .	882
8.158.2.9 PVLOGGER_LOGBIN . . . . .	882
8.158.2.10 PVLOGGER_LOGBIN_PVLOGMSG_INST_HLDBG . . . . .	883
8.158.2.11 PVLOGGER_LOGBIN_PVLOGMSG_INST_LLDBG . . . . .	883
8.158.2.12 PVLOGGER_LOGBIN_PVLOGMSG_INST_MLDBG . . . . .	883
8.158.2.13 PVLOGGER_LOGBIN_PVLOGMSG_INST_PROF . . . . .	883
8.158.2.14 PVLOGGER_LOGBIN_PVLOGMSG_INST_REL . . . . .	883
8.158.2.15 PVLOGGER_LOGBIN_V . . . . .	883
8.158.2.16 PVLOGGER_LOGBIN_V_PVLOGMSG_INST_HLDBG . . . . .	883
8.158.2.17 PVLOGGER_LOGBIN_V_PVLOGMSG_INST_LLDBG . . . . .	883
8.158.2.18 PVLOGGER_LOGBIN_V_PVLOGMSG_INST_PROF . . . . .	883
8.158.2.19 PVLOGGER_LOGBIN_V_PVLOGMSG_INST_REL . . . . .	883
8.158.2.20 PVLOGGER_LOGBIN_V_PVLOGMSG_V_INST_MLDBG . . . . .	883
8.158.2.21 PVLOGGER_LOGMSG . . . . .	883
8.158.2.22 PVLOGGER_LOGMSG_PVLOGMSG_INST_HLDBG . . . . .	884
8.158.2.23 PVLOGGER_LOGMSG_PVLOGMSG_INST_LLDBG . . . . .	884

8.158.2.24PVLOGGER_LOGMSG_PVLOGMSG_INST_MLDBG . . . . .	884
8.158.2.25PVLOGGER_LOGMSG_PVLOGMSG_INST_PROF . . . . .	884
8.158.2.26PVLOGGER_LOGMSG_PVLOGMSG_INST_REL . . . . .	884
8.158.2.27PVLOGGER_LOGMSG_V . . . . .	884
8.158.2.28PVLOGGER_LOGMSG_V_PVLOGMSG_INST_HLDBG . . . . .	884
8.158.2.29PVLOGGER_LOGMSG_V_PVLOGMSG_INST_LLDBG . . . . .	884
8.158.2.30PVLOGGER_LOGMSG_V_PVLOGMSG_INST_MLDBG . . . . .	884
8.158.2.31PVLOGGER_LOGMSG_V_PVLOGMSG_INST_PROF . . . . .	884
8.158.2.32PVLOGGER_LOGMSG_V_PVLOGMSG_INST_REL . . . . .	884
8.158.2.33PVLOGMSG_INST_HLDBG . . . . .	884
8.158.2.34PVLOGMSG_INST_LLDBG . . . . .	884
8.158.2.35PVLOGMSG_INST_MLDBG . . . . .	885
8.158.2.36PVLOGMSG_INST_PROF . . . . .	885
8.158.2.37PVLOGMSG_INST_REL . . . . .	885
8.158.3 Variable Documentation . . . . .	885
8.158.3.1 PVLOGGER_LEVEL_UNINITIALIZED . . . . .	885
8.158.3.2 PVLOGMSG_ALERT . . . . .	885
8.158.3.3 PVLOGMSG_CRIT . . . . .	885
8.158.3.4 PVLOGMSG_DEBUG . . . . .	885
8.158.3.5 PVLOGMSG_EMERG . . . . .	885
8.158.3.6 PVLOGMSG_ERR . . . . .	885
8.158.3.7 PVLOGMSG_FATAL_ERROR . . . . .	886
8.158.3.8 PVLOGMSG_INFO . . . . .	886
8.158.3.9 PVLOGMSG_NONFATAL_ERROR . . . . .	886
8.158.3.10PVLOGMSG_NOTICE . . . . .	886
8.158.3.11PVLOGMSG_STACK_TRACE . . . . .	886
8.158.3.12PVLOGMSG_STATISTIC . . . . .	886
8.158.3.13PVLOGMSG_VERBOSE . . . . .	886
8.158.3.14PVLOGMSG_WARNING . . . . .	886
8.159pvlogger_accessories.h File Reference . . . . .	887
8.159.1 Variable Documentation . . . . .	887
8.159.1.1 PVLOGGER_FILTER_ACCEPT . . . . .	887
8.159.1.2 PVLOGGER_FILTER_NEUTRAL . . . . .	887
8.159.1.3 PVLOGGER_FILTER_REJECT . . . . .	887
8.160pvlogger_c.h File Reference . . . . .	888
8.160.1 Detailed Description . . . . .	888

8.160.2 Define Documentation . . . . .	889
8.160.2.1 PVLOGGER_C_INST_LEVEL . . . . .	889
8.160.2.2 PVLOGMSG_C_ALERT . . . . .	889
8.160.2.3 PVLOGMSG_C_CRIT . . . . .	889
8.160.2.4 PVLOGMSG_C_EMERG . . . . .	889
8.160.2.5 PVLOGMSG_C_ERR . . . . .	889
8.160.2.6 PVLOGMSG_C_INFO . . . . .	889
8.160.2.7 PVLOGMSG_C_INST_HLDBG . . . . .	889
8.160.2.8 PVLOGMSG_C_INST_LLDBG . . . . .	889
8.160.2.9 PVLOGMSG_C_INST_MLDBG . . . . .	889
8.160.2.10 PVLOGMSG_C_INST_PROF . . . . .	889
8.160.2.11 PVLOGMSG_C_INST_REL . . . . .	889
8.160.2.12 PVLOGMSG_C_NOTICE . . . . .	889
8.160.2.13 PVLOGMSG_C_STACK_DEBUG . . . . .	889
8.160.2.14 PVLOGMSG_C_STACK_TRACE . . . . .	889
8.160.2.15 PVLOGMSG_C_WARNING . . . . .	889
8.160.3 Function Documentation . . . . .	889
8.160.3.1 pvLogger_GetLoggerObject . . . . .	889
8.160.3.2 pvLogger_IsActive . . . . .	889
8.160.3.3 pvLogger_LogMsgString . . . . .	889
8.161 pvlogger_registry.h File Reference . . . . .	890

# **Chapter 1**

## **Todo List**

Global **MAX\_NUMBER\_OF\_BYTE\_PER\_UTF8** Handle 4-byte surrogate pair representation



# Chapter 2

## Module Index

### 2.1 Modules

Here is a list of all modules:

OSCL config . . . . .	23
OSCL Base . . . . .	26
OSCL Memory . . . . .	52
OSCL Util . . . . .	68
OSCL Error . . . . .	93
OSCL IO . . . . .	103
OSCL Proc . . . . .	130
OSCL Init . . . . .	134



# Chapter 3

## Data Structure Index

### 3.1 Class Hierarchy

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

_OsclHeapBase . . . . .	137
HeapBase . . . . .	170
Oscl_File . . . . .	217
OSCL_String . . . . .	297
OSCL_FastString . . . . .	213
OSCL_HeapString< Alloc > . . . . .	233
OSCL_HeapStringA . . . . .	235
OSCL_StackString< MaxBufSize > . . . . .	294
OsclActiveObject . . . . .	342
OsclAsyncFile . . . . .	348
OsclDNSRequestAO . . . . .	396
OsclGetHostByNameRequest . . . . .	445
OsclSocketRequestAO . . . . .	585
OsclAcceptRequest . . . . .	341
OsclBindRequest . . . . .	355
OsclConnectRequest . . . . .	383
OsclIListenRequest . . . . .	454
OsclRecvFromRequest . . . . .	520
OsclRecvRequest . . . . .	524
OsclSendRequest . . . . .	559
OsclSendToRequest . . . . .	562
OsclShutdownRequest . . . . .	566
PVSchedulerStopper . . . . .	660
OsclAsyncFileBuffer . . . . .	351
OsclBuf . . . . .	372
OsclDNS . . . . .	385
OsclFileCache . . . . .	434
OsclNativeFile . . . . .	492
OsclPtr . . . . .	509
OsclPtrC . . . . .	511
OsclRegistryClient . . . . .	542
OsclSocketServ . . . . .	589
OsclTCPSocket . . . . .	600

OsclTimerObject . . . . .	618
CallbackTimer< Alloc > . . . . .	152
OsclDNSMethod . . . . .	392
OsclGetHostByNameMethod . . . . .	444
OsclSocketMethod . . . . .	580
OsclAcceptMethod . . . . .	339
OsclBindMethod . . . . .	354
OsclConnectMethod . . . . .	381
OsclListenMethod . . . . .	453
OsclRecvFromMethod . . . . .	518
OsclRecvMethod . . . . .	522
OsclSendMethod . . . . .	557
OsclSendToMethod . . . . .	560
OsclShutdownMethod . . . . .	565
OsclSocketServI . . . . .	590
OsclUDPSocket . . . . .	634
OsclExecSchedulerBase . . . . .	425
OsclExecScheduler . . . . .	423
allocator . . . . .	140
BufferMgr . . . . .	145
BufferState . . . . .	146
BufFragGroup< ChainClass, max_frags > . . . . .	148
MediaData< ChainClass, max_frags, local_bufsize > . . . . .	178
BufFragStatusClass . . . . .	151
MediaStatusClass . . . . .	182
CallbackTimerObserver . . . . .	154
OsclTimer< Alloc > . . . . .	613
CFastRep . . . . .	155
CHheapRep . . . . .	157
Oscl_TagTree< T, Alloc >::const_iterator . . . . .	161
CStackRep . . . . .	164
DNSRequestParam . . . . .	166
GetHostByNameParam . . . . .	168
internalLeave . . . . .	172
Oscl_TagTree< T, Alloc >::iterator . . . . .	173
LinkedListElement< LLClass > . . . . .	176
MemAllocator< T > . . . . .	183
OsclMemPoolResizableAllocator::MemPoolBlockInfo . . . . .	184
OsclMemPoolResizableAllocator::MemPoolBufferInfo . . . . .	185
MM_AllocBlockFence . . . . .	186
MM_AllocBlockHdr . . . . .	187
MM_AllocInfo . . . . .	189
MM_AllocNode . . . . .	191
MM_AllocQueryInfo . . . . .	193
MM_Audit_Imp . . . . .	194
MM_AuditOverheadStats . . . . .	195
MM_FailInsertParam . . . . .	196
MM_Stats_CB . . . . .	198
MM_Stats_t . . . . .	200
Oscl_TagTree< T, Alloc >::Node . . . . .	202
NTPTime . . . . .	204
Oscl_Alloc . . . . .	208

Oscl_DefAlloc . . . . .	210
_OsclBasicAllocator . . . . .	135
OsclAllocDestructDealloc . . . . .	346
OsclMemAllocDestructDealloc< T > . . . . .	458
OsclMemBasicAllocDestructDealloc< T > . . . . .	466
OsclMemAllocator . . . . .	457
OsclMemBasicAllocator . . . . .	465
OsclMemPoolFixedChunkAllocator . . . . .	470
OsclMemPoolResizableAllocator . . . . .	475
OsclReadyAlloc . . . . .	514
Oscl_Dealloc . . . . .	209
Oscl_DefAlloc . . . . .	210
Oscl_FileFind . . . . .	226
Oscl_FileServer . . . . .	230
oscl_fsstat . . . . .	232
Oscl_Int64_Utils . . . . .	240
Oscl_Less< T > . . . . .	242
Oscl_Linked_List_Base . . . . .	248
Oscl_Linked_List< LLClass, Alloc > . . . . .	243
Oscl_Map< Key, T, Alloc, Compare > . . . . .	253
Oscl_MTLinked_List< LLClass, Alloc, TheLock > . . . . .	260
Oscl_Opaque_Type_Alloc . . . . .	264
Oscl_Queue< T, Alloc > . . . . .	271
Oscl_Vector< T, Alloc > . . . . .	314
Oscl_Vector< TOsclReady, OsclReadyAlloc > . . . . .	314
Oscl_Opaque_Type_Alloc_LL . . . . .	266
Oscl_Linked_List< LLClass, Alloc > . . . . .	243
Oscl_Opaque_Type_Compare . . . . .	268
OsclPriorityQueue< Qelem, Alloc, Container, Compare > . . . . .	500
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare > . . . . .	500
OsclReadyQ . . . . .	516
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare > . . . . .	500
OsclTimerQ . . . . .	623
Oscl_Pair< T1, T2 > . . . . .	270
Oscl_Queue_Base . . . . .	274
Oscl_Queue< T, Alloc > . . . . .	271
Oscl_Rb_Tree_Base . . . . .	283
Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > . . . . .	277
Oscl_Rb_Tree_Const_Iterator< Value > . . . . .	284
Oscl_Rb_Tree_Iterator< Value > . . . . .	287
Oscl_Rb_Tree_Node_Base . . . . .	291
Oscl_Rb_Tree_Node< Value > . . . . .	290
Oscl_Select1st< V, U > . . . . .	293
oscl_stat_buf . . . . .	296
Oscl_Tag_Base . . . . .	304
Oscl_Tag< Alloc > . . . . .	302
Oscl_TagTree< T, Alloc > . . . . .	306
Oscl_Vector_Base . . . . .	320

Oscl_Vector< T, Alloc > . . . . .	314
Oscl_Vector< TOsclReady, OsclReadyAlloc > . . . . .	314
OSCL_wString . . . . .	335
OSCL_wFastString . . . . .	325
OSCL_wHeapString< Alloc > . . . . .	328
OSCL_wHeapStringA . . . . .	330
OSCL_wStackString< MaxBufSize > . . . . .	333
OsclAOStatus . . . . .	347
OsclAuditCB . . . . .	353
OsclBinStream . . . . .	368
OsclBinIStream . . . . .	356
OsclBinIStreamBigEndian . . . . .	358
OsclBinIStreamLittleEndian . . . . .	361
OsclBinOStream . . . . .	363
OsclBinOStreamBigEndian . . . . .	364
OsclBinOStreamLittleEndian . . . . .	366
Oscl_File::OsclCacheObserver . . . . .	374
OsclCompareLess< T > . . . . .	375
OsclComponentRegistry . . . . .	376
OsclComponentRegistryData . . . . .	378
OsclComponentRegistryElement . . . . .	379
OsclDestructDealloc . . . . .	384
Oscl_TAlloc< T, Alloc > . . . . .	311
OsclAllocDestructDealloc . . . . .	346
OsclDNSIBase . . . . .	389
OsclDNSI . . . . .	387
OsclDNSObserver . . . . .	395
OsclDoubleLink . . . . .	399
OsclPriorityLink . . . . .	498
OsclDoubleListBase . . . . .	401
OsclDoubleList< T > . . . . .	400
OsclPriorityList< T > . . . . .	499
OsclDoubleRunner< T > . . . . .	403
OsclError . . . . .	405
OsclErrorAllocator . . . . .	407
OsclErrorTrap . . . . .	409
OsclErrorTrapImp . . . . .	410
OsclException< LeaveCode > . . . . .	412
OsclExclusiveArrayPtr< T > . . . . .	413
OsclExclusivePtr< T > . . . . .	416
OsclExclusivePtrA< T, Alloc > . . . . .	419
OsclExecSchedulerCommonBase . . . . .	426
OsclExecScheduler . . . . .	423
OsclFileCacheBuffer . . . . .	436
OsclFileHandle . . . . .	438
OsclFileManager . . . . .	439
OsclFileStats . . . . .	441
OsclFileStatsItem . . . . .	442
Oscl_File::OsclFixedCacheParam . . . . .	443
OsclInit . . . . .	446
OsclInteger64Transport . . . . .	447

OsclIpMReq . . . . .	448
OsclIPSocketI . . . . .	449
OsclTCPSocketI . . . . .	602
OsclUDPSocketI . . . . .	636
OsclJump . . . . .	452
OsclLockBase . . . . .	455
OsclMutex . . . . .	488
OsclNullLock . . . . .	497
OsclThreadLock . . . . .	610
OsclMem . . . . .	456
OsclMemAudit . . . . .	460
OSCLMemAutoPtr< T, _Allocator > . . . . .	461
OsclMemGlobalAuditObject . . . . .	468
OsclMemoryFragment . . . . .	469
BufferFragment . . . . .	144
OsclMemPoolFixedChunkAllocatorObserver . . . . .	474
OsclMemPoolResizableAllocatorMemoryObserver . . . . .	484
OsclMemPoolResizableAllocatorObserver . . . . .	485
OsclMemStatsNode . . . . .	486
OsclNameString< __len > . . . . .	490
OsclNativeFileParams . . . . .	495
OsclNetworkAddress . . . . .	496
OsclPriorityQueueBase . . . . .	505
OsclPriorityQueue< Qelem, Alloc, Container, Compare > . . . . .	500
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare > . . . . .	500
OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare > . . . . .	500
OsclProcStatus . . . . .	507
OsclRand . . . . .	513
OsclReadyCompare . . . . .	515
OsclRefCounter . . . . .	525
Oscl_DefAllocWithRefCounter< DefAlloc > . . . . .	211
OsclRefCounterDA . . . . .	527
OsclRefCounterMTDA< LockType > . . . . .	531
OsclRefCounterMTSA< DeallocType, LockType > . . . . .	533
OsclRefCounterSA< DeallocType > . . . . .	535
OsclRefCounterMemFrag . . . . .	529
OsclRegistryAccessClient . . . . .	537
OsclRegistryAccessElement . . . . .	541
OsclRegistryClientImpl . . . . .	544
OsclRegistryAccessClientImpl . . . . .	539
OsclRegistryServTlsImpl . . . . .	547
OsclRegistryAccessClientTlsImpl . . . . .	540
OsclRegistryClientTlsImpl . . . . .	546
OsclScheduler . . . . .	549
OsclSchedulerObserver . . . . .	550
OsclScopedLock< LockClass > . . . . .	551
OsclSelect . . . . .	552
OsclSemaphore . . . . .	554
OsclSharedPtr< TheClass > . . . . .	563
OsclSingletonEx< T, ID, Registry > . . . . .	567

OsclSingletonRegistryEx . . . . .	569
OsclSocketIBase . . . . .	575
OsclSocketI . . . . .	570
OsclSocketObserver . . . . .	584
OsclSocketServIBase . . . . .	592
OsclSocketServI . . . . .	590
OsclSocketServRequestList . . . . .	595
OsclSocketServRequestQElem . . . . .	597
OsclSocketTOS . . . . .	598
OsclThread . . . . .	606
OsclTickCount . . . . .	611
OsclTimerCompare . . . . .	617
OsclTimerObserver . . . . .	622
OsclTLS< T, ID, Registry > . . . . .	624
OsclTLSE< T, ID, Registry > . . . . .	626
OsclTLSRegistry . . . . .	628
OsclTLSRegistryEx . . . . .	629
OsclTrapItem . . . . .	630
OsclTrapStack . . . . .	631
OsclTrapStackItem . . . . .	632
OsclUuid . . . . .	639
PVActiveBase . . . . .	641
OsclActiveObject . . . . .	342
OsclTimerObject . . . . .	618
PVLogger . . . . .	645
PVLoggerAppender . . . . .	652
PVLoggerFilter . . . . .	653
AllPassFilter . . . . .	141
PVLoggerLayout . . . . .	655
PVLoggerRegistry . . . . .	657
PVSockBufRecv . . . . .	661
PVSockBufSend . . . . .	662
PVThreadContext . . . . .	663
Oscl_TAlloc< T, Alloc >::rebind< U, V > . . . . .	665
SocketRequestParam . . . . .	672
AcceptParam . . . . .	139
BindParam . . . . .	143
ConnectParam . . . . .	160
ListenParam . . . . .	177
RecvFromParam . . . . .	666
RecvParam . . . . .	668
SendParam . . . . .	669
SendToParam . . . . .	670
ShutdownParam . . . . .	671
StrPtrLen . . . . .	677
StrCSumPtrLen . . . . .	674
TimeValue . . . . .	680
TLSStorageOps . . . . .	687
TReadyQueLink . . . . .	688
Oscl_Map< Key, T, Alloc, Compare >::value_compare . . . . .	690
WStrPtrLen . . . . .	692

# Chapter 4

## Data Structure Index

### 4.1 Data Structures

Here are the data structures with brief descriptions:

_OsclBasicAllocator . . . . .	135
_OsclHeapBase . . . . .	137
AcceptParam . . . . .	139
allocator . . . . .	140
AllPassFilter . . . . .	141
BindParam . . . . .	143
BufferFragment . . . . .	144
BufferMgr . . . . .	145
BufferState . . . . .	146
BufFragGroup< ChainClass, max_frags > . . . . .	148
BufFragStatusClass . . . . .	151
CallbackTimer< Alloc > . . . . .	152
CallbackTimerObserver . . . . .	154
CFastRep . . . . .	155
CHheapRep . . . . .	157
ConnectParam . . . . .	160
Oscl_TagTree< T, Alloc >::const_iterator . . . . .	161
CStackRep . . . . .	164
DNSRequestParam . . . . .	166
GetHostNameParam . . . . .	168
HeapBase . . . . .	170
internalLeave . . . . .	172
Oscl_TagTree< T, Alloc >::iterator . . . . .	173
LinkedListElement< LLClass > . . . . .	176
ListenParam . . . . .	177
MediaData< ChainClass, max_frags, local_bufsize > . . . . .	178
MediaStatusClass . . . . .	182
MemAllocator< T > . . . . .	183
OsclMemPoolResizableAllocator::MemPoolBlockInfo . . . . .	184
OsclMemPoolResizableAllocator::MemPoolBufferInfo . . . . .	185
MM_AllocBlockFence . . . . .	186
MM_AllocBlockHdr . . . . .	187
MM_AllocInfo . . . . .	189

MM_AllocNode . . . . .	191
MM_AllocQueryInfo . . . . .	193
MM_Audit_Imp . . . . .	194
MM_AuditOverheadStats . . . . .	195
MM_FailInsertParam . . . . .	196
MM_Stats_CB . . . . .	198
MM_Stats_t . . . . .	200
Oscl_TagTree< T, Alloc >::Node . . . . .	202
NTPTime (Time value as the number of seconds since 0h (UTC) Jan. 1, 1900 ) . . . . .	204
Oscl_Alloc . . . . .	208
Oscl_Dealloc . . . . .	209
Oscl_DefAlloc . . . . .	210
Oscl_DefAllocWithRefCounter< DefAlloc > . . . . .	211
OSCL_FastString . . . . .	213
Oscl_File . . . . .	217
Oscl_FileFind . . . . .	226
Oscl_FileServer . . . . .	230
oscl_fstat . . . . .	232
OSCL_HeapString< Alloc > . . . . .	233
OSCL_HeapStringA . . . . .	235
Oscl_Int64_Utils (Wrapper for commonly used int64/uint64 operations ) . . . . .	240
Oscl_Less< T > . . . . .	242
Oscl_Linked_List< LLClass, Alloc > . . . . .	243
Oscl_Linked_List_Base . . . . .	248
Oscl_Map< Key, T, Alloc, Compare > . . . . .	253
Oscl_MTLinked_List< LLClass, Alloc, TheLock > . . . . .	260
Oscl_Opaque_Type_Alloc . . . . .	264
Oscl_Opaque_Type_Alloc_LL . . . . .	266
Oscl_Opaque_Type_Compare . . . . .	268
Oscl_Pair< T1, T2 > . . . . .	270
Oscl_Queue< T, Alloc > . . . . .	271
Oscl_Queue_Base . . . . .	274
Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc > . . . . .	277
Oscl_Rb_Tree_Base . . . . .	283
Oscl_Rb_Tree_Const_Iterator< Value > . . . . .	284
Oscl_Rb_Tree_Iterator< Value > . . . . .	287
Oscl_Rb_Tree_Node< Value > . . . . .	290
Oscl_Rb_Tree_Node_Base . . . . .	291
Oscl_Select1st< V, U > . . . . .	293
OSCL_StackString< MaxBufSize > . . . . .	294
oscl_stat_buf . . . . .	296
OSCL_String . . . . .	297
Oscl_Tag< Alloc > . . . . .	302
Oscl_Tag_Base . . . . .	304
Oscl_TagTree< T, Alloc > . . . . .	306
Oscl_TAlloc< T, Alloc > . . . . .	311
Oscl_Vector< T, Alloc > . . . . .	314
Oscl_Vector_Base . . . . .	320
OSCL_wFastString . . . . .	325
OSCL_wHeapString< Alloc > . . . . .	328
OSCL_wHeapStringA . . . . .	330
OSCL_wStackString< MaxBufSize > . . . . .	333
OSCL_wString . . . . .	335
OsclAcceptMethod . . . . .	339

OsclAcceptRequest	341
OsclActiveObject	342
OsclAllocDestructDealloc	346
OsclAOStatus	347
OsclAsyncFile	348
OsclAsyncFileBuffer	351
OsclAuditCB	353
OsclBindMethod	354
OsclBindRequest	355
OsclBinIStream	356
OsclBinIStreamBigEndian	358
OsclBinIStreamLittleEndian	361
OsclBinOStream (Class OsclBinOStream implements the basic stream functions for an output stream )	363
OsclBinOStreamBigEndian (Class OsclBinOStreamBigEndian implements a binary output stream using big endian byte ordering )	364
OsclBinOStreamLittleEndian (Class OsclBinOStreamLittleEndian implements a binary output stream using little endian byte ordering )	366
OsclBinStream	368
OsclBuf	372
Oscl_File::OsclCacheObserver	374
OsclCompareLess< T >	375
OsclComponentRegistry	376
OsclComponentRegistryData	378
OsclComponentRegistryElement	379
OsclConnectMethod	381
OsclConnectRequest	383
OsclDestructDealloc	384
OsclDNS	385
OsclDNSI	387
OsclDNSIBase	389
OsclDNSMethod	392
OsclDNSObserver	395
OsclDNSRequestAO	396
OsclDoubleLink	399
OsclDoubleList< T >	400
OsclDoubleListBase	401
OsclDoubleRunner< T >	403
OsclError	405
OsclErrorAllocator (This class provides static methods to invoke the user defined memory allocation routines )	407
OsclErrorTrap	409
OsclErrorTrapImp	410
OsclException< LeaveCode > (Oscl_exception.h contains all the exception handling macros and classes )	412
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 or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory )	413
OsclExclusivePtr< T > (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 )	416

<b>OsclExclusivePtrA&lt; T, Alloc &gt;</b> (The <code>OsclExclusivePtrA</code> 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 <code>OsclExclusivePtrA</code> expires, Alloc is used to free the memory ) . . . . .	419
<code>OsclExecScheduler</code> . . . . .	423
<code>OsclExecSchedulerBase</code> . . . . .	425
<code>OsclExecSchedulerCommonBase</code> . . . . .	426
<code>OsclFileCache</code> . . . . .	434
<code>OsclFileCacheBuffer</code> . . . . .	436
<code>OsclFileHandle</code> . . . . .	438
<code>OsclFileManager</code> . . . . .	439
<code>OsclFileStats</code> . . . . .	441
<code>OsclFileStatsItem</code> . . . . .	442
<code>Oscl_File::OsclFixedCacheParam</code> . . . . .	443
<code>OsclGetHostByNameMethod</code> . . . . .	444
<code>OsclGetHostByNameRequest</code> . . . . .	445
<code>OsclInit</code> . . . . .	446
<code>OsclInteger64Transport</code> . . . . .	447
<code>OsclIpMReq</code> . . . . .	448
<code>OsclIPSocketI</code> . . . . .	449
<code>OsclJump</code> . . . . .	452
<code>OsclListenMethod</code> . . . . .	453
<code>OsclListenRequest</code> . . . . .	454
<code>OsclLockBase</code> . . . . .	455
<code>OsclMem</code> . . . . .	456
<code>OsclMemAllocator</code> . . . . .	457
<code>OsclMemAllocDestructDealloc&lt; T &gt;</code> . . . . .	458
<code>OsclMemAudit</code> . . . . .	460
<b>OSCLMemAutoPtr&lt; T, _Allocator &gt;</b> (The <code>oscl_auto_ptr</code> class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the <code>oscl_auto_ptr</code> expires, its destructor uses delete to free the memory ) . . . . .	461
<code>OsclMemBasicAllocator</code> . . . . .	465
<code>OsclMemBasicAllocDestructDealloc&lt; T &gt;</code> . . . . .	466
<code>OsclMemGlobalAuditObject</code> . . . . .	468
<code>OsclMemoryFragment</code> . . . . .	469
<code>OsclMemPoolFixedChunkAllocator</code> . . . . .	470
<code>OsclMemPoolFixedChunkAllocatorObserver</code> . . . . .	474
<code>OsclMemPoolResizableAllocator</code> . . . . .	475
<code>OsclMemPoolResizableAllocatorMemoryObserver</code> . . . . .	484
<code>OsclMemPoolResizableAllocatorObserver</code> . . . . .	485
<code>OsclMemStatsNode</code> . . . . .	486
<code>OsclMutex</code> . . . . .	488
<code>OsclNameString&lt; __len &gt;</code> . . . . .	490
<code>OsclNativeFile</code> . . . . .	492
<code>OsclNativeFileParams</code> . . . . .	495
<code>OsclNetworkAddress</code> . . . . .	496
<code>OsclNullLock</code> . . . . .	497
<code>OsclPriorityLink</code> . . . . .	498
<code>OsclPriorityList&lt; T &gt;</code> . . . . .	499
<code>OsclPriorityQueue&lt; Qelem, Alloc, Container, Compare &gt;</code> . . . . .	500
<code>OsclPriorityQueueBase</code> . . . . .	505
<code>OsclProcStatus</code> . . . . .	507
<code>OsclPtr</code> . . . . .	509

OsclPtrC . . . . .	511
OsclRand . . . . .	513
OsclReadyAlloc . . . . .	514
OsclReadyCompare . . . . .	515
OsclReadyQ . . . . .	516
OsclRecvFromMethod . . . . .	518
OsclRecvFromRequest . . . . .	520
OsclRecvMethod . . . . .	522
OsclRecvRequest . . . . .	524
OsclRefCounter . . . . .	525
OsclRefCounterDA . . . . .	527
OsclRefCounterMemFrag . . . . .	529
OsclRefCounterMTDA< LockType > . . . . .	531
OsclRefCounterMTSA< DeallocType, LockType > . . . . .	533
OsclRefCounterSA< DeallocType > . . . . .	535
OsclRegistryAccessClient . . . . .	537
OsclRegistryAccessClientImpl . . . . .	539
OsclRegistryAccessClientTlsImpl . . . . .	540
OsclRegistryAccessElement . . . . .	541
OsclRegistryClient . . . . .	542
OsclRegistryClientImpl . . . . .	544
OsclRegistryClientTlsImpl . . . . .	546
OsclRegistryServTlsImpl . . . . .	547
OsclScheduler . . . . .	549
OsclSchedulerObserver . . . . .	550
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 ) . . . . .	551
OsclSelect . . . . .	552
OsclSemaphore . . . . .	554
OsclSendMethod . . . . .	557
OsclSendRequest . . . . .	559
OsclSendToMethod . . . . .	560
OsclSendToRequest . . . . .	562
OsclSharedPtr< TheClass > (A parameterized smart pointer class ) . . . . .	563
OsclShutdownMethod . . . . .	565
OsclShutdownRequest . . . . .	566
OsclSingletonEx< T, ID, Registry > . . . . .	567
OsclSingletonRegistryEx . . . . .	569
OsclSocketI . . . . .	570
OsclSocketIBase . . . . .	575
OsclSocketMethod . . . . .	580
OsclSocketObserver . . . . .	584
OsclSocketRequestAO . . . . .	585
OsclSocketServ . . . . .	589
OsclSocketServi . . . . .	590
OsclSocketServIBase . . . . .	592
OsclSocketServRequestList . . . . .	595
OsclSocketServRequestQElem . . . . .	597
OsclSocketTOS . . . . .	598
OsclTCPSocket . . . . .	600
OsclTCPSocketI . . . . .	602
OsclThread . . . . .	606
OsclThreadLock . . . . .	610

OsclTickCount . . . . .	611
OsclTimer< Alloc > . . . . .	613
OsclTimerCompare . . . . .	617
OsclTimerObject . . . . .	618
OsclTimerObserver . . . . .	622
OsclTimerQ . . . . .	623
OsclTLS< T, ID, Registry > . . . . .	624
OsclTLSEx< T, ID, Registry > . . . . .	626
OsclTLSRegistry . . . . .	628
OsclTLSRegistryEx . . . . .	629
OsclTrapItem . . . . .	630
OsclTrapStack . . . . .	631
OsclTrapStackItem . . . . .	632
OsclUDPSocket . . . . .	634
OsclUDPSocketI . . . . .	636
OsclUuid . . . . .	639
PVActiveBase . . . . .	641
PVLogger . . . . .	645
PVLoggerAppender . . . . .	652
PVLoggerFilter . . . . .	653
PVLoggerLayout . . . . .	655
PVLoggerRegistry . . . . .	657
PVSchedulerStopper . . . . .	660
PVSockBufRecv . . . . .	661
PVSockBufSend . . . . .	662
PVThreadContext . . . . .	663
Oscl_TAlloc< T, Alloc >::rebind< U, V > . . . . .	665
RecvFromParam . . . . .	666
RecvParam . . . . .	668
SendParam . . . . .	669
SendToParam . . . . .	670
ShutdownParam . . . . .	671
SocketRequestParam . . . . .	672
StrCSumPtrLen (Same as StrPtrLen, but includes checksum field and method to speed up querying) . . . . .	674
StrPtrLen (This data structure encapsulates a set of functions used to perform) . . . . .	677
TimeValue (Time value in a format native to the system) . . . . .	680
TLSStorageOps . . . . .	687
TReadyQueLink . . . . .	688
Oscl_Map< Key, T, Alloc, Compare >::value_compare . . . . .	690
WStrPtrLen (This data structure encapsulates a set of functions used to perform) . . . . .	692

# Chapter 5

## File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

<code>oscl_aostatus.h</code> (Some basic types used with active objects ) . . . . .	695
<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 ) . . . . .	696
<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 ) . . . . .	697
<code>oscl_base_alloc.h</code> (A basic allocator that does not rely on other modules ) . . . . .	698
<code>oscl_base_macros.h</code> (This file defines common macros and constants for basic compilation support ) . . . . .	699
<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 ) . . . . .	700
<code>oscl_byte_order.h</code> (This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders) ) . . . . .	701
<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 ) . . . . .	702
<code>oscl_dll.h</code> (Defines a DLL entry point ) . . . . .	703
<code>oscl_dns.h</code> (The file <code>oscl_socket.h</code> defines the OSCL DNS APIs ) . . . . .	704
<code>oscl_dns_gethostbyname.h</code> . . . . .	705
<code>oscl_dns_imp.h</code> . . . . .	706
<code>oscl_dns_imp_base.h</code> . . . . .	707
<code>oscl_dns_imp_pv.h</code> . . . . .	708
<code>oscl_dns_method.h</code> . . . . .	709
<code>oscl_dns_param.h</code> . . . . .	710
<code>oscl_dns_request.h</code> . . . . .	711
<code>oscl_dns_tuneables.h</code> . . . . .	712
<code>oscl_double_list.h</code> (Internal use types for scheduler ) . . . . .	713
<code>oscl_errno.h</code> (Defines functions to access additional information on errors where supported through an errno or similar service ) . . . . .	714
<code>oscl_error.h</code> (OSCL Error trap and cleanup include file ) . . . . .	715
<code>oscl_error_allocator.h</code> (Defines a memory allocation class used by the oscl error layer ) . . . . .	716
<code>oscl_error_codes.h</code> (Defines basic error and leave codes ) . . . . .	717
<code>oscl_error_imp.h</code> (Internal error implementation support ) . . . . .	718
<code>oscl_error_imp_cppexceptions.h</code> (Implementation File for Leave using C++ exceptions ) . . . . .	719
<code>oscl_error_imp_fatalerror.h</code> (Implementation File for Leave using system fatal error ) . . . . .	720

<a href="#">oscl_error_imp_jumps.h</a> (Implementation of using Setjmp / Longjmp ) . . . . .	721
<a href="#">oscl_error_trapcleanup.h</a> (OSCL Error trap and cleanup implementation include file ) . . . . .	722
<a href="#">oscl_exception.h</a> (All the exception handling macros and classes ) . . . . .	723
<a href="#">oscl_exclusive_ptr.h</a> (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 ) . . . . .	724
<a href="#">oscl_file_async_read.h</a> . . . . .	725
<a href="#">oscl_file_cache.h</a> (The file <code>oscl_file_cache.h</code> defines the class <code>OsclFileCache</code> ) . . . . .	726
<a href="#">oscl_file_dir_utils.h</a> (The file <code>oscl_file_dir_utils.h</code> defines some unix-style directory ops ) . . . . .	727
<a href="#">oscl_file_find.h</a> (The file <code>oscl_file_find.h</code> defines the class <code>Oscl_FileFind</code> ) . . . . .	729
<a href="#">oscl_file_handle.h</a> (The file <code>oscl_file_handle.h</code> defines the class <code>OsclFileHandle</code> ) . . . . .	730
<a href="#">oscl_file_io.h</a> (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 ) . . . . .	731
<a href="#">oscl_file_manager.h</a> (File management class ) . . . . .	732
<a href="#">oscl_file_native.h</a> (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 ) . . . . .	733
<a href="#">oscl_file_server.h</a> (The file <code>oscl_file_server.h</code> defines the class <code>Oscl_FileServer</code> . This is the porting layer for file server implementations ) . . . . .	734
<a href="#">oscl_file_stats.h</a> (File stats class ) . . . . .	735
<a href="#">oscl_file_types.h</a> (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 ) . . . . .	736
<a href="#">oscl_heapbase.h</a> (OSCL Heap Base include file ) . . . . .	737
<a href="#">oscl_init.h</a> (Global oscl initialization ) . . . . .	738
<a href="#">oscl_int64_utils.h</a> . . . . .	739
<a href="#">oscl_ip_socket.h</a> . . . . .	740
<a href="#">oscl_linked_list.h</a> (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 ) . . . . .	741
<a href="#">oscl_lock_base.h</a> (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 ) . . . . .	742
<a href="#">oscl_map.h</a> (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 ) . . . . .	743
<a href="#">oscl_math.h</a> (Provides math functions ) . . . . .	744
<a href="#">oscl_media_data.h</a> (Defines a container class for media data made up of a collection of memory fragments ) . . . . .	745
<a href="#">oscl_media_status.h</a> (Defines a status values for the <code>MediaData</code> containers ) . . . . .	746
<a href="#">oscl_mem.h</a> (This file contains basic memory definitions for common use across platforms ) . . . . .	747
<a href="#">oscl_mem_audit.h</a> (This file contains the definition and partial implementation of MM_Audit class ) . . . . .	749
<a href="#">oscl_mem_audit_internals.h</a> (This file contains the internal definitions for the mem audit library ) . . . . .	751
<a href="#">oscl_mem_auto_ptr.h</a> (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 ) . . . . .	752
<a href="#">oscl_mem_basic_functions.h</a> (This file contains prototypes for the basic memory functions ) . . . . .	753
<a href="#">oscl_mem_inst.h</a> (The file defines default memory instrumentation level ) . . . . .	754
<a href="#">oscl_mem_mempool.h</a> (This file contains the definition of memory pool allocators ) . . . . .	755
<a href="#">oscl_mutex.h</a> (This file provides implementation of mutex ) . . . . .	756
<a href="#">oscl_namestring.h</a> (Name string class include file ) . . . . .	757

<a href="#">oscl_opaque_type.h</a> (The file <code>oscl_opaque_type.h</code> defines pure virtual classes for working with opaque types ) . . . . .	758
<a href="#">oscl_pqueue.h</a> (Implements a priority queue data structure similar to STL ) . . . . .	759
<a href="#">oscl_procstatus.h</a> . . . . .	760
<a href="#">oscl_queue.h</a> (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	
) . . . . .	761
<a href="#">oscl_rand.h</a> (Provides pseudo-random number generation ) . . . . .	762
<a href="#">oscl_refcounter.h</a> (A general purpose reference counter to object lifetimes ) . . . . .	763
<a href="#">oscl_refcounter_memfrag.h</a> (This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount ) . . . . .	764
<a href="#">oscl_registry_access_client.h</a> (Client-side implementation Registry Access implementation ) . .	765
<a href="#">oscl_registry_client.h</a> (Client-side implementation of OsclRegistry ) . . . . .	766
<a href="#">oscl_registry_client_impl.h</a> (Client-side implementation of OsclRegistryInterface ) . . . . .	767
<a href="#">oscl_registry_serv_impl.h</a> (Server-side implementation of OsclRegistry interfaces ) . . . . .	768
<a href="#">oscl_registry_serv_impl_global.h</a> . . . . .	769
<a href="#">oscl_registry_serv_impl_tls.h</a> . . . . .	770
<a href="#">oscl_registry_types.h</a> (Common types used in Oscl registry interfaces ) . . . . .	771
<a href="#">oscl_scheduler.h</a> . . . . .	772
<a href="#">oscl_scheduler_ao.h</a> (Oscl Scheduler user execution object classes ) . . . . .	773
<a href="#">oscl_scheduler_aobase.h</a> (Oscl Scheduler internal active object classes ) . . . . .	774
<a href="#">oscl_scheduler_readyq.h</a> (Ready q types for oscl scheduler ) . . . . .	775
<a href="#">oscl_scheduler_threadcontext.h</a> (Thread context functions needed by oscl scheduler ) . . . . .	776
<a href="#">oscl_scheduler_tuneables.h</a> (Tunable settings for Oscl Scheduler ) . . . . .	777
<a href="#">oscl_scheduler_types.h</a> (Scheduler common types include file ) . . . . .	778
<a href="#">oscl_semaphore.h</a> (This file provides implementation of mutex ) . . . . .	779
<a href="#">oscl_shared_ptr.h</a> (This file defines a template class <code>OsclSharedPtr</code> which is a "smart pointer" to the parameterized type ) . . . . .	780
<a href="#">oscl_singleton.h</a> (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 ) . . . . .	781
<a href="#">oscl_snprintf.h</a> (Provides a portable implementation of snprintf ) . . . . .	782
<a href="#">oscl_socket.h</a> (The file <code>oscl_socket.h</code> defines the OSCL Socket APIs ) . . . . .	783
<a href="#">oscl_socket_accept.h</a> . . . . .	784
<a href="#">oscl_socket_bind.h</a> . . . . .	785
<a href="#">oscl_socket_connect.h</a> . . . . .	786
<a href="#">oscl_socket_imp.h</a> . . . . .	787
<a href="#">oscl_socket_imp_base.h</a> . . . . .	788
<a href="#">oscl_socket_imp_pv.h</a> . . . . .	789
<a href="#">oscl_socket_listen.h</a> . . . . .	790
<a href="#">oscl_socket_method.h</a> . . . . .	791
<a href="#">oscl_socket_recv.h</a> . . . . .	792
<a href="#">oscl_socket_recv_from.h</a> . . . . .	793
<a href="#">oscl_socket_request.h</a> . . . . .	794
<a href="#">oscl_socket_send.h</a> . . . . .	795
<a href="#">oscl_socket_send_to.h</a> . . . . .	796

<a href="#">oscl_socket_serv_imp.h</a>	797
<a href="#">oscl_socket_serv_imp_base.h</a>	798
<a href="#">oscl_socket_serv_imp_pv.h</a>	799
<a href="#">oscl_socket_serv_imp_reqlist.h</a>	800
<a href="#">oscl_socket_shutdown.h</a>	801
<a href="#">oscl_socket_stats.h</a>	802
<a href="#">oscl_socket_tuneables.h</a>	804
<a href="#">oscl_socket_types.h</a>	805
<a href="#">oscl_stdstring.h</a> (This file provides standard string operations such as strlen, strcpy, etc. ANSI defines undefined behavior when the destination pointer is null for operations such as strcpy, strncat, etc. But, we chose to define one. In such cases, we return the destination as null )	807
<a href="#">oscl_str_ptr_len.h</a> (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally )	809
<a href="#">oscl_string.h</a> (Provides a standardized set of string containers that can be used in place of character arrays )	810
<a href="#">oscl_string_containers.h</a> (Provides a standardized set of string containers that can be used in place of character arrays )	811
<a href="#">oscl_string_rep.h</a> (Contains some internal implementation for string containers )	812
<a href="#">oscl_string_uri.h</a> (Utilities to unescape URIs )	813
<a href="#">oscl_string_utf8.h</a> (Utilities to validate and truncate UTF-8 encoded strings )	814
<a href="#">oscl_string_utils.h</a> (Utilities to parse and convert strings )	815
<a href="#">oscl_string_xml.h</a> (Utilities to escape special characters in XML strings )	816
<a href="#">oscl_tagtree.h</a> (The file <a href="#">oscl_tagtree.h</a> .. )	817
<a href="#">oscl_tcp_socket.h</a>	818
<a href="#">oscl_thread.h</a>	819
<a href="#">oscl_tickcount.h</a> (Defines a data structure for string containment/manipulations where the storage for the string is maintained externally )	821
<a href="#">oscl_time.h</a> (The file <a href="#">oscl_time.h</a> defines two classes <a href="#">NTPTime</a> and <a href="#">TimeValue</a> for getting, manipulating, and formatting time values. The <a href="#">TimeValue</a> class is based on the native system time format while <a href="#">NTPTime</a> is used for the standard Network Time Protocol format )	822
<a href="#">oscl_timer.h</a>	824
<a href="#">oscl_tls.h</a>	825
<a href="#">oscl_tree.h</a> (The file <a href="#">oscl_tree.h</a> defines the template class <a href="#">Oscl_Rb_Tree</a> which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the <a href="#">Oscl_Map</a> class. Memory allocation is abstracted through the use of an allocator template parameter )	826
<a href="#">oscl_types.h</a> (This file contains basic type definitions for common use across platforms )	827
<a href="#">oscl_udp_socket.h</a>	828
<a href="#">oscl_utf8conv.h</a> (Utilities to convert unicode to utf8 and vice versa )	829
<a href="#">oscl_uuid.h</a> (This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32 )	830
<a href="#">oscl_uuid_utils.h</a>	831
<a href="#">oscl_vector.h</a> (The file <a href="#">oscl_vector.h</a> defines the template class <a href="#">Oscl_Vector</a> which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter )	832
<a href="#">osclconfig.h</a> (This file contains configuration information for the linux platform )	833
<a href="#">osclconfig_ansi_memory.h</a> (This file contains common typedefs based on the ANSI C limits.h header )	835
<a href="#">osclconfig_check.h</a>	836
<a href="#">osclconfig_compiler_warnings.h</a> (This file contains the ability to turn off/on compiler warnings )	837
<a href="#">osclconfig_error.h</a> (This file contains the common typedefs and header files needed to compile osclerror )	838

osclconfig_error_check.h . . . . .	839
osclconfig_global_new_delete.h . . . . .	840
osclconfig_global_placement_new.h . . . . .	841
osclconfig_io.h (This file contains common typedefs based on the ANSI C limits.h header ) . . . . .	842
osclconfig_io_check.h . . . . .	852
osclconfig_ix86.h (This file contains configuration information for the ix86 processor family ) . . . . .	853
osclconfig_lib.h (This file contains configuration information for the ANSI build ) . . . . .	854
osclconfig_lib_check.h . . . . .	855
osclconfig_limits_typedefs.h (This file contains common typedefs based on the ANSI C limits.h header ) . . . . .	856
osclconfig_memory.h . . . . .	857
osclconfig_memory_check.h . . . . .	858
osclconfig_no_os.h . . . . .	859
osclconfig_proc.h (This file contains configuration information for the linux platform ) . . . . .	860
osclconfig_proc_check.h . . . . .	861
osclconfig_proc_unix_android.h . . . . .	863
osclconfig_proc_unix_common.h . . . . .	865
osclconfig_time.h . . . . .	867
osclconfig_time_check.h . . . . .	868
osclconfig_unix_android.h . . . . .	869
osclconfig_unix_common.h . . . . .	873
osclconfig_util.h . . . . .	877
osclconfig_util_check.h . . . . .	878
pvlogger.h (This file contains basic logger interfaces for common use across platforms ) . . . . .	879
pvlogger_accessories.h . . . . .	887
pvlogger_c.h (This file contains basic logger interfaces for common use across platforms. C-callable version ) . . . . .	888
pvlogger_registry.h . . . . .	890



# Chapter 6

## Module Documentation

### 6.1 OSCL config

#### Defines

- #define OSCL\_HAS\_UNIX\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_HAS\_SAVAJE\_SUPPORT 0
- #define OSCL\_HAS\_PV\_C\_OS\_SUPPORT 0
- #define OSCL\_HAS\_ANDROID\_SUPPORT 0
- #define OSCL\_HAS\_IPHONE\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0
- #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0
- #define OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_FUNCS 0
- #define OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS 0
- #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 0
- #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- #define OSCL\_HAS\_SYMBIAN\_MATH 0
- #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 0
- #define OSCL\_HAS\_PTHREAD\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION 0
- #define OSCL\_HAS\_SAVAJE\_IO\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER 0
- #define OSCL\_HAS\_SYMBIAN\_DNS\_SERVER 0
- #define OSCL\_HAS\_BERKELEY\_SOCKETS 0

#### Typedefs

- typedef int8 \_\_int8\_\_check\_\_
- typedef uint8 \_\_uint8\_\_check\_\_
- typedef int16 \_\_int16\_\_check\_\_
- typedef uint16 \_\_uint16\_\_check\_\_
- typedef int32 \_\_int32\_\_check\_\_
- typedef uint32 \_\_uint32\_\_check\_\_



### 6.1.1 Define Documentation

- 6.1.1.1 #define OSCL\_HAS\_ANDROID\_SUPPORT 0
- 6.1.1.2 #define OSCL\_HAS\_BERKELEY\_SOCKETS 0
- 6.1.1.3 #define OSCL\_HAS\_IPHONE\_SUPPORT 0
- 6.1.1.4 #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- 6.1.1.5 #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- 6.1.1.6 #define OSCL\_HAS\_PTHREAD\_SUPPORT 0
- 6.1.1.7 #define OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_FUNCS 0
- 6.1.1.8 #define OSCL\_HAS\_PV\_C\_OS\_SUPPORT 0
- 6.1.1.9 #define OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS 0
- 6.1.1.10 #define OSCL\_HAS\_SAVAJE\_IO\_SUPPORT 0
- 6.1.1.11 #define OSCL\_HAS\_SAVAJE\_SUPPORT 0
- 6.1.1.12 #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 0
- 6.1.1.13 #define OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION 0
- 6.1.1.14 #define OSCL\_HAS\_SYMBIAN\_DNS\_SERVER 0
- 6.1.1.15 #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0
- 6.1.1.16 #define OSCL\_HAS\_SYMBIAN\_MATH 0
- 6.1.1.17 #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0
- 6.1.1.18 #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- 6.1.1.19 #define OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER 0
- 6.1.1.20 #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- 6.1.1.21 #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- 6.1.1.22 #define OSCL\_HAS\_UNIX\_SUPPORT 0
- 6.1.1.23 #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 0

### 6.1.2 Typedef Documentation

6.1.2.1 `typedef int16 __int16__check__`

6.1.2.2 `typedef int32 __int32__check__`

6.1.2.3 `typedef int8 __int8__check__`

25

Licensed under the Apache License, Version 2.0

6.1.2.4 `typedef uint16 __uint16__check__`

6.1.2.5 `typedef uint32 __uint32__check__`

6.1.2.6 `typedef uint8 __uint8__check__`

## 6.2 OSCL Base

### Data Structures

- class [\\_OsclBasicAllocator](#)
- class [Oscl\\_Alloc](#)
- class [Oscl\\_Dalloc](#)
- class [Oscl\\_DefAlloc](#)
- class [OsclDestructDealloc](#)
- class [OsclAllocDestructDealloc](#)
- class [Oscl\\_TAlloc< T, Alloc >](#)
- 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 or indirectly) by new. When the [OsclExclusivePtr](#) expires, its destructor uses delete to free the memory.*

- 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 or indirectly) by new. When the [OsclExclusiveArrayPtr](#) expires, its destructor uses delete to free the memory.*

- 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 or indirectly) through Alloc. When the [OsclExclusivePtrA](#) expires, Alloc is used to free the memory.*

- class [LinkedListElement< LLClass >](#)
- class [Oscl\\_Linked\\_List\\_Base](#)
- class [Oscl\\_Linked\\_List< LLClass, Alloc >](#)
- class [Oscl\\_MTLinkedList< LLClass, Alloc, TheLock >](#)
- class [OsclLockBase](#)
- class [OsclNullLock](#)
- 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.*

- struct [Oscl\\_Less< T >](#)
- struct [Oscl\\_Select1st< V, U >](#)
- class [Oscl\\_Map< Key, T, Alloc, Compare >](#)
- class [Oscl\\_Opaque\\_Type\\_Alloc](#)
- class [Oscl\\_Opaque\\_Type\\_Compare](#)
- class [Oscl\\_Opaque\\_Type\\_Alloc\\_LL](#)
- class [Oscl\\_Queue\\_Base](#)
- class [Oscl\\_Queue< T, Alloc >](#)
- class [OsclRefCounter](#)
- class [OsclRefCounterDA](#)
- class [OsclRefCounterSA< DeallocType >](#)
- class [OsclRefCounterMTDA< LockType >](#)
- class [OsclRefCounterMTSA< DeallocType, LockType >](#)
- class [Oscl\\_DefAllocWithRefCounter< DefAlloc >](#)
- class [OsclRefCounterMemFrag](#)

- class [OsclSharedPtr< TheClass >](#)

*A parameterized smart pointer class.*

- struct [Oscl\\_Tag\\_Base](#)
- struct [Oscl\\_Tag< Alloc >](#)
- class [Oscl\\_TagTree< T, Alloc >](#)
- 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.*

- class [TLSStorageOps](#)
- class [OsclTLSRegistry](#)
- class [OsclTLS< T, ID, Registry >](#)
- struct [Oscl\\_Pair< T1, T2 >](#)
- struct [Oscl\\_Rb\\_Tree\\_Node\\_Base](#)
- struct [Oscl\\_Rb\\_Tree\\_Node< Value >](#)
- struct [Oscl\\_Rb\\_Tree\\_Iterator< Value >](#)
- struct [Oscl\\_Rb\\_Tree\\_Const\\_Iterator< Value >](#)
- class [Oscl\\_Rb\\_Tree\\_Base](#)
- class [Oscl\\_Rb\\_Tree< Key, Value, KeyOfValue, Compare, Alloc >](#)
- struct [OsclMemoryFragment](#)
- class [Oscl\\_Vector\\_Base](#)
- class [Oscl\\_Vector< T, Alloc >](#)

## Files

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

*The file [oscl\\_assert.h](#) provides an OSCL\_ASSERT macro to document assumptions and test them during development.*

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

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

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

*A basic allocator that does not rely on other modules.*

- file [oscl\\_base\\_macros.h](#)

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

- file [oscl\\_byte\\_order.h](#)

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

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

*The file defines simple default memory allocator classes. These allocators are used by the [Oscl\\_Vector](#) and [Oscl\\_Map](#) class, etc.*

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

*Defines a DLL entry point.*

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

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

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

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

- file [oscl\\_lock\\_base.h](#)

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

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

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

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

*The file defines default memory instrumentation level.*

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

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

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

*The file [oscl\\_queue.h](#) defines the template class [Oscl\\_Queue](#). It is similar to the `STL::queue` class, with some differences:*

- less complete
- based on array rather than a deque
- some interfaces modeled on [oscl\\_vector](#), for ease of transition Memory allocation is abstracted through the use of an allocator template parameter.

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

*A general purpose reference counter to object lifetimes.*

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

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

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

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

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

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

- file `oscl_tagtree.h`

*The file `oscl_tagtree.h` ...*

- file `oscl_time.h`

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

- file `oscl_tree.h`

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

- file `oscl_types.h`

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

- file `oscl_vector.h`

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

## Defines

- `#define OSCL_ASSERT(_expr) ((_expr)?((void)0):OSCL_Assert(# _expr,__FILE__,__LINE__))`
- `#define OSCL_HAS_SINGLETON_SUPPORT 1`
- `#define NULL (0)`

*The `NULL_TERM_CHAR` is used to terminate c-style strings.*

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

*Type casting macros.*

- `#define OSCL_STATIC_CAST(type, exp) ((type)(exp))`
- `#define OSCL_REINTERPRET_CAST(type, exp) ((type)(exp))`
- `#define OSCL_DYNAMIC_CAST(type, exp) ((type)(exp))`
- `#define OSCL_VIRTUAL_BASE(type) type`
- `#define OSCL_UNUSED_ARG(vbl) (void)(vbl)`
- `#define OSCL_UNUSED_RETURN(value) return value`
- `#define OSCL_MIN(a, b) ((a) < (b) ? (a) : (b))`
- `#define OSCL_MAX(a, b) ((a) > (b) ? (a) : (b))`
- `#define OSCL_ABS(a) ((a) > (0) ? (a) : -(a))`
- `#define EPV_ARM_GNUC 1`
- `#define EPV_ARM_RVCT 2`
- `#define EPV_ARM_MSEVC 3`
- `#define OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`
- `#define ALLOCATE(n) allocate(n)`
- `#define ALLOC_AND_CONSTRUCT(n) alloc_and_construct(n)`

- #define **OSCL\_DLL\_ENTRY\_POINT()** void oscl\_dll\_entry\_point() {}
- #define **OSCL\_DLL\_ENTRY\_POINT\_DEFAULT()**
- #define **OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE**
- #define **PVMEM\_INST\_LEVEL** 0
- #define **OSCL\_DISABLE\_WARNING\_RETURN\_TYPE\_NOT\_UDT**
- #define **OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE**
- #define **OSCL\_TLS\_BASE\_SLOTS OSCL\_TLS\_ID\_BASE\_LAST +1**
- #define **OSCL\_TLS\_MAX\_SLOTS** ( **OSCL\_TLS\_BASE\_SLOTS** + **OSCL\_TLS\_EXTERNAL\_SLOTS** )
- #define **OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE**

## Typedefs

- typedef char **CtimeStrBuf** [**CTIME\_BUFFER\_SIZE**]
- typedef char **PV8601timeStrBuf** [**PV8601TIME\_BUFFER\_SIZE**]
- typedef char **ISO8601timeStrBuf** [**ISO8601TIME\_BUFFER\_SIZE**]
- typedef **OsclAny** **TOsclTlsKey**
- typedef int **c\_bool**

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

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

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

*The uint type is a convenient abbreviation for unsigned int.*
- typedef **uint8 octet**

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

*The Float type defined as OsclFloat.*
- typedef size\_t **OsclSizeT**
- 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.*
- **OsclSharedPtr::OsclSharedPtr (TheClass \*inClassPtr, OsclRefCounter \*in\_refcnt)**

*Constructor.*
- **OsclSharedPtr::OsclSharedPtr (const OsclSharedPtr &inSharedPtr)**

*Copy constructor.*
- **virtual OsclSharedPtr::~OsclSharedPtr ()**

*Destructor.*
- **TheClass \* OsclSharedPtr::operator-> ()**
- **TheClass & OsclSharedPtr::operator\* ()**

*The indirection operator returns a reference to an object of the parameterized type.*
- **OsclSharedPtr::operator TheClass \* ()**

*Casting operator.*
- **TheClass \* OsclSharedPtr::GetRep ()**

*Use this function to get a pointer to the wrapped object.*
- **OsclRefCounter \* OsclSharedPtr::GetRefCounter ()**

*Get the refcount pointer. This should primarily be used for conversion operations.*
- **int OsclSharedPtr::get\_count ()**

*Get a count of how many references to the object exist.*
- **void OsclSharedPtr::Bind (const OsclSharedPtr &inHandle)**

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

- void `OsclSharedPtr::Bind` (TheClass \*ptr, `OsclRefCounter` \*in\_refcnt)

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

- void `OsclSharedPtr::Unbind` ()

*Use this function of unbind an existing `OsclSharedPtr`.*

- `OsclSharedPtr & OsclSharedPtr::operator=` (const `OsclSharedPtr` &inSharedPtr)

*Assignment operator.*

- bool `OsclSharedPtr::operator==` (const `OsclSharedPtr` &b) const

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

- 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_CIstrncmp` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2, uint32 count)

- OSCL\_IMPORT\_REF char `oscl_tolower` (const char car)

- OSCL\_IMPORT\_REF `oscl_wchar` `oscl_tolower` (const `oscl_wchar` car)

- OSCL\_IMPORT\_REF bool `oscl_isLetter` (const char car)

- OSCL\_IMPORT\_REF const char \* `oscl strstr` (const char \*str1, const char \*str2)

- OSCL\_IMPORT\_REF char \* `oscl strstr` (char \*str1, const char \*str2)

- OSCL\_IMPORT\_REF const `oscl_wchar` \* `oscl strstr` (const `oscl_wchar` \*str1, const `oscl_wchar` \*str2)

- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl strstr` (`oscl_wchar` \*str1, const `oscl_wchar` \*str2)

- OSCL\_IMPORT\_REF char \* `oscl_streat` (char \*dest, const char \*src)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_streat` (`oscl_wchar` \*dest, const `oscl_wchar` \*src)
- OSCL\_IMPORT\_REF void `PV8601ToRFC822` (`PV8601timeStrBuf` pv8601\_buffer, `CtimeStrBuf` ctime\_buffer)
- OSCL\_IMPORT\_REF void `ISO8601ToRFC822` (`ISO8601timeStrBuf` iso8601\_buffer, `CtimeStrBuf` ctime\_buffer)
- OSCL\_IMPORT\_REF void `RFC822ToPV8601` (`CtimeStrBuf` ctime\_buffer, `PV8601timeStrBuf`)
- OSCL\_COND\_IMPORT\_REF `TimeValue operator-` (const `TimeValue` &a, const `TimeValue` &b)
- OSCL\_COND\_IMPORT\_REF `TimeValue operator+` (const `TimeValue` &a, const int32 bSeconds)
- OSCL\_COND\_IMPORT\_REF `TimeValue operator+` (const int32 aSeconds, const `TimeValue` &b)
- OSCL\_COND\_IMPORT\_REF `TimeValue operator-` (const `TimeValue` &a, const int32 bSeconds)
- OSCL\_COND\_IMPORT\_REF `TimeValue operator-` (const int32 aSeconds, const `TimeValue` &b)

## Variables

- const int `CTIME_BUFFER_SIZE` = 26
- const int `PV8601TIME_BUFFER_SIZE` = 21
- const int `ISO8601TIME_BUFFER_SIZE` = 21
- const long `USEC_PER_SEC` = 1000000
- const long `MSEC_PER_SEC` = 1000
- const uint32 `unix_ntp_offset` = 2208988800U
- const uint32 `OSCL_TLS_ID_MAGICNUM` = 0
- const uint32 `OSCL_TLS_ID_ERRORHOOK` = 1
- const uint32 `OSCL_TLS_ID_PVLOGGER` = 2
- const uint32 `OSCL_TLS_ID_TEST` = 3
- const uint32 `OSCL_TLS_ID_PVSCHEDULER` = 4
- const uint32 `OSCL_TLS_ID_PVERRORTRAP` = 5
- const uint32 `OSCL_TLS_ID_SDPMEDIAPARSER` = 6
- const uint32 `OSCL_TLS_ID_PAYLOADPARSER` = 7
- const uint32 `OSCL_TLS_ID_PVMFRECOGNIZER` = 8
- const uint32 `OSCL_TLS_ID_WMDRM` = 9
- const uint32 `OSCL_TLS_ID_OSCLREGISTRY` = 10
- const uint32 `OSCL_TLS_ID_SQLITE3` = 11
- const uint32 `OSCL_TLS_ID_BASE_LAST` = 11

### 6.2.1 Detailed Description

Additional osclbase comment

## 6.2.2 Define Documentation

**6.2.2.1 #define ALLOC\_AND\_CONSTRUCT(n) alloc\_and\_construct(n)**

**6.2.2.2 #define ALLOCATE(n) allocate(n)**

**6.2.2.3 #define EPV\_ARM\_GNUC 1**

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

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

**6.2.2.6 #define NULL (0)**

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

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

Referenced by BufFragGroup< ChainClass, max\_frags >::AddFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), Oscl\_TAlloc< node\_type, Alloc >::alloc\_and\_construct(), OsclMemBasicAllocDestructDealloc< T >::allocate(), OsclMemAllocDestructDealloc< T >::allocate(), OsclMemAllocator::allocate(), OsclErrorAllocator::allocate(), Oscl\_TAlloc< node\_type, Alloc >::allocate(), OsclSharedPtr< TheClass >::Bind(), OsclErrorAllocator::deallocate(), OsclSharedPtr< TheClass >::get\_count(), OSCL\_wHeapString< Alloc >::get\_cstr(), OSCL\_HeapString< Alloc >::get\_cstr(), OSCL\_wHeapString< Alloc >::get\_str(), OSCL\_HeapString< Alloc >::get\_str(), BufFragGroup< ChainClass, max\_frags >::GetBufferState(), OsclRegistryClientImpl::GetFactory(), BufFragGroup< ChainClass, max\_frags >::GetFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::GetLocalFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::GetMediaFragment(), Oscl\_TagTree< PVLogger \*, alloc\_type >::insert(), PVActiveBase::IsInAnyQ(), MediaData< ChainClass, max\_frags, local\_bufsize >::IsLocalData(), LinkedListElement< LLClass >::LinkedListElement(), OsclDoubleRunner< T >::operator T \*(), OsclDoubleRunner< T >::operator++(), OSCL\_HeapString< Alloc >::OSCL\_HeapString(), OSCL\_StackString< MaxBufSize >::OSCL\_StackString(), Oscl\_TagTree< PVLogger \*, alloc\_type >::Oscl\_TagTree(), OSCL\_wHeapString< Alloc >::OSCL\_wHeapString(), OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString(), OsclDoubleRunner< T >::OsclDoubleRunner(), OsclMemStatsNode::OsclMemStatsNode(), OsclRefCounterDA::OsclRefCounterDA(), OsclRefCounterMTDA< LockType >::OsclRefCounterMTDA(), OsclRefCounterMTSA< DeallocType, LockType >::OsclRefCounterMTSA(), OsclRefCounterSA< DeallocType >::OsclRefCounterSA(), OsclSocketServIBase::OsclSocketServIBase(), OsclTrapStackItem::OsclTrapStackItem(), OsclExclusivePtrA< T, Alloc >::release(), OsclExclusiveArrayPtr< T >::release(), OsclExclusivePtr< T >::release(), OsclSocketServRequestList::Remove(), OSCL\_wStackString< MaxBufSize >::set(), OSCL\_StackString< MaxBufSize >::set(), OsclExclusivePtrA< T, Alloc >::set(), OsclExclusiveArrayPtr< T >::set(), OsclExclusivePtr< T >::set(), TReadyQueLink::TReadyQueLink(), OsclSharedPtr< TheClass >::Unbind(), OsclSharedPtr< TheClass >::~OsclSharedPtr(), and OsclTimer< Alloc >::~OsclTimer().

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

Referenced by OsclTimer< Alloc >::TimerBaseElapsed().

---

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

Referenced by OsclErrorAllocator::allocate(), OsclPtr::Append(), Oscl\_Queue< T, Alloc >::back(), OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::compare\_EQ(), OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::compare\_LT(), OsclErrorAllocator::deallocate(), Oscl\_Queue< T, Alloc >::front(), OsclTLSRegistryEx::getInstance(), OsclSingletonRegistryEx::getInstance(), OsclJump::Jump(), OsclPtrC::Left(), OsclSingletonRegistryEx::lockAndGetInstance(), OsclRefCounterDA::OsclRefCounterDA(), OsclRefCounterMTDA< LockType >::OsclRefCounterMTDA(), OsclRefCounterMTSA< DeallocType, LockType >::OsclRefCounterMTSA(), OsclRefCounterSA< DeallocType >::OsclRefCounterSA(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::pop\_back(), OsclTLSRegistryEx::registerInstance(), OsclSingletonRegistryEx::registerInstance(), OsclSingletonRegistryEx::registerInstanceAndUnlock(), OsclPtrC::Right(), OsclPtr::SetLength(), OsclPtrC::SetLength(), OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::swap(), OsclJump::Top(), and OsclJump::~OsclJump().

**6.2.2.9 #define OSCL\_COND\_EXPORT\_REF**
**6.2.2.10 #define OSCL\_COND\_IMPORT\_REF**
**6.2.2.11 #define OSCL\_CONST\_CAST(type, exp) ((type)(exp))**

Type casting macros.

**Parameters**

*type* Destination type of cast

*exp* Expression to cast

**6.2.2.12 #define OSCL\_DISABLE\_WARNING\_RETURN\_TYPE\_NOT\_UDT**
**6.2.2.13 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE**
**6.2.2.14 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE**
**6.2.2.15 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE**
**6.2.2.16 #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE**
**6.2.2.17 #define OSCL\_DLL\_ENTRY\_POINT() void oscl\_dll\_entry\_point() {}**

DLL entry/exit point.

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

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

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

Usage :

---

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

#### **6.2.2.18 #define OSCL\_DLL\_ENTRY\_POINT\_DEFAULT()**

Default DLL entry/exit point function.

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

Usage :

```
OSCL_DLL_ENTRY_POINT_DEFAULT()
```

#### **6.2.2.19 #define OSCL\_DYNAMIC\_CAST(type, exp) ((type)(exp))**

#### **6.2.2.20 #define OSCL\_HAS\_SINGLETON\_SUPPORT 1**

#### **6.2.2.21 #define OSCL\_INLINE inline**

#### **6.2.2.22 #define OSCL\_MAX(a, b) ((a) > (b) ? (a) : (b))**

Referenced by OsclTimer< Alloc >::TimerBaseElapsed().

#### **6.2.2.23 #define OSCL\_MIN(a, b) ((a) < (b) ? (a) : (b))**

#### **6.2.2.24 #define OSCL\_REINTERPRET\_CAST(type, exp) ((type)(exp))**

#### **6.2.2.25 #define OSCL\_STATIC\_CAST(type, exp) ((type)(exp))**

Referenced by Oscl\_TAlloc< node\_type, Alloc >::alloc\_and\_construct(), Oscl\_TAlloc< node\_type, Alloc >::alloc\_and\_construct\_fl(), Oscl\_TAlloc< node\_type, Alloc >::allocate(), Oscl\_TAlloc< node\_type, Alloc >::allocate\_fl(), GetHostByNameParam::canPersistMoreHostAddresses(), Oscl\_TAlloc< node\_type, Alloc >::destruct\_and\_dealloc(), OsclTLS< T, ID, Registry >::OsclTLS(), GetHostByNameParam::PersistHostAddress(), OsclTLS< T, ID, Registry >::set(), OsclTLSEEx< T, ID, Registry >::set(), and OsclSingletonEx< T, ID, Registry >::set().

#### **6.2.2.26 #define OSCL\_TLS\_BASE\_SLOTS OSCL\_TLS\_ID\_BASE\_LAST +1**

#### **6.2.2.27 #define OSCL\_TLS\_MAX\_SLOTS ( OSCL\_TLS\_BASE\_SLOTS + OSCL\_TLS\_EXTERNAL\_SLOTS)**

#### **6.2.2.28 #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.

Referenced by PVLogger::AddAppender(), PVLogger::AddFilter(), Oscl\_DefAlloc::allocate\_fl(), Oscl\_Alloc::allocate\_fl(), Oscl\_TAlloc< node\_type, Alloc >::deallocate(), Oscl\_TAlloc< node\_type, Alloc >::destroy(), OsclMemBasicAllocDestructDealloc< T >::destruct\_and\_dealloc(), OsclMemAllocDestructDealloc< T >::destruct\_and\_dealloc(), AllPassFilter::FilterOpaqueMessge(), AllPassFilter::FilterString(), OsclErrorAllocator::operator delete(), MM\_AllocNode::operator new(), MM\_AllocInfo::operator new(), MM\_Stats\_CB::operator new(), OsclMemStatsNode::operator new(), MM\_FailInsertParam::operator new(), MM\_Stats\_t::operator new(), OsclErrorAllocator::operator new(), HeapBase::operator new[ ](), PVLogger::RemoveAppender(), PVLogger::SetLogLevel(), and OsclAsyncFile::Write().

#### 6.2.2.29 #define OSCL\_UNUSED\_RETURN(value) return value

#### 6.2.2.30 #define OSCL\_VIRTUAL\_BASE(type) type

#### 6.2.2.31 #define PVMEM\_INST\_LEVEL 0

### 6.2.3 Typedef Documentation

#### 6.2.3.1 typedef int c\_bool

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

#### 6.2.3.2 typedef char CtimeStrBuf[CTIME\_BUFFER\_SIZE]

#### 6.2.3.3 typedef OSCL\_NATIVE\_INT64\_TYPE int64

#### 6.2.3.4 typedef char ISO8601timeStrBuf[ISO8601TIME\_BUFFER\_SIZE]

#### 6.2.3.5 typedef char mbchar

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

#### 6.2.3.6 typedef uint8 octet

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

#### 6.2.3.7 typedef oscl\_wchar OSCL\_TCHAR

define OSCL\_TCHAR

#### 6.2.3.8 typedef OSCL\_NATIVE\_WCHAR\_TYPE oscl\_wchar

#### 6.2.3.9 typedef void OsclAny

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

### 6.2.3.10 `typedef float OsclFloat`

The Float type defined as OsclFloat.

### 6.2.3.11 `typedef size_t OsclSizeT`

### 6.2.3.12 `typedef char PV8601timeStrBuf[PV8601TIME_BUFFER_SIZE]`

### 6.2.3.13 `typedef OsclAny TOsclTlsKey`

### 6.2.3.14 `typedef unsigned int uint`

The uint type is a convenient abbreviation for unsigned int.

### 6.2.3.15 `typedef OSCL_NATIVE_UINT64_TYPE uint64`

## 6.2.4 Enumeration Type Documentation

### 6.2.4.1 `enum TimeUnits`

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

**Enumerator:**

*SECONDS*

*MILLISECONDS*

*MICROSECONDS*

## 6.2.5 Function Documentation

### 6.2.5.1 `OSCL_COND_IMPORT_REF void _OSCL_Abort ()`

This function terminates the current process abnormally.

Referenced by OsclJump::Jump().

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

Convert big endian to host format.

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

#### Parameters

*data* A pointer to the input/output buffer

*size* The number of bytes in the buffer.

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

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

References NULL, and [OsclRefCounter::removeRef\(\)](#).

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

Use this function to bind an existing [OsclSharedPtr](#) to a already-wrapped object.

References [OsclRefCounter::addRef\(\)](#), NULL, and [OsclRefCounter::removeRef\(\)](#).

Referenced by [OsclSharedPtr< TheClass >::operator=\(\)](#), and [OsclSharedPtr< TheClass >::Unbind\(\)](#).

**6.2.5.5 template<class TheClass> int OsclSharedPtr< TheClass >::get\_count () [inline, inherited]**

Get a count of how many references to the object exist.

References [OsclRefCounter::getCount\(\)](#), and NULL.

**6.2.5.6 template<class TheClass> OsclRefCounter\* OsclSharedPtr< TheClass >::GetRefCounter () [inline, inherited]**

Get the refcount pointer. This should primarily be used for conversion operations.

**6.2.5.7 template<class TheClass> TheClass\* OsclSharedPtr< TheClass >::GetRep () [inline, inherited]**

Use this function to get a pointer to the wrapped object.

Referenced by [PVLLogger::RemoveAppender\(\)](#).

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

Convert host to big endian format.

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

**Parameters**

*data* A pointer to the input/output buffer

*size* The number of bytes in the buffer.

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

Convert host to little endian format.

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

### Parameters

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

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

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

Convert little endian to host format.

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

### Parameters

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

**6.2.5.12 template<class TheClass> OsclSharedPtr< TheClass >::operator TheClass \* ()  
[inline, inherited]**

Casting operator.

**6.2.5.13 template<class TheClass> TheClass& OsclSharedPtr< TheClass >::operator\* ()  
[inline, inherited]**

The indirection operator returns a reference to an object of the parameterized type.

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

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

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

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

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

**6.2.5.19 template<class TheClass> TheClass\* OsclSharedPtr< TheClass >::operator-> ()  
[inline, inherited]**

The dereferencing operator returns a pointer to the parameterized type and can be used to access member elements of TheClass.

**6.2.5.20 template<class TheClass> OsclSharedPtr& OsclSharedPtr< TheClass >::operator= (const OsclSharedPtr< TheClass > & *inSharedPtr*) [inline, inherited]**

Assignment operator.

References OsclSharedPtr< TheClass >::Bind().

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

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

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

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

#### Parameters

*expr* is the expression to be evaluated

*filename* is the name of the current source file

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

**6.2.5.23 OSCL\_IMPORT\_REF int32 oscl\_CIstrcmp (const oscl\_wchar \* *str1*, const oscl\_wchar \* *str2*)**

Case in-sensitive string comparision.

#### Parameters

*str1* string to compare

*str2* string to compare

#### Returns

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

**6.2.5.24 OSCL\_IMPORT\_REF int32 oscl\_CIstrcmp (const char \* *str1*, const char \* *str2*)**

Case in-sensitive string comparision.

#### Parameters

*str1* string to compare

*str2* string to compare

#### Returns

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

---

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

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

#### Parameters

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

#### Returns

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

---

**6.2.5.26 OSCL\_IMPORT\_REF int32 oscl\_Clstrncmp (const char \* str1, const char \* str2, uint32 count)**

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

#### Parameters

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

#### Returns

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

---

**6.2.5.27 OSCL\_IMPORT\_REF bool oscl\_isLetter (const char car)**

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

#### Parameters

*car*

#### Returns

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

---

**6.2.5.28 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_streat (oscl\_wchar \* dest, const oscl\_wchar \* src)**

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

**Parameters**

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

**Returns**

dest

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

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

**Parameters**

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

**Returns**

dest

**6.2.5.30 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strechr (oscl\_wchar \* str, int32 c)****6.2.5.31 OSCL\_IMPORT\_REF const oscl\_wchar\* oscl\_strechr (const oscl\_wchar \* str, int32 c)**

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

**Parameters**

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

**Returns****6.2.5.32 OSCL\_IMPORT\_REF char\* oscl\_strechr (char \* str, int32 c)****6.2.5.33 OSCL\_IMPORT\_REF const char\* oscl\_strechr (const char \* str, int32 c)**

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

**Parameters**

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

**Returns**

---

**6.2.5.34 OSCL\_IMPORT\_REF int32 oscl\_strcmp (const oscl\_wchar \* str1, const oscl\_wchar \* str2)**

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

**Parameters**

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

**Returns**

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

**6.2.5.35 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

Referenced by GetHostNameParam::canPersistMoreHostAddresses(), OsclNetworkAddress::operator==(), and GetHostNameParam::PersistHostAddress().

**6.2.5.36 OSCL\_IMPORT\_REF uint32 oscl\_strlen (const oscl\_wchar \* str)**

Gets the length of a wide char string

**Parameters**

- str* NULL terminated string.

**Returns**

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

**6.2.5.37 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.

Referenced by WStrPtrLen::operator=(), StrPtrLen::operator=(), OSCL\_wStackString< MaxBufSize >::set(), OSCL\_StackString< MaxBufSize >::set(), OSCL\_wHeapString< Alloc >::set(), OSCL\_HeapString< Alloc >::set(), StrPtrLen::StrPtrLen(), Oscl\_Tag\_Base::tag\_cmp(), Oscl\_Tag\_Base::tag\_copy(), Oscl\_Tag\_Base::tag\_len(), and WStrPtrLen::WStrPtrLen().

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

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

**Parameters**

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

**Returns**

dest

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

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

**Parameters**

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

**Returns**

dest

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

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

**Parameters**

*str1* String to compare

*str2* String to compare

*count* Number of characters to compare

**Returns**

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

### **6.2.5.41 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

Referenced by WStrPtrLen::operator==(), StrPtrLen::operator==(), and Oscl\_Tag\_Base::tag\_cmp().

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

Copies the chars of one string to another.

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

**Parameters**

*dest* Destination string

*src* NULL terminated source string

*count* Number of chars to copy

**Returns**

Returns dest.

**6.2.5.43 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.

Referenced by Oscl\_Tag\_Base::tag\_copy().

**6.2.5.44 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strrchr (oscl\_wchar \* str, int32 c)****6.2.5.45 OSCL\_IMPORT\_REF const oscl\_wchar\* oscl\_strrchr (const oscl\_wchar \* str, int32 c)****6.2.5.46 OSCL\_IMPORT\_REF char\* oscl\_strrchr (char \* str, int32 c)****6.2.5.47 OSCL\_IMPORT\_REF const char\* oscl\_strrchr (const char \* str, int32 c)**

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

**Parameters**

*str* null terminated source string

*c* character to search for

**Returns****6.2.5.48 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strset (oscl\_wchar \* dest, oscl\_wchar val, uint32 count)**

Sets the characters of a string to a specified character

**Parameters**

*dest* buffer to modify

*val* character to set

*count* number of chars to set

**Returns**

the value of dest

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

Sets the characters of a string to a specified character

**Parameters**

*dest* buffer to modify

*val* character to set

*count* number of chars to set

**Returns**

the value of dest

**6.2.5.50 OSCL\_IMPORT\_REF oscl\_wchar\* oscl\_strstr (oscl\_wchar \* str1, const oscl\_wchar \* str2)****6.2.5.51 OSCL\_IMPORT\_REF const oscl\_wchar\* oscl\_strstr (const oscl\_wchar \* str1, const oscl\_wchar \* str2)**

find the occurrence of sub-string in a string.

**Parameters**

*str1* string.

*str2* sub-string

**Returns**

pointer to the begining of sub-string.

**6.2.5.52 OSCL\_IMPORT\_REF char\* oscl\_strstr (char \* str1, const char \* str2)****6.2.5.53 OSCL\_IMPORT\_REF const char\* oscl\_strstr (const char \* str1, const char \* str2)**

find the occurrence of sub-string in a string.

**Parameters**

*str1* string.

*str2* sub-string

**Returns**

pointer to the begining of sub-string.

**6.2.5.54 OSCL\_IMPORT\_REF oscl\_wchar oscl\_tolower (const oscl\_wchar car)**

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

**Parameters**

*car* upper case character.

**Returns**

lower case character.

**6.2.5.55 OSCL\_IMPORT\_REF char oscl\_tolower (const char *car*)**

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

**Parameters**

*car* upper case character.

**Returns**

lower case character.

**6.2.5.56 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr (const OsclSharedPtr< TheClass > & *inSharedPtr*) [inline, inherited]**

Copy constructor.

References OsclRefCounter::addRef().

**6.2.5.57 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr (TheClass \* *inClassPtr*, OsclRefCounter \* *in\_refcnt*) [inline, inherited]**

Constructor.

**Parameters**

*inClassPtr* A pointer to an instance of the parameterized type that the new [OsclSharedPtr](#) will wrap.

**6.2.5.58 OSCL\_IMPORT\_REF void PV8601ToRFC822 (PV8601timeStrBuf *pv8601\_buffer*, CtimeStrBuf *ctime\_buffer*)****6.2.5.59 void PVOsclBase\_Cleanup ()**

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

**6.2.5.60 void PVOsclBase\_Init ()**

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

**Exceptions**

*leaves* if out-of-memory

**6.2.5.61 OSCL\_IMPORT\_REF void RFC822ToPV8601 (CtimeStrBuf *ctime\_buffer*, PV8601timeStrBuf)**

**6.2.5.62 template<class TheClass> void OsclSharedPtr< TheClass >::Unbind () [inline, inherited]**

Use this function of unbind an existing [OsclSharedPtr](#).

References [OsclSharedPtr< TheClass >::Bind\(\)](#), and [NULL](#).

**6.2.5.63 template<class TheClass> virtual OsclSharedPtr< TheClass >::~OsclSharedPtr () [inline, virtual, inherited]**

Destructor.

References [NULL](#), and [OsclRefCounter::removeRef\(\)](#).

### 6.2.6 Variable Documentation

- 6.2.6.1 **const int CTIME\_BUFFER\_SIZE = 26**
- 6.2.6.2 **const int ISO8601TIME\_BUFFER\_SIZE = 21**
- 6.2.6.3 **const long MSEC\_PER\_SEC = 1000**
- 6.2.6.4 **const uint32 OSCL\_TLS\_ID\_BASE\_LAST = 11**
- 6.2.6.5 **const uint32 OSCL\_TLS\_ID\_ERRORHOOK = 1**
- 6.2.6.6 **const uint32 OSCL\_TLS\_ID\_MAGICNUM = 0**
- 6.2.6.7 **const uint32 OSCL\_TLS\_ID\_OSCLREGISTRY = 10**
- 6.2.6.8 **const uint32 OSCL\_TLS\_ID\_PAYLOADPARSER = 7**
- 6.2.6.9 **const uint32 OSCL\_TLS\_ID\_PVERRORTRAP = 5**
- 6.2.6.10 **const uint32 OSCL\_TLS\_ID\_PVLOGGER = 2**
- 6.2.6.11 **const uint32 OSCL\_TLS\_ID\_PVMFRECOGNIZER = 8**
- 6.2.6.12 **const uint32 OSCL\_TLS\_ID\_PVSCHEDULER = 4**
- 6.2.6.13 **const uint32 OSCL\_TLS\_ID\_SDPMEDIAPARSER = 6**
- 6.2.6.14 **const uint32 OSCL\_TLS\_ID\_SQLITE3 = 11**
- 6.2.6.15 **const uint32 OSCL\_TLS\_ID\_TEST = 3**
- 6.2.6.16 **const uint32 OSCL\_TLS\_ID\_WMDRM = 9**
- 6.2.6.17 **const int PV8601TIME\_BUFFER\_SIZE = 21**
- 6.2.6.18 **const uint32 unix\_ntp\_offset = 2208988800U**
- 6.2.6.19 **const long USEC\_PER\_SEC = 1000000**

## 6.3 OSCL Memory

### Data Structures

- class [OsclMem](#)
- class [OsclAuditCB](#)
- class [OsclMemAllocator](#)
- class [OsclMemBasicAllocator](#)
- class [OsclMemAllocDestructDealloc< T >](#)
- class [OsclMemBasicAllocDestructDealloc< T >](#)
- class [OsclMemGlobalAuditObject](#)
- class [HeapBase](#)
- struct [MM\\_Stats\\_t](#)
- struct [MM\\_FailInsertParam](#)
- class [OsclMemStatsNode](#)
- struct [MM\\_Stats\\_CB](#)
- struct [MM\\_AllocQueryInfo](#)
- struct [MM\\_AllocInfo](#)
- struct [MM\\_AllocNode](#)
- struct [MM\\_AuditOverheadStats](#)
- class [MM\\_Audit\\_Imp](#)
- class [OsclMemAudit](#)
- struct [MM\\_AllocBlockHdr](#)
- struct [MM\\_AllocBlockFence](#)
- class [OSCLMemAutoPtr< T, \\_Allocator >](#)

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

- class [OsclMemPoolFixedChunkAllocatorObserver](#)
- class [OsclMemPoolFixedChunkAllocator](#)
- class [OsclMemPoolResizableAllocatorObserver](#)
- class [OsclMemPoolResizableAllocatorMemoryObserver](#)
- class [OsclMemPoolResizableAllocator](#)
- class [allocator](#)

### Files

- file [oscl\\_mem\\_basic\\_functions.h](#)

*This file contains prototypes for the basic memory functions.*

- 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\\_mempool.h](#)

*This file contains the definition of memory pool allocators.*

## Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #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\_malloc(count)
- #define [oscl\\_malloc\(a\)](#) OSCL\_MALLOC(a)
- #define [OSCL\\_DEFAULT\\_MALLOC\(x\)](#) OSCL\_MALLOC(x)
- #define [OSCL\\_AUDIT\\_MALLOC\(auditCB, count\)](#) \_oscl\_malloc(count)
- #define [OSCL\\_CALLOC\(num, size\)](#) \_oscl\_calloc(num,size)
- #define [oscl\\_calloc\(a, b\)](#) OSCL\_CALLOC(a,b)
- #define [OSCL\\_AUDIT\\_CALLOC\(auditCB, num, size\)](#) \_oscl\_calloc(num,size)
- #define [OSCL\\_REALLOC\(ptr, new\\_size\)](#) \_oscl\_realloc(ptr,new\_size)
- #define [oscl\\_realloc\(a, b\)](#) OSCL\_REALLOC(a,b)
- #define [OSCL\\_AUDIT\\_REALLOC\(auditCB, ptr, new\\_size\)](#) \_oscl\_realloc(ptr,new\_size)
- #define [OSCL\\_FREE\(ptr\)](#) \_oscl\_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\_new(sizeof(T)),\_oscl\_free,T\_ptr,T,params)
- #define [OSCL\\_AUDIT\\_NEW\(auditCB, T, params\)](#) new(\_oscl\_default\_new(sizeof(T))) T params
- #define [OSCL\\_TRAP\\_AUDIT\\_NEW\(T\\_ptr, auditCB, T, params\)](#) \_OSCL\_TRAP\_NEW(\_oscl\_default\_new(sizeof(T)),\_oscl\_free,T\_ptr,T,params)
- #define [OSCL\\_DELETE\(ptr\)](#)
- #define [OSCL\\_AUDIT\\_ARRAY\\_NEW\(auditCB, T, count\)](#) new(\_oscl\_default\_new(sizeof(T)\*(count))) 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 [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
- #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
- #define OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE
- #define OSCL\_DISABLE\_WARNING\_RETURN\_TYPE\_NOT\_UDT

## Typedefs

- typedef OSCLMemAutoPtr< char, Oscl\_TAlloc< char, OsclMemBasicAllocator > > MMAuditCharAutoPtr
- typedef OSCLMemAutoPtr< uint8, Oscl\_TAlloc< uint8, \_OsclBasicAllocator > > MMAuditUInt8AutoPtr
- 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

## 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\_default\_new (size\_t nBytes)

- void \* **operator new** (size\_t aSize)
- void **operator delete** (void \*aPtr)
- void \* **operator new[ ]** (size\_t aSize)
- void **operator delete[ ]** (void \*aPtr)

## Variables

- static const uint32 **MM\_AllocBlockHdr::ALLOC\_NODE\_FLAG** = 0x80000000

### 6.3.1 Define Documentation

#### 6.3.1.1 #define \_OSCL\_CLEANUP\_BASE\_CLASS(T) this->T::~T()

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

##### Parameters

*T*,: base class name.

#### 6.3.1.2 #define \_OSCL\_TRAP\_NEW(exp, freeFunc, T\_ptr, T, params)

##### Value:

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

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

##### Parameters

*exp*,: expression to allocate memory.

*Tptr:variable* to hold result.

*T*,: type

*params*,: constructor arg list

*freeFunc*,: delete or free function.

---

**6.3.1.3 #define COMPUTE\_MEM\_ALIGN\_SIZE(x, y, z) (y+(((x+y)%z) ? (z - (x+y)%z) : 0))**

**6.3.1.4 #define DEFAULT\_MM\_AUDIT\_MODE 0**

**6.3.1.5 #define DEFAULT\_POSTFILL\_PATTERN 0x5A**

**6.3.1.6 #define DEFAULT\_PREFILL\_PATTERN 0x96**

**6.3.1.7 #define FENCE\_PATTERN 0xAA**

Referenced by MM\_AllocBlockFence::check\_fence(), and MM\_AllocBlockFence::fill\_fence().

**6.3.1.8 #define MEM\_ALIGN\_SIZE 8**

**6.3.1.9 #define MIN\_FENCE\_SIZE 4**

**6.3.1.10 #define MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN 128**

**6.3.1.11 #define MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN 64**

**6.3.1.12 #define MM\_AUDIT\_ALLOC\_NODE\_ENABLE\_FLAG 0x10**

**6.3.1.13 #define MM\_AUDIT\_ALLOC\_NODE\_SUPPORT 1**

**6.3.1.14 #define MM\_AUDIT\_FAILURE\_SIMULATION\_SUPPORT 1**

**6.3.1.15 #define MM\_AUDIT\_FENCE\_SUPPORT 0**

**6.3.1.16 #define MM\_AUDIT\_FILL\_SUPPORT 0**

**6.3.1.17 #define MM\_AUDIT\_INCLUDE\_ALL\_HEAP\_VALIDATION 1**

**6.3.1.18 #define MM\_AUDIT\_POSTFILL\_FLAG 0x2**

**6.3.1.19 #define MM\_AUDIT\_PREFILL\_FLAG 0x1**

**6.3.1.20 #define MM\_AUDIT\_SUPPRESS\_FILENAME\_FLAG 0x20**

**6.3.1.21 #define MM\_AUDIT\_VALIDATE\_ALL\_HEAP\_FLAG 0x4**

**6.3.1.22 #define MM\_AUDIT\_VALIDATE\_BLOCK 1**

**6.3.1.23 #define MM\_AUDIT\_VALIDATE\_ON\_FREE\_FLAG 0x8**

**6.3.1.24 #define OSCL\_ALLOC\_DELETE(ptr, T\_allocator, T)**

**Value:**

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

Deletes the object of type T using the given allocator

#### Parameters

*T\_allocator* allocator for objects of type T

*T* type of object to delete

*ptr* pointer to previously created object

#### Exceptions

*none,unless* thrown by the given allocator

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

\*\*\*\*\* Macros for new/delete with a given allocator/deallocator. Creates an object of type T using the given allocator to acquire the memory needed.

#### Parameters

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

*T* type of object to create

*params* object initialization parameters

#### Returns

pointer to created object

#### Exceptions

*none,unless* thrown by the given allocator

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

Oscl array delete operator..

#### Parameters

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

#### Returns

void

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

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

#### Parameters

*T* data type for 'new' operation

*count* number of elements to create

#### Returns

pointer to the newly created object array of type T

#### Exceptions

*may* leave with code = bad alloc

### 6.3.1.28 #define OSCL\_AUDIT\_ARRAY\_NEW(auditCB, T, count) new(\_oscl\_default\_- new(sizeof(T)\*(count))) T

\*\*\*\*\* Macros for array new/delete with memory management. These only work for simple array types and cannot be used for class types with constructor/destructors.

Note: some compilers do not support placement array new operator, so these macros don't use it. Oscl array "new" operator. This uses the input memory audit object.

#### Parameters

*auditCB* input memory management audit object

*T* data type for 'new' operation

*count* number of elements to create

#### Returns

pointer to the newly created object array of type T

#### Exceptions

*may* leave with code = bad alloc

### 6.3.1.29 #define OSCL\_AUDIT\_CALLOC(auditCB, num, size) \_oscl\_calloc(num,size)

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

#### Parameters

*auditCB* input memory management audit object

*num* number of elements

*size* number of bytes to allocate for each element

#### Returns

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

#### Exceptions

*none*

**6.3.1.30 #define OSCL\_AUDIT\_MALLOC(auditCB, count) \_oscl\_malloc(count)**

Allocates a memory block using the given audit object.

**Parameters**

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

**Returns**

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

**Exceptions**

*none*

**6.3.1.31 #define OSCL\_AUDIT\_NEW(auditCB, T, params) new(\_oscl\_default\_new(sizeof(T))) T  
params**

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

**Parameters**

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

**Returns**

pointer to the newly created object of type T

**Exceptions**

*may* leave with code = bad alloc

**6.3.1.32 #define OSCL\_AUDIT\_REALLOC(auditCB, ptr, new\_size) \_oscl\_realloc(ptr,new\_size)**

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

**Parameters**

*auditCB* input memory management audit object  
*ptr* original memory block  
*new\_size* New size of the block

**Returns**

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

**Exceptions**

*none*

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

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

**Parameters**

*num* number of elements

*size* number of bytes to allocate for each element

**Returns**

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

**Exceptions**

*none*

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

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

**Parameters**

*T*: name of the base class.

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

Another back-compatibility definition.

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

Another back-compatibility definition.

**6.3.1.38 #define OSCL\_DELETE(ptr)****Value:**

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

Oscl "delete" operator.

**Parameters**

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

**Returns**

void

Referenced by OsclBuf::Delete(), OsclBuf::NewL(), MM\_AllocNode::~MM\_AllocNode(), and OsclMemStatsNode::~OsclMemStatsNode().

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

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

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

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

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

Previously this was in oscl\_mem\_imp.h

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

Previously this was in oscl\_mem\_imp.h

**6.3.1.45 #define oscl\_free(x) OSCL\_FREE(x)**

**6.3.1.46 #define OSCL\_FREE(ptr) \_oscl\_free(ptr)**

Deallocates or frees a memory block.

**Parameters**

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

Referenced by OsclMemAllocator::deallocate(), and OsclBuf::Delete().

**6.3.1.47 #define oscl\_malloc(a) OSCL\_MALLOC(a)**

**6.3.1.48 #define OSCL\_MALLOC(count) \_oscl\_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*

Referenced by OsclBuf::NewL().

### 6.3.1.49 #define OSCL\_NEW(T, params) new T params

\*\*\*\*\* Macros for new/delete with memory management. 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

Referenced by OsclBuf::NewL().

### 6.3.1.50 #define OSCL\_PLACEMENT\_NEW(ptr, constructor) new(ptr) constructor

Referenced by OsclTimer< Alloc >::OsclTimer().

### 6.3.1.51 #define oscl\_realloc(a, b) OSCL\_REALLOC(a,b)

### 6.3.1.52 #define OSCL\_REALLOC(ptr, new\_size) \_oscl\_realloc(ptr,new\_size)

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

#### Parameters

*ptr* original memory block  
*new\_size* New size of the block

#### Returns

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

#### Exceptions

*none*

### 6.3.1.53 #define OSCL\_TRAP\_ALLOC\_NEW(T\_ptr, T\_allocator, T, params) \_OSCL\_TRAP\_NEW(T\_allocator.allocate(1),T\_allocator.deallocate,T\_ptr,T,params)

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

#### Parameters

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

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

*T* type of object to create

*params* object initialization parameters

#### Returns

pointer to created object

#### Exceptions

*none,unless* thrown by the given allocator

**6.3.1.54 #define OSCL\_TRAP\_AUDIT\_NEW(T\_ptr, auditCB, T, params) \_OSCL\_TRAP\_NEW(\_oscl\_default\_new(sizeof(T)),\_oscl\_free,T\_ptr,T,params)**

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

#### Parameters

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

*auditCB* input memory management audit object

*T* data type for 'new' operation

*params* object initialization parameters

#### Returns

pointer to the newly created object of type T

#### Exceptions

*may* leave with code = bad alloc

**6.3.1.55 #define OSCL\_TRAP\_NEW(T\_ptr, T, params) \_OSCL\_TRAP\_NEW(\_oscl\_default\_new(sizeof(T)),\_oscl\_free,T\_ptr,T,params)**

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

#### Parameters

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

*T* data type for 'new' operation

*params* object initialization parameters

#### Returns

pointer to the newly created object of type T

#### Exceptions

*may* leave with code = bad alloc

### 6.3.2 Typedef Documentation

- 6.3.2.1 **typedef OSCLMemAutoPtr<MM\_AllocNode, Oscl\_TAlloc<MM\_AllocNode, OsclMemBasicAllocator> > MM\_AllocNodeAutoPtr**
- 6.3.2.2 **typedef OSCLMemAutoPtr<OsclMemStatsNode, Oscl\_TAlloc<OsclMemStatsNode, OsclMemBasicAllocator> > MM\_StatsNodeTagTreeType**
- 6.3.2.3 **typedef OSCLMemAutoPtr<char, Oscl\_TAlloc<char, OsclMemBasicAllocator> > MMAuditCharAutoPtr**
- 6.3.2.4 **typedef OSCLMemAutoPtr<uint8, Oscl\_TAlloc<uint8, \_OsclBasicAllocator> > MMAuditUInt8AutoPtr**
- 6.3.2.5 **typedef OSCLMemAutoPtr<OsclMemStatsNode, Oscl\_TAlloc<OsclMemStatsNode, OsclMemBasicAllocator> > OsclMemStatsNodeAutoPtr**
- 6.3.2.6 **typedef Oscl\_TagTree<MM\_StatsNodeTagTreeType, TagTree\_Allocator> OsclTagTreeType**
- 6.3.2.7 **typedef Oscl\_TAlloc<MM\_StatsNodeTagTreeType, OsclMemBasicAllocator> TagTree\_Allocator**

### 6.3.3 Function Documentation

6.3.3.1 **OSCL\_COND\_IMPORT\_REF void\* \_oscl\_calloc (int32 *nelems*, int32 *size*)**

6.3.3.2 **OSCL\_IMPORT\_REF void\* \_oscl\_default\_new (size\_t *nBytes*)**

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

Referenced by HeapBase::operator new(), operator new[], HeapBase::operator new[ ](), and operator new[ ]().

6.3.3.3 **OSCL\_COND\_IMPORT\_REF void \_oscl\_free (void \* *src*)**

Referenced by OsclMemBasicAllocator::deallocate(), HeapBase::operator delete(), operator delete[], HeapBase::operator delete[ ](), and operator delete[ ]().

6.3.3.4 **OSCL\_COND\_IMPORT\_REF void\* \_oscl\_malloc (int32 *count*)**

Referenced by OsclMemBasicAllocator::allocate(), and OsclMemAllocator::allocate().

6.3.3.5 **OSCL\_COND\_IMPORT\_REF void\* \_oscl\_realloc (void \* *src*, int32 *count*)**

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

References \_oscl\_free().

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

References `_oscl_free()`.

**6.3.3.8 void\* operator new (size\_t *aSize*) [inline]**

References `_oscl_default_new()`.

**6.3.3.9 void\* operator new[ ] (size\_t *aSize*) [inline]**

References `_oscl_default_new()`.

**6.3.3.10 OSCL\_COND\_IMPORT\_REF uint oscl\_mem\_aligned\_size (uint *size*)**

Get memory-aligned size of an object.

**Parameters**

*size* size of object

**Returns**

memory-aligned size

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

Compare characters in two buffers

**Parameters**

*buf1* first buffer

*buf2* second buffer

*count* number of bytes to compare

**Returns**

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

**6.3.3.12 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

Referenced by MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), OsclUuid::operator=(), and OsclUuid::OsclUuid().

### 6.3.3.13 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

Referenced by BufFragGroup< ChainClass, max\_frags >::AddFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), and OsclPtr::Append().

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

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

#### Parameters

***dest*** new buffer

***src*** buffer to copy

***count*** number of bytes to copy

#### Returns

the value of dest

### 6.3.3.15 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

Referenced by OsclMemBasicAllocator::allocate(), OsclMemAllocator::allocate(), BufFragGroup<ChainClass, max\_frags>::BufFragGroup(), MediaData<ChainClass, max\_frags, local\_bufsize>::Clear(), BufFragGroup<ChainClass, max\_frags>::Clear(), MM\_AllocBlockFence::fill\_fence(), MM\_AllocInfo::MM\_AllocInfo(), MM\_AllocNode::MM\_AllocNode(), MM\_FailInsertParam::MM\_FailInsertParam(), MM\_Stats\_CB::MM\_Stats\_CB(), MM\_Stats\_t::MM\_Stats\_t(), OsclUuid::OsclUuid(), MM\_FailInsertParam::reset(), and MM\_Stats\_t::reset().

#### 6.3.3.16 OSCL\_IMPORT\_REF void OsclMemInit (OsclAuditCB & *auditCB*)

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

### Parameters

*auditCB* memory management audit object

### 6.3.4 Variable Documentation

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

Referenced by MM\_AllocBlockHdr::isAllocNodePtr(), and MM\_AllocBlockHdr::setAllocNodeFlag().

## 6.4 OSCL Util

### Data Structures

- class [OsclRegistryAccessClient](#)
- class [OsclRegistryClient](#)
- class [OsclRegistryClientImpl](#)
- class [OsclRegistryAccessClientImpl](#)
- class [OsclRegistryClientTlsImpl](#)
- class [OsclRegistryAccessClientTlsImpl](#)
- class [OsclRegistryAccessElement](#)
- class [OsclComponentRegistryElement](#)
- class [OsclComponentRegistryData](#)
- class [OsclComponentRegistry](#)
- class [OsclRegistryServTlsImpl](#)
- class [OsclBinStream](#)
- class [OsclBinIStream](#)
- class [OsclBinIStreamLittleEndian](#)
- class [OsclBinIStreamBigEndian](#)
- class [OsclBinOStream](#)

*Class [OsclBinOStream](#) implements the basic stream functions for an output stream.*

- class [OsclBinOStreamLittleEndian](#)

*Class [OsclBinOStreamLittleEndian](#) implements a binary output stream using little endian byte ordering.*

- class [OsclBinOStreamBigEndian](#)

*Class [OsclBinOStreamBigEndian](#) implements a binary output stream using big endian byte ordering.*

- class [MemAllocator< T >](#)
- class [BufferMgr](#)
- class [BufferState](#)
- class [BufferFragment](#)
- class [BufFragGroup< ChainClass, max\\_frags >](#)
- class [MediaData< ChainClass, max\\_frags, local\\_bufsize >](#)
- class [BufFragStatusClass](#)
- class [MediaStatusClass](#)
- class [OsclPriorityQueueBase](#)
- class [OsclCompareLess< T >](#)
- class [OsclPriorityQueue< Qelem, Alloc, Container, Compare >](#)
- class [OsclRand](#)
- 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.*

- struct [StrCSumPtrLen](#)

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

- class [OSCL\\_String](#)

- class [OSCL\\_wString](#)
- class [OSCL\\_HeapString< Alloc >](#)
- class [OSCL\\_wHeapString< Alloc >](#)
- class [OSCL\\_HeapStringA](#)
- class [OSCL\\_wHeapStringA](#)
- class [OSCL\\_StackString< MaxBufSize >](#)
- class [OSCL\\_wStackString< MaxBufSize >](#)
- class [OSCL\\_FastString](#)
- class [OSCL\\_wFastString](#)
- class [CHheapRep](#)
- class [CStackRep](#)
- class [CFastRep](#)
- class [OsclTickCount](#)

## Files

- file [oscl\\_string\\_utils.h](#)  
*Utilities to parse and convert strings.*
- 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\\_types.h](#)  
*Common types used in Oscl registry interfaces.*
- file [oscl\\_registry\\_serv\\_impl.h](#)  
*Server-side implementation of OsclRegistry interfaces.*
- file [oscl\\_registry\\_serv\\_impl.h](#)  
*Server-side implementation of OsclRegistry interfaces.*
- file [oscl\\_registry\\_serv\\_impl.h](#)  
*Server-side implementation of OsclRegistry interfaces.*
- file [oscl\\_bin\\_stream.h](#)  
*Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.*
- file [oscl\\_math.h](#)  
*Provides math functions.*
- file [oscl\\_media\\_data.h](#)  
*Defines a container class for media data made up of a collection of memory fragments.*

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

*Defines a status values for the [MediaData](#) containers.*

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

*Implements a priority queue data structure similar to STL.*

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

*Provides pseudo-random number generation.*

- file [oscl\\_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\\_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.*

## Defines

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

## Typedefs

- `typedef OsclAny * OsclComponentFactory`
- `typedef void(* BufferFreeFuncPtr )(void *)`
- `typedef uint32 MediaTimestamp`
- `typedef struct StrPtrLen StrPtrLen`

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

- `typedef struct WStrPtrLen WStrPtrLen`

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

- `typedef StrCSumPtrLen StrCSumPtrLen`

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

- `typedef WStrPtrLen OSCL_TStrPtrLen`

## Enumerations

- `enum TOSCL_StringOp { EOSCL_StringOp_CompressASCII, EOSCL_StringOp_UTF16ToUTF8 }`
- `enum TOSCL_wStringOp { EOSCL_wStringOp_ExpandASCII, EOSCL_wStringOp_UTF8ToUTF16 }`

## Functions

- `OSCL_IMPORT_REF const char * skip_whitespace (const char *ptr)`
- `OSCL_IMPORT_REF char * skip_whitespace (char *ptr)`
- `OSCL_IMPORT_REF const char * skip_whitespace (const char *start, const char *end)`
- `OSCL_IMPORT_REF const char * skip_to_whitespace (const char *start, const char *end)`
- `OSCL_IMPORT_REF const char * skip_to_line_term (const char *start_ptr, const char *end_ptr)`
- `OSCL_IMPORT_REF const char * skip_whitespace_and_line_term (const char *start, const char *end)`
- `OSCL_IMPORT_REF int extract_string (const char *in_ptr, char *outstring, int maxsize)`
- `OSCL_IMPORT_REF int extract_string (const char *start, const char *end, char *outstring, int maxsize)`
- `OSCL_IMPORT_REF bool PV_atoi (const char *buf, const char new_format, uint32 &value)`
- `OSCL_IMPORT_REF bool PV_atoi (const char *buf, const char new_format, int length, uint32 &value)`
- `OSCL_IMPORT_REF bool PV_atoi (const char *buf, const char new_format, int length, uint64 &value)`
- `OSCL_IMPORT_REF bool PV_atof (const char *buf, OsclFloat &value)`
- `OSCL_IMPORT_REF bool PV_atof (const char *buf, int length, OsclFloat &value)`
- `OSCL_IMPORT_REF int oscl_abs (int aVal)`
- `OSCL_COND_IMPORT_REF double oscl_log (double value)`
- `OSCL_COND_IMPORT_REF double oscl_log10 (double value)`
- `OSCL_COND_IMPORT_REF double oscl_sqrt (double value)`
- `OSCL_COND_IMPORT_REF double oscl_pow (double x, double y)`
- `OSCL_COND_IMPORT_REF double oscl_exp (double value)`
- `OSCL_COND_IMPORT_REF double oscl_sin (double value)`
- `OSCL_COND_IMPORT_REF double oscl_cos (double value)`

- OSCL\_COND\_IMPORT\_REF double `oscl_tan` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_asin` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_atan` (double value)
- OSCL\_COND\_IMPORT\_REF double `oscl_floor` (double value)
- OSCL\_IMPORT\_REF int32 `oscl_snprintf` (char \*str, uint32 count, const char \*fmt,...)
- OSCL\_IMPORT\_REF int32 `oscl_snprintf` (`oscl_wchar` \*str, uint32 count, const `oscl_wchar` \*fmt,...)
- OSCL\_IMPORT\_REF int32 `oscl_vsnprintf` (char \*str, uint32 count, const char \*fmt, va\_list args)
- OSCL\_IMPORT\_REF int32 `oscl_vsnprintf` (`oscl_wchar` \*str, uint32 count, const `oscl_wchar` \*fmt, va\_list args)
- OSCL\_IMPORT\_REF bool `oscl_str_unescape_uri` (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes, uint32 &out\_buf\_len)
 

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

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

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

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

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

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

*Convert UTF8 byte sequence to Unicode UTF-16 string.*
- OSCL\_IMPORT\_REF int32 `oscl_UnicodeToUTF8` (const `oscl_wchar` \*input, int32 inLength, char \*output, int32 outLength)
 

*Convert UTF-16 Unicode string to UTF8 byte sequence.*
- `BufferFragment` \* `BufFragGroup::GetFragment` (const int32 idx)
- `BufferState` \* `BufFragGroup::GetBufferState` (const int32 idx)
- uint32 `OSCL_HeapString::get_size` () const
- uint32 `OSCL_wHeapString::get_size` () const
- uint32 `OSCL_HeapString::get_maxsize` () const

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

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

## Variables

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

### 6.4.1 Define Documentation

#### 6.4.1.1 #define MAX\_NUMBER\_OF\_BYTE\_PER\_UTF8 3

Define the maximum UTF8 representation in bytes.

##### **Todo**

Handle 4-byte surrogate pair representation

#### 6.4.1.2 #define oscl\_isdigit(c) ((c) >= '0' && (c) <= '9')

#### 6.4.1.3 #define OSCLTICKCOUNT\_MAX\_TICKS 0xffffffff

### 6.4.2 Typedef Documentation

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

#### 6.4.2.2 typedef uint32 MediaTimestamp

#### 6.4.2.3 typedef WStrPtrLen OSCL\_TStrPtrLen

#### 6.4.2.4 typedef OsclAny\* OsclComponentFactory

OsclComponentFactory is an opaque pointer.

#### 6.4.2.5 typedef StrCSumPtrLen StrCSumPtrLen

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

#### 6.4.2.6 `typedef struct StrPtrLen StrPtrLen`

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

#### 6.4.2.7 `typedef struct WStrPtrLen WStrPtrLen`

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

### 6.4.3 Enumeration Type Documentation

#### 6.4.3.1 `enum TOSCL_StringOp`

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

**Enumerator:**

*EOSCL\_StringOp\_CompressASCII*  
*EOSCL\_StringOp\_UTF16ToUTF8*

#### 6.4.3.2 `enum TOSCL_wStringOp`

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

**Enumerator:**

*EOSCL\_wStringOp\_ExpandASCII*  
*EOSCL\_wStringOp\_UTF8ToUTF16*

### 6.4.4 Function Documentation

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

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

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

Implements [OSCL\\_wString](#).

References CStackRep::buffer.

Referenced by OSCL\_wStackString<MaxBufSize>::set().

---

**6.4.4.4 template<uint32 MaxBufSize> const OSCL\_StackString< MaxBufSize >::chartype \* OSCL\_StackString< MaxBufSize >::get\_cstr () const [inline, virtual, inherited]**

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

Implements [OSCL\\_String](#).

References CStackRep::buffer.

Referenced by OSCL\_StackString< MaxBufSize >::set().

**6.4.4.5 template<class Alloc > const OSCL\_wHeapString< Alloc >::chartype \* OSCL\_wHeapString< Alloc >::get\_cstr () const [inline, virtual, inherited]**

Implements [OSCL\\_wString](#).

References CHheapRep::buffer, and NULL.

Referenced by OSCL\_wHeapString< Alloc >::set().

**6.4.4.6 template<class Alloc > const OSCL\_HeapString< Alloc >::chartype \* OSCL\_HeapString< Alloc >::get\_cstr () const [inline, virtual, inherited]**

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

Implements [OSCL\\_String](#).

References CHheapRep::buffer, and NULL.

Referenced by OSCL\_HeapString< Alloc >::set().

**6.4.4.7 template<uint32 MaxBufSize> uint32 OSCL\_wStackString< MaxBufSize >::get\_maxsize () const [inline, virtual, inherited]**

Implements [OSCL\\_wString](#).

References CStackRep::maxsize.

**6.4.4.8 template<uint32 MaxBufSize> uint32 OSCL\_StackString< MaxBufSize >::get\_maxsize () const [inline, 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](#).

References CStackRep::maxsize.

**6.4.4.9 template<class Alloc > uint32 OSCL\_wHeapString< Alloc >::get\_maxsize () const [inline, virtual, inherited]**

Implements [OSCL\\_wString](#).

References CHheapRep::maxsize.

**6.4.4.10 template<class Alloc > uint32 OSCL\_HeapString< Alloc >::get\_maxsize () const [inline, 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](#).

References CHeapRep::maxsize.

**6.4.4.11 template<uint32 MaxBufSize> uint32 OSCL\_wStackString< MaxBufSize >::get\_size () const [inline, virtual, inherited]**

Implements [OSCL\\_wString](#).

References CStackRep::size.

**6.4.4.12 template<uint32 MaxBufSize> uint32 OSCL\_StackString< MaxBufSize >::get\_size () const [inline, virtual, inherited]**

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

Implements [OSCL\\_String](#).

References CStackRep::size.

**6.4.4.13 template<class Alloc > uint32 OSCL\_wHeapString< Alloc >::get\_size () const [inline, virtual, inherited]**

Implements [OSCL\\_wString](#).

References CHeapRep::size.

**6.4.4.14 template<class Alloc > uint32 OSCL\_HeapString< Alloc >::get\_size () const [inline, virtual, inherited]**

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

Implements [OSCL\\_String](#).

References CHeapRep::size.

**6.4.4.15 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::chartype \* OSCL\_wStackString< MaxBufSize >::get\_str () const [inline, virtual, inherited]**

Implements [OSCL\\_wString](#).

References CStackRep::buffer.

---

**6.4.4.16 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::chartype \* OSCL\_StackString< MaxBufSize >::get\_str () const [inline, virtual, inherited]**

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

Implements [OSCL\\_String](#).

References CStackRep::buffer.

**6.4.4.17 template<class Alloc > OSCL\_wHeapString< Alloc >::chartype \* OSCL\_wHeapString< Alloc >::get\_str () const [inline, virtual, inherited]**

Implements [OSCL\\_wString](#).

References CHheapRep::buffer, and NULL.

**6.4.4.18 template<class Alloc > OSCL\_HeapString< Alloc >::chartype \* OSCL\_HeapString< Alloc >::get\_str () const [inline, virtual, inherited]**

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

Implements [OSCL\\_String](#).

References CHheapRep::buffer, and NULL.

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

References BufFragGroup< ChainClass, max\_frags >::buffer\_states, NULL, and BufFragGroup< ChainClass, max\_frags >::num\_frgments.

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

References BufFragGroup< ChainClass, max\_frags >::fragments, NULL, and BufFragGroup< ChainClass, max\_frags >::num\_frgments.

**6.4.4.21 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize > & OSCL\_wStackString< MaxBufSize >::operator= (const chartype \* cstr) [inline, inherited]**

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

References OSCL\_wString::set\_rep().

**6.4.4.22 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize > & OSCL\_StackString< MaxBufSize >::operator= (const chartype \* cstr) [inline, inherited]**

Assignment operator

**Parameters**

null-terminated string

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

References OSCL\_wString::set\_rep().

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

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

References OSCL\_wString::set\_rep().

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

Assignment operator

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

References OSCL\_wString::set\_rep().

**6.4.4.25 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize > &  
OSCL\_wStackString< MaxBufSize >::operator= (const OSCL\_wStackString< MaxBufSize > & src) [inline, inherited]**

References OSCL\_wString::set\_rep().

**6.4.4.26 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize > &  
OSCL\_StackString< MaxBufSize >::operator= (const OSCL\_StackString< MaxBufSize > & src) [inline, inherited]**

Assignment operators

References OSCL\_wString::set\_rep().

**6.4.4.27 template<class Alloc > OSCL\_wHeapString< Alloc > & OSCL\_wHeapString< Alloc >::operator= (const chartype \* cstr) [inline, inherited]**

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

References OSCL\_wString::set\_rep().

**6.4.4.28 template<class Alloc > OSCL\_HeapString< Alloc > & OSCL\_HeapString< Alloc >::operator= (const chartype \* cstr) [inline, inherited]**

Assignment operator

**Parameters**

null-terminated string

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

References OSCL\_wString::set\_rep().

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

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

References OSCL\_wString::set\_rep().

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

Assignment operator

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

References OSCL\_wString::set\_rep().

**6.4.4.31 template<class Alloc > OSCL\_wHeapString< Alloc > & OSCL\_wHeapString< Alloc >::operator= (const OSCL\_wHeapString< Alloc > & src) [inline, inherited]**

References CHeapRep::assign(), and OSCL\_wString::set\_rep().

**6.4.4.32 template<class Alloc > OSCL\_HeapString< Alloc > & OSCL\_HeapString< Alloc >::operator= (const OSCL\_HeapString< Alloc > & src) [inline, inherited]**

Assignment operators

References CHeapRep::assign(), and OSCL\_wString::set\_rep().

**6.4.4.33 OSCL\_IMPORT\_REF int oscl\_abs (int aVal)**

**6.4.4.34 OSCL\_COND\_IMPORT\_REF double oscl\_asin (double value)**

Calculates the arc sine of a number

**Parameters**

*value* source value

**6.4.4.35 OSCL\_COND\_IMPORT\_REF double oscl\_atan (double value)**

Calculates the arc tangent of a number

**Parameters**

*value* source value

**6.4.4.36 OSCL\_COND\_IMPORT\_REF double oscl\_cos (double *value*)**

Calculates the cosine of a number

**Parameters**

*value* source value

**6.4.4.37 OSCL\_COND\_IMPORT\_REF double oscl\_exp (double *value*)**

Calculates the exponential of e for a number

**Parameters**

*value* source value

**6.4.4.38 OSCL\_COND\_IMPORT\_REF double oscl\_floor (double *value*)**

Calculates the floor of a number

**Parameters**

*value* source value

**6.4.4.39 template<class Alloc > OSCL\_HeapString< Alloc >::OSCL\_HeapString (const OSCL\_String & *src*) [inline, inherited]**

References OSCL\_wString::set\_rep().

**6.4.4.40 template<class Alloc > OSCL\_HeapString< Alloc >::OSCL\_HeapString (const OSCL\_HeapString< Alloc > & *src*) [inline, inherited]**

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

**Parameters**

*src*,: input string.

References CHeapRep::assign(), and OSCL\_wString::set\_rep().

**6.4.4.41 template<class Alloc > OSCL\_HeapString< Alloc >::OSCL\_HeapString (const chartype \* *buf*, uint32 *length*) [inline, inherited]**

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

**Parameters**

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

*length*,: number of characters to copy.

---

**6.4.4.42 template<class Alloc > OSCL\_HeapString< Alloc >::OSCL\_HeapString (const chartype \* *cstr*) [inline, inherited]**

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

**Parameters**

*cp*,: null-terminated string.

References OSCL\_wString::set\_rep().

**6.4.4.43 template<class Alloc > OSCL\_HeapString< Alloc >::OSCL\_HeapString () [inline, inherited]**

The default constructor creates an empty string.

References NULL, and OSCL\_wString::set\_rep().

**6.4.4.44 OSCL\_COND\_IMPORT\_REF double oscl\_log (double *value*)**

Calculates the natural log of a number

**Parameters**

*value* source value

**6.4.4.45 OSCL\_COND\_IMPORT\_REF double oscl\_log10 (double *value*)**

Calculates tthe logarithm to base 10 of a number

**Parameters**

*value* source value

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

Calculates the value of x to the power of y

**Parameters**

*x* base value

*y* power

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

Calculates the sine of a number

**Parameters**

*value* source value

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

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

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

Calculates the square root of a number

#### Parameters

*value* source value

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

References OSCL\_wString::set\_rep().

**6.4.4.52 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::OSCL\_StackString (const OSCL\_StackString< MaxBufSize > &src) [inline, 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.

References OSCL\_wString::set\_rep().

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

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

#### Parameters

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

*length*,: the number of characters to copy.

**6.4.4.54 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::OSCL\_StackString (const chartype \*cstr) [inline, 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.

References OSCL\_wString::set\_rep().

---

**6.4.4.55 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::OSCL\_StackString  
()** [inline, inherited]

Creates an [OSCL\\_StackString](#) initialized with an empty string.

References NULL, and OSCL\_wString::set\_rep().

**6.4.4.56 OSCL\_IMPORT\_REF int32 oscl\_str\_escape\_xml (const char \* str\_buf\_in, char  
\* str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes = 0, uint32 \*  
num\_bytes\_written = NULL)**

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

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

#### Parameters

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

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

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

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

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

#### Returns

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

**6.4.4.57 OSCL\_IMPORT\_REF bool oscl\_str\_is\_valid\_utf8 (const uint8 \* str\_buf, uint32 &  
num\_valid\_characters, uint32 max\_bytes = 0, uint32 max\_char\_2\_valid = 0, uint32 \*  
num\_byte\_4\_char = NULL)**

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

### UTF-8 String Manipulation

These routines operate on UTF-8 character string.

#### Parameters

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

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

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

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

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

### Returns

True if the string is valid and false otherwise.

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

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

### XML String Manipulation

These routines handle the special characters, which needs to be escaped, for xml document.

#### Parameters

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

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

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

#### Returns

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

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

Truncates the UTF-8 string upto the required size.

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

#### Parameters

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

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

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

#### Returns

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

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

unescape any of the special escape sequence in the uri string

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

##### Parameters

*oscl\_str\_in* Ptr to an input [OSCL\\_String](#)

*oscl\_str\_out* Ptr to an output [OSCL\\_String](#) which stores the modified string

*out\_buf\_len* The length of the result string (not including the null character)

##### Returns

It returns true if succeeds, otherwise false.

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

unescape any of the special escape sequence in the uri string

#### URI String Manipualation

These routines handle all of the special escape sequences in the URI.

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

##### Parameters

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

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

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

*max\_bytes* The maximum number of bytes to read. It is the length of str\_buf\_in.

*out\_buf\_len* The length of the result string (not including the null character)

##### Returns

It returns true if succeeds, otherwise false.

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

Calculates the tangential of a number

##### Parameters

*value* source value

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

Convert UTF-16 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 ” at the end of the output UTF8 byte sequence. and returns length of the output UTF8 byte sequence(without counting terminated ”). In case of (B), it converts as much as possible to the output buffer and adds a terminated ” 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. Please note that Unicode character exceeding U+FFFF are not supported.

#### Parameters

*input* Ptr to an input Unicode string. ” termanation is not neccesary.

*inLength* The length of the input Unicode string, without counting terminated ”(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 ”) : completely converts all input string and appends ” to output; 0 : insufficient buffer or error in conversion

#### 6.4.4.64 OSCL\_IMPORT\_REF int32 oscl\_UTF8ToUnicode (const char \* *input*, int32 *inLength*, oscl\_wchar \* *output*, int32 *outLength*)

Convert UTF8 byte sequence to Unicode UTF-16 string.

The function converts UTF8 byte sequence (or ASCII sequence) to UTF-16 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 ” at the end of the output Unicode string, and returns length of the output Unicode string(without counting terminated ”). In case of (B), it converts as much as possible to the output buffer and adds a terminated ” 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. Please note that Unicode character exceeding U+FFFF are not supported.

#### Parameters

*input* Ptr to an input UTF8 byte sequence. ” termanation is not neccesary.

*inLength* The length of the input UTF8 byte sequence, without counting terminated ”(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 ”) : completely converts all input string and appends ” to output; 0 : insufficient buffer or error in conversion

- 6.4.4.65 OSCL\_IMPORT\_REF int32 oscl\_vsnprintf (oscl\_wchar \*str, uint32 count, const oscl\_wchar \*fmt, va\_list args)**
- 6.4.4.66 OSCL\_IMPORT\_REF int32 oscl\_vsnprintf (char \*str, uint32 count, const char \*fmt, va\_list args)**
- 6.4.4.67 template<class Alloc > OSCL\_wHeapString< Alloc >::OSCL\_wHeapString (const OSCL\_wString & src) [inline, inherited]**

References OSCL\_wString::set\_rep().

- 6.4.4.68 template<class Alloc > OSCL\_wHeapString< Alloc >::OSCL\_wHeapString (const OSCL\_wHeapString< Alloc > & src) [inline, inherited]**

References CHeapRep::assign(), and OSCL\_wString::set\_rep().

- 6.4.4.69 template<class Alloc > OSCL\_wHeapString< Alloc >::OSCL\_wHeapString (const chartype \*buf, uint32 length) [inline, inherited]**
- 6.4.4.70 template<class Alloc > OSCL\_wHeapString< Alloc >::OSCL\_wHeapString (const chartype \*cstr) [inline, inherited]**

References OSCL\_wString::set\_rep().

- 6.4.4.71 template<class Alloc > OSCL\_wHeapString< Alloc >::OSCL\_wHeapString () [inline, inherited]**

References NULL, and OSCL\_wString::set\_rep().

- 6.4.4.72 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString (const OSCL\_wString & src) [inline, inherited]**

References OSCL\_wString::set\_rep().

- 6.4.4.73 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString (const OSCL\_wStackString< MaxBufSize > & src) [inline, inherited]**

References OSCL\_wString::set\_rep().

- 6.4.4.74 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString (const chartype \*buf, uint32 length) [inline, inherited]**

- 6.4.4.75 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString (const chartype \*cstr) [inline, inherited]**

References OSCL\_wString::set\_rep().

---

**6.4.4.76 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::OSCL\_wStackString () [inline, inherited]**

References NULL, and OSCL\_wString::set\_rep().

**6.4.4.77 OSCL\_IMPORT\_REF bool PV\_atof (const char \* buf, int length, OsclFloat & value)**

**6.4.4.78 OSCL\_IMPORT\_REF bool PV\_atof (const char \* buf, OsclFloat & value)**

**6.4.4.79 OSCL\_IMPORT\_REF bool PV\_atoi (const char \* buf, const char new\_format, int length, uint64 & value)**

**6.4.4.80 OSCL\_IMPORT\_REF bool PV\_atoi (const char \* buf, const char new\_format, int length, uint32 & value)**

**6.4.4.81 OSCL\_IMPORT\_REF bool PV\_atoi (const char \* buf, const char new\_format, uint32 & value)**

**6.4.4.82 template<uint32 MaxBufSize> void OSCL\_wStackString< MaxBufSize >::set (const other\_chartype \* buf, uint32 length, optype op) [inline, inherited]**

References NULL, OSCL\_wString::setrep\_to\_wide\_char(), and CStackRep::size.

**6.4.4.83 template<uint32 MaxBufSize> void OSCL\_StackString< MaxBufSize >::set (const other\_chartype \* buf, uint32 length, optype op) [inline, inherited]**

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

#### Parameters

*buf*,: string or character array.

*length*,: number of characters to copy.

*op*,: conversion operation to apply

References NULL, OSCL\_String::setrep\_to\_char(), and CStackRep::size.

**6.4.4.84 template<uint32 MaxBufSize> void OSCL\_wStackString< MaxBufSize >::set (const other\_chartype \* buf, optype op) [inline, inherited]**

References NULL, oscl\_strlen(), OSCL\_wString::setrep\_to\_wide\_char(), and CStackRep::size.

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

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

#### Parameters

*buf*,: NULL-terminated wide string.

*op*,: conversion operation to apply

References NULL, oscl\_strlen(), OSCL\_String::setrep\_to\_char(), and CStackRep::size.

---

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

References OSCL\_wStackString< MaxBufSize >::get\_cstr(), oscl\_strlen(), CStackRep::set(), and CStackRep::size.

**6.4.4.87 template<uint32 MaxBufSize> void OSCL\_StackString< MaxBufSize >::set (const chartype \* *buf*, uint32 *length*) [inline, 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.

References OSCL\_StackString< MaxBufSize >::get\_cstr(), oscl\_strlen(), CStackRep::set(), and CStackRep::size.

**6.4.4.88 template<class Alloc > void OSCL\_wHeapString< Alloc >::set (const other\_chartype \* *buf*, uint32 *length*, optype *op*) [inline, inherited]**

References OSCL\_wString::setrep\_to\_wide\_char(), and CHheapRep::size.

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

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

**Parameters**

*buf*,: string or character array.

*length*,: number of characters to copy.

*op*,: conversion operation to apply

References OSCL\_String::setrep\_to\_char(), and CHheapRep::size.

**6.4.4.90 template<class Alloc > void OSCL\_wHeapString< Alloc >::set (const other\_chartype \* *buf*, optype *op*) [inline, inherited]**

References oscl\_strlen(), OSCL\_wString::setrep\_to\_wide\_char(), and CHheapRep::size.

**6.4.4.91 template<class Alloc > void OSCL\_HeapString< Alloc >::set (const other\_chartype \* *buf*, optype *op*) [inline, inherited]**

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

**Parameters**

*buf*,: NULL-terminated wide string.

*op*,: conversion operation to apply

References oscl\_strlen(), OSCL\_String::setrep\_to\_char(), and CHeapRep::size.

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

References OSCL\_wHeapString< Alloc >::get\_cstr(), oscl\_strlen(), OSCL\_wString::set\_rep(), and CHeapRep::size.

**6.4.4.93 template<class Alloc > void OSCL\_HeapString< Alloc >::set (const chartype \* *buf*, uint32 *length*) [inline, 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.

References OSCL\_HeapString< Alloc >::get\_cstr(), oscl\_strlen(), OSCL\_wString::set\_rep(), and CHeapRep::size.

**6.4.4.94 OSCL\_IMPORT\_REF const char\* skip\_to\_line\_term (const char \* *start\_ptr*, const char \* *end\_ptr*)**

**6.4.4.95 OSCL\_IMPORT\_REF const char\* skip\_to\_whitespace (const char \* *start*, const char \* *end*)**

**6.4.4.96 OSCL\_IMPORT\_REF const char\* skip\_whitespace (const char \* *start*, const char \* *end*)**

**6.4.4.97 OSCL\_IMPORT\_REF char\* skip\_whitespace (char \* *ptr*)**

**6.4.4.98 OSCL\_IMPORT\_REF const char\* skip\_whitespace (const char \* *ptr*)**

**6.4.4.99 OSCL\_IMPORT\_REF const char\* skip\_whitespace\_and\_line\_term (const char \* *start*, const char \* *end*)**

**6.4.4.100 template<class Alloc > OSCL\_HeapString< Alloc >::~OSCL\_HeapString () [inline, inherited]**

References CHeapRep::remove\_ref().

**6.4.4.101 template<uint32 MaxBufSize> OSCL\_StackString< MaxBufSize >::~OSCL\_StackString () [inline, inherited]**

**6.4.4.102 template<class Alloc > OSCL\_wHeapString< Alloc >::~OSCL\_wHeapString () [inline, inherited]**

References CHeapRep::remove\_ref().

**6.4.4.103 template<uint32 MaxBufSize> OSCL\_wStackString< MaxBufSize >::~OSCL\_wStackString() [inline, inherited]**

## 6.4.5 Variable Documentation

**6.4.5.1 const int32 APPEND\_MEDIA\_AT\_END = -1**

**6.4.5.2 const uint8 OSCL\_ASCII\_CASE\_MAGIC\_BIT = 0x20**

Referenced by WStrPtrLen::isCIEquivalentTo(), and StrPtrLen::isCIPrefixOf().

## 6.5 OSCL Error

### Data Structures

- class [OsclErrorTrap](#)
- class [OsclError](#)
- class [OsclSingletonRegistryEx](#)
- class [OsclSingletonEx< T, ID, Registry >](#)
- class [OsclTLSRegistryEx](#)
- class [OsclTLSEx< T, ID, Registry >](#)
- class [OsclErrorAllocator](#)

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

- class [internalLeave](#)
- class [OsclJump](#)
- class [OsclTrapStackItem](#)
- class [OsclTrapStack](#)
- class [OsclErrorTrapImp](#)
- class [OsclException< LeaveCode >](#)

*oscl\_exception.h contains all the exception handling macros and classes*

- class [\\_OsclHeapBase](#)
- class [OsclTrapItem](#)
- class [OsclNameString< \\_\\_len >](#)

### Files

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

*Defines functions to access additional information on errors where supported through an errno or similar service.*

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

*OSCL Error trap and cleanup include file.*

- file [oscl\\_error\\_allocator.h](#)

*Defines a memory allocation class used by the oscl error layer.*

- file [oscl\\_error\\_codes.h](#)

*Defines basic error and leave codes.*

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

*Internal error implementation support.*

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

*Implementation File for Leave using C++ exceptions.*

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

*Implementation File for Leave using system fatal error.*

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

*Implementation of using Setjmp / Longjmp.*

- file [oscl\\_error\\_trapcleanup.h](#)  
*OSCL Error trap and cleanup implementation include file.*
- file [oscl\\_exception.h](#)  
*contains all the exception handling macros and classes*
- file [oscl\\_heapbase.h](#)  
*OSCL Heap Base include file.*
- file [oscl\\_namestring.h](#)  
*Name string class include file.*

## 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 **PVError\_DoLeave()** \_OSCL\_Abort()
- #define **\_PV\_TRAP(\_\_r, \_\_s)**
- #define **\_PV\_TRAP\_NO\_TLS(\_\_tr, \_\_r, \_\_s)**
- #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)**
- #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.*

- #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** ()
 

*oscl\_errno.h* contains functions to access the global errno
- OSCL\_IMPORT\_REF int **OSCL\_GetLastError** ()
 

This function returns the value of the system's global error number variable.
- OSCL\_IMPORT\_REF bool **OSCL\_SetLastError** (int newVal)
 

This function sets the last error code for the system.
- OSCL\_IMPORT\_REF char \* **OSCL\_StrError** (int errnum)
 

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

### 6.5.1 Define Documentation

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

**Value:**

```
__r=OsclErrNone; \
{ \
    OsclErrorTrapImp* __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();} \
}
```

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

**Value:**

```
__r=OsclErrNone; \
{__s;}
```

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

**Value:**

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

**6.5.1.4 #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();} \
}
```

**6.5.1.5 #define \_PV\_TRAP\_NO\_TLS(\_tr, \_\_r, \_\_s)****Value:**

```
__r=OsclErrNone; \
{__s;}
```

**6.5.1.6 #define \_PV\_TRAP\_NO\_TLS(\_trapimp, \_\_r, \_\_s)****Value:**

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

**6.5.1.7 #define internalLeave (-1)****6.5.1.8 #define OSCL\_BAD\_ALLOC\_EXCEPTION\_CODE OsclErrNoMemory****6.5.1.9 #define OSCL\_CATCH(\_leave\_status, \_catch\_value, \_statements) else if  
(\_leave\_status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}**

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

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

**Parameters**

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

*exceptiontype* is the exception handled by this catch block

---

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

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

#### Parameters

*\_leave\_status*

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

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

For backward compatibility with old definitions

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

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

#### Parameters

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

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

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

This section defines the macros to be used in the catch block following the try block.

Use this macro to call a function that handles all exception types thrown in the preceding try block

#### Parameters

*\_leave\_status*

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

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

**6.5.1.15 #define OSCL\_LAST\_CATCH(\_leave\_status) else if (\_leave\_-  
status!=OsclErrNone){OSCL\_LEAVE(\_leave\_status);}**

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

#### Parameters

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

**6.5.1.16 #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

Referenced by OsclTimer< Alloc >::OsclTimer().

**6.5.1.17 #define OSCL\_MAX\_TRAP\_LEVELS 20****6.5.1.18 #define OSCL\_TRAPSTACK\_POP() OsclError::Pop()****6.5.1.19 #define OSCL\_TRAPSTACK\_POPDEALLOC() OsclError::PopDealloc()****6.5.1.20 #define OSCL\_TRAPSTACK\_PUSH(a) OsclError::PushL(a)**

Cleanup Stack user macros

**6.5.1.21 #define OSCL\_TRY(\_leave\_status, \_statements) \_PV\_TRAP(\_leave\_status,\_statements)**

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

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

**Parameters**

*oscl\_leave\_status* oscl\_leave\_status will receive the result of any OSCL\_LEAVE (which will get called from a OSCL\_THROW) on systems that do not support exception handling. This is unused on systems that do support exception handling

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

**6.5.1.22 #define OSCL\_TRY\_NO\_TLS(\_\_trapimp, \_leave\_status, \_statements) \_PV\_TRAP\_-  
NO\_TLS(\_\_trapimp,\_leave\_status,\_statements)****6.5.1.23 #define OsclErrAlreadyExists 106****6.5.1.24 #define OsclErrAlreadyInstalled 116****6.5.1.25 #define OsclErrArgument 104**

Referenced by OsclTimer< Alloc >::OsclTimer().

**6.5.1.26 #define OsclErrBadHandle 105**

**6.5.1.27 #define OsclErrBusy 107**

**6.5.1.28 #define OsclErrCancelled 102**

**6.5.1.29 #define OsclErrCorrupt 109**

**6.5.1.30 #define OsclErrGeneral 100**

Referenced by OsclSocketMethod::ConstructL(), and OsclDNSRequestAO::ConstructL().

**6.5.1.31 #define OsclErrInvalidState 113**

**6.5.1.32 #define OsclErrNoHandler 118**

**6.5.1.33 #define OsclErrNoMemory 101**

Referenced by OsclBuf::NewL().

**6.5.1.34 #define OsclErrNone 0**

**6.5.1.35 #define OsclErrNoResources 114**

**6.5.1.36 #define OsclErrNotInstalled 115**

**6.5.1.37 #define OsclErrNotReady 108**

**6.5.1.38 #define OsclErrNotSupported 103**

Referenced by OsclRegistryClientImpl::Connect(), OsclRegistryClientImpl::Register(), and OsclRegistryClientImpl::UnRegister().

**6.5.1.39 #define OsclErrOverflow 111**

**6.5.1.40 #define OsclErrSystemCallFailed 117**

**6.5.1.41 #define OsclErrThreadContextIncorrect 119**

**6.5.1.42 #define OsclErrTimeout 110**

**6.5.1.43 #define OsclErrUnderflow 112**

**6.5.1.44 #define OsclFailure -1**

**6.5.1.45 #define OsclPending 1**

**6.5.1.46 #define OsclSuccess 0**

**6.5.1.47 #define PVError\_DoLeave() OsclJump::StaticJump(internalLeave)**

**6.5.1.48 #define PVError\_DoLeave() \_OSCL\_Abort()**

**6.5.1.49 #define PVError\_DoLeave() internalLeave \_\_ilv;\_\_ilv.a=0;throw(\_\_ilv)**

**6.5.1.50 #define PVERRORTRAP\_IMP\_JUMPS**

Internal leave/trap implementation.

**6.5.1.51 #define PVERRORTRAP\_REGISTRY OsclTLSRegistry**

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

## 6.5.2 Typedef Documentation

**6.5.2.1 typedef int32 OsclLeaveCode**

Leave Codes

**6.5.2.2 typedef int32 OsclReturnCode**

Return Codes

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

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

## 6.5.3 Function Documentation

**6.5.3.1 OSCL\_IMPORT\_REF int OSCL\_GetLastError ()**

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

**Returns**

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

**6.5.3.2 OSCL\_IMPORT\_REF bool OSCL\_IsErrnoSupported ()**

[oscl\\_errno.h](#) contains functions to access the global errno

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

**Returns**

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

**6.5.3.3 OSCL\_IMPORT\_REF bool OSCL\_SetLastError (int newVal)**

This function sets the last error code for the system.

**Parameters**

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

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

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

**Parameters**

*errnum* This value represents the error number to map

**Returns**

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

## 6.6 OSCL IO

### Data Structures

- class [OsclDNSObserver](#)
- class [OsclDNS](#)
- class [OsclFileCacheBuffer](#)
- class [OsclFileCache](#)
- struct [oscl\\_fstat](#)
- struct [oscl\\_stat\\_buf](#)
- class [Oscl\\_FileFind](#)
- class [OsclFileHandle](#)
- class [Oscl\\_File](#)
- class [OsclFileManager](#)
- class [OsclNativeFile](#)
- class [Oscl\\_FileServer](#)
- class [OsclFileStatsItem](#)
- class [OsclFileStats](#)
- class [OsclNativeFileParams](#)
- class [OsclSocketServ](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

### 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 [OsclFileHandle](#).*

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

*The file [oscl\\_file\\_io.h](#) defines the class [Oscl\\_File](#). This is the public API to the basic file I/O operations.*

- file [oscl\\_file\\_manager.h](#)

*File management class.*

- file [oscl\\_file\\_native.h](#)

*The file [oscl\\_file\\_native.h](#) defines the class [OsclNativeFile](#). This is the porting layer for basic file I/O operations.*

- file [oscl\\_file\\_server.h](#)

*The file [oscl\\_file\\_server.h](#) defines the class [Oscl\\_FileServer](#). This is the porting layer for file server implementations.*

- file [oscl\\_file\\_stats.h](#)

*File stats class.*

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

*The file [oscl\\_file\\_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.*

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

*The file [oscl\\_socket.h](#) defines the OSCL Socket APIs.*

## Defines

- #define [TOsclFileOffsetInt32](#) int32
- #define [OSCL\\_FILE\\_STATS\\_LOGGER\\_NODE](#) "OsclFileStats"
- #define [OSCL\\_IO\\_FILENAME\\_MAXLEN](#) 512
- #define [OSCL\\_IO\\_EXTENSION\\_MAXLEN](#) 512
- #define [OSCL\\_FILE\\_WCHAR\\_PATH\\_DELIMITER](#) \_STRLIT("/")
- #define [OSCL\\_FILE\\_CHAR\\_PATH\\_DELIMITER](#) \_STRLIT\_CHAR("/")

## Typedefs

- typedef struct [oscl\\_fsstat](#) OSCL\_FSSTAT
- typedef struct [oscl\\_stat\\_buf](#) OSCL\_STAT\_BUF
- typedef FILE \* [TOsclFileHandle](#)

## Enumerations

- enum [TPVDNSFxn](#) { [EPVDNSGetHostByName](#) }
- enum [TPVDNSEvent](#) {
 [EPVDNSSuccess](#), [EPVDNSPending](#), [EPVDNSTimeout](#), [EPVDNSFailure](#),
 [EPVDNSCancel](#) }
- enum [OSCL\\_FILEMGMT\\_PERMS](#) { [OSCL\\_FILEMGMT\\_PERMS\\_READ](#) = 0x1, [OSCL\\_FILEMGMT\\_PERMS\\_WRITE](#) = 0x2, [OSCL\\_FILEMGMT\\_PERMS\\_EXECUTE](#) = 0x4 }
- enum [OSCL\\_FILEMGMT\\_MODES](#) { [OSCL\\_FILEMGMT\\_MODE\\_DIR](#) = 0x1 }
- enum [OSCL\\_FILEMGMT\\_ERR\\_TYPE](#) {
 [OSCL\\_FILEMGMT\\_E\\_OK](#) = 0, [OSCL\\_FILEMGMT\\_E\\_PATH\\_TOO\\_LONG](#), [OSCL\\_FILEMGMT\\_E\\_PATH\\_NOT\\_FOUND](#), [OSCL\\_FILEMGMT\\_E\\_ALREADY\\_EXISTS](#),
 [OSCL\\_FILEMGMT\\_E\\_NOT\\_EMPTY](#), [OSCL\\_FILEMGMT\\_E\\_PERMISSION\\_DENIED](#), [OSCL\\_FILEMGMT\\_E\\_NO\\_MATCH](#), [OSCL\\_FILEMGMT\\_E\\_UNKNOWN](#),
 [OSCL\\_FILEMGMT\\_E\\_SYS\\_SPECIFIC](#), [OSCL\\_FILEMGMT\\_E\\_NOT\\_IMPLEMENTED](#) }

- enum `TOsclFileOp` {
   
`EOsclFileOp_Open, EOscFileOp_Close, EOscFileOp_Read, EOscFileOp_Write,`
  
`EOscFileOp_Seek, EOscFileOp_Tell, EOscFileOp_Size, EOscFileOp_Flush,`
  
`EOscFileOp_EndOfFile, EOscFileOp_SetSize, EOscFileOp_NativeOpen, EOscFileOp_NativeClose,`
  
`EOscFileOp_NativeRead, EOscFileOp_NativeWrite, EOscFileOp_NativeSeek, EOscFileOp_NativeTell,`
  
`EOscFileOp_NativeSize, EOscFileOp_NativeFlush, EOscFileOp_NativeEndOfFile,`
  
`EOscFileOp_NativeSetSize,`
  
`EOscFileOp_Last }`

## Functions

- virtual `OsclDNSObserver::~OsclDNSObserver()`
- `OSCL_IMPORT_REF OsclDNS::~OsclDNS()`
- `OSCL_IMPORT_REF TPVDNSEvent OsclDNS::GetHostByName (char *name, OsclNetworkAddress &addr, int32 aTimeoutMsec=-1, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aAddressList=NULL)`
- `OSCL_IMPORT_REF void OsclDNS::CancelGetHostByName()`
- `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)`
- static `OSCL_IMPORT_REF bool OsclFileManager::OsclGetFileSize (const oscl_wchar *aFileName, uint64 &aFileSize)`
- static `OSCL_IMPORT_REF bool OsclFileManager::OsclGetFileSize (const char *aFileName, uint64 &aFileSize)`
- static `OSCL_IMPORT_REF bool OsclFileManager::OsclGetFileCreationTime (const oscl_wchar *aFileName, uint64 &aFileCreationTime)`
- static `OSCL_IMPORT_REF bool OsclFileManager::OsclGetFileCreationTime (const char *aFileName, uint64 &aFileCreationTime)`
- static `OSCL_IMPORT_REF bool OsclFileManager::OsclGetFileLastAccessTime (const oscl_wchar *aFileName, uint64 &aFileLastAccessTime)`

- static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileLastAccessTime (const char \*aFileName, uint64 &aFileLastAccessTime)
- static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileLastWriteTime (const oscl\_wchar \*aFileName, uint64 &aFileLastWriteTime)
- static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileLastWriteTime (const char \*aFileName, uint64 &aFileLastWriteTime)
- static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileAttributes (const oscl\_wchar \*aFileName, uint32 &aFileAttributes)
- static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileAttributes (const char \*aFileName, uint32 &aFileAttributes)
- static OSCL\_IMPORT\_REF void OsclFileManager::OsclExtractFilenameFromFullPath (const char \*aPath, char \*&aFileName)
- static OSCL\_IMPORT\_REF void OsclFileManager::OsclExtractFilenameFromFullPath (const oscl\_wchar \*aPath, oscl\_wchar \*&aFileName)
- OSCL\_IMPORT\_REF OsclSocketServ::~OsclSocketServ ()
- OSCL\_IMPORT\_REF int32 OsclSocketServ::Connect (uint32 aMessageSlots=8, bool aShareSession=false)
- OSCL\_IMPORT\_REF void OsclSocketServ::Close (bool aCleanup=true)
- OSCL\_IMPORT\_REF OsclUDPSocket::~OsclUDPSocket ()
- OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::ThreadLogoff ()
- OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::ThreadLogon (OsclSocketServ &aServ, OsclSocketObserver \*aObserver)
- OSCL\_IMPORT\_REF int32 OsclUDPSocket::Close ()
- OSCL\_IMPORT\_REF int32 OsclUDPSocket::Bind (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF int32 OsclUDPSocket::Join (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF int32 OsclUDPSocket::JoinMulticastGroup (OsclIpMReq &aMReq)
- OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetMulticastTTL (int32 aTTL)
- OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetOptionToReuseAddress ()
- OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetTOS (const OsclSocketTOS &aTOS)
- OSCL\_IMPORT\_REF int32 OsclUDPSocket::GetPeerName (OsclNetworkAddress &aPeerName)
- OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void OsclUDPSocket::CancelBind ()
- OSCL\_IMPORT\_REF uint8 \* OsclUDPSocket::GetRecvData (int32 \*aLength)
- OSCL\_IMPORT\_REF uint8 \* OsclUDPSocket::GetSendData (int32 \*aLength)
- OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::SendTo (const uint8 \*aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void OsclUDPSocket::CancelSendTo ()
- 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)
- OSCL\_IMPORT\_REF void OsclUDPSocket::CancelRecvFrom ()
- OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetRecvBufferSize (uint32 size)
- OSCL\_IMPORT\_REF OsclTCPSocket::~OsclTCPSocket ()
- OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::ThreadLogoff ()
- OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::ThreadLogon (OsclSocketServ &aServ, OsclSocketObserver \*aObserver)
- OSCL\_IMPORT\_REF int32 OsclTCPSocket::Close ()
- OSCL\_IMPORT\_REF int32 OsclTCPSocket::Bind (OsclNetworkAddress &aAddress)

- OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void OsclTCPSocket::CancelBind ()
- OSCL\_IMPORT\_REF int32 OsclTCPSocket::SetOptionToReuseAddress ()
- OSCL\_IMPORT\_REF int32 OsclTCPSocket::SetTOS (const OsclSocketTOS &aTOS)
- OSCL\_IMPORT\_REF int32 OsclTCPSocket::GetPeerName (OsclNetworkAddress &aPeerName)
- OSCL\_IMPORT\_REF int32 OsclTCPSocket::Listen (int32 aQueueSize)
- OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::ListenAsync (int32 aQueueSize, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void OsclTCPSocket::CancelListen ()
- OSCL\_IMPORT\_REF OsclTCPSocket \* OsclTCPSocket::GetAcceptedSocketL (uint32 aId)
- OSCL\_IMPORT\_REF uint8 \* OsclTCPSocket::GetRecvData (int32 \*aLength)
- OSCL\_IMPORT\_REF uint8 \* OsclTCPSocket::GetSendData (int32 \*aLength)
- OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Connect (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void OsclTCPSocket::CancelConnect ()
- OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void OsclTCPSocket::CancelShutdown ()
- OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Accept (int32 aTimeout=-1)
- OSCL\_IMPORT\_REF void OsclTCPSocket::CancelAccept ()
- OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Send (const uint8 \*aPtr, uint32 aLen, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void OsclTCPSocket::CancelSend ()
- OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Recv (uint8 \*aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void OsclTCPSocket::CancelRecv ()

## Friends

- class OsclDNS::OsclDNSRequestAO
- class OsclSocketServ::OsclTCPSocket
- class OsclSocketServ::OsclUDPSocket
- class OsclSocketServ::OsclDNS

### 6.6.1 Define Documentation

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

6.6.1.2 `#define OSCL_FILE_STATS_LOGGER_NODE "OsclFileStats"`

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

6.6.1.4 `#define OSCL_IO_EXTENSION_MAXLEN 512`

6.6.1.5 `#define OSCL_IO_FILENAME_MAXLEN 512`

6.6.1.6 `#define TOsclFileOffsetInt32 int32`

### 6.6.2 Typedef Documentation

6.6.2.1 `typedef struct oscl_fsstat OSCL_FSSTAT`

6.6.2.2 `typedef struct oscl_stat_buf OSCL_STAT_BUF`

6.6.2.3 `typedef FILE* TOsclFileHandle`

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

### 6.6.3 Enumeration Type Documentation

6.6.3.1 `enum OSCL_FILEMGMT_ERR_TYPE`

Enumerator:

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

6.6.3.2 `enum OSCL_FILEMGMT_MODES`

Enumerator:

*OSCL\_FILEMGMT\_MODE\_DIR*

### 6.6.3.3 enum OSCL\_FILEMGMT\_PERMS

Enumerator:

*OSCL\_FILEMGMT\_PERMS\_READ*  
*OSCL\_FILEMGMT\_PERMS\_WRITE*  
*OSCL\_FILEMGMT\_PERMS\_EXECUTE*

### 6.6.3.4 enum TOsclFileOp

Enumerator:

*EOsclFileOp\_Open*  
*EOsclFileOp\_Close*  
*EOsclFileOp\_Read*  
*EOsclFileOp\_Write*  
*EOsclFileOp\_Seek*  
*EOsclFileOp\_Tell*  
*EOsclFileOp\_Size*  
*EOsclFileOp\_Flush*  
*EOsclFileOp\_EndOfFile*  
*EOsclFileOp\_SetSize*  
*EOsclFileOp\_NativeOpen*  
*EOsclFileOp\_NativeClose*  
*EOsclFileOp\_NativeRead*  
*EOsclFileOp\_NativeWrite*  
*EOsclFileOp\_NativeSeek*  
*EOsclFileOp\_NativeTell*  
*EOsclFileOp\_NativeSize*  
*EOsclFileOp\_NativeFlush*  
*EOsclFileOp\_NativeEndOfFile*  
*EOsclFileOp\_NativeSetSize*  
*EOsclFileOp\_Last*

### 6.6.3.5 enum TPVDNSEvent

Enumerator:

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

### 6.6.3.6 enum TPVDNSFxn

Enumerator:

*EPVNDNSGetHostName*

## 6.6.4 Function Documentation

### 6.6.4.1 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Accept (int32 *aTimeout* = -1) [inherited]

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.

### 6.6.4.2 OSCL\_IMPORT\_REF int32 OsclTCPSocket::Bind (OsclNetworkAddress & *aAddress*) [inherited]

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.

### 6.6.4.3 OSCL\_IMPORT\_REF int32 OsclUDPSocket::Bind (OsclNetworkAddress & *aAddress*) [inherited]

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.

**6.6.4.4 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::BindAsync  
(OsclNetworkAddress & aAddress, int32 aTimeoutMsec = (-1)) [inherited]**

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.

**6.6.4.5 OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::BindAsync  
(OsclNetworkAddress & aAddress, int32 aTimeoutMsec = (-1)) [inherited]**

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.

**6.6.4.6 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelAccept () [inherited]**

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.

**6.6.4.7 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelBind () [inherited]**

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.

**6.6.4.8 OSCL\_IMPORT\_REF void OsclUDPSocket::CancelBind () [inherited]**

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.

#### **6.6.4.9 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelConnect () [inherited]**

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.

#### **6.6.4.10 OSCL\_IMPORT\_REF void OsclDNS::CancelGetHostName () [inherited]**

Cancel GetHostName

This method will cancel any pending GetHostName operation on the current object, causing the GetHostName to complete with error EPVDNSCancel. If there is no pending GetHostName operation, this method will have no effect.

#### **6.6.4.11 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelListen () [inherited]**

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.

#### **6.6.4.12 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelRecv () [inherited]**

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.

#### **6.6.4.13 OSCL\_IMPORT\_REF void OsclUDPSocket::CancelRecvFrom () [inherited]**

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.

#### **6.6.4.14 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelSend () [inherited]**

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.

**6.6.4.15 OSCL\_IMPORT\_REF void OsclUDPSocket::CancelSendTo () [inherited]**

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.

**6.6.4.16 OSCL\_IMPORT\_REF void OsclTCPSocket::CancelShutdown () [inherited]**

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.

**6.6.4.17 OSCL\_IMPORT\_REF int32 OsclTCPSocket::Close () [inherited]**

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.

**6.6.4.18 OSCL\_IMPORT\_REF int32 OsclUDPSocket::Close () [inherited]**

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.

**6.6.4.19 OSCL\_IMPORT\_REF void OsclSocketServ::Close (bool *aCleanup* = true) [inherited]**

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.

**6.6.4.20 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Connect (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1) [inherited]**

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.

#### 6.6.4.21 OSCL\_IMPORT\_REF int32 OsclSocketServ::Connect (uint32 *aMessageSlots* = 8, bool *aShareSession* = **false**) [inherited]

Connect to socket server. This is a synchronous method.

**Parameters**

*Number* of message slots.

**Returns**

Returns OsclErrNone for success, or a platform-specific code.

#### 6.6.4.22 OSCL\_IMPORT\_REF OsclTCPSocket\* OsclTCPSocket::GetAcceptedSocketL (uint32 *aId*) [inherited]

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.

#### 6.6.4.23 OSCL\_IMPORT\_REF TPVDNSEvent OsclDNS::GetHostByName (char \* *name*, OsclNetworkAddress & *addr*, int32 *aTimeoutMsec* = -1, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \* *aAddressList* = NULL) [inherited]

GetHostByName. This is an asynchronous method.

**Parameters**

*name*,: Null-terminated string containing the host name.

*addr*,: The output address corresponding to the host. The ipAddr field will contain the network address of the host in dotted decimal notation.

*aTimeoutMsec*,: A timeout for the request in milliseconds, or (-1) to indicate infinite wait.

*aAddressList* : A list of addresses for the host.

#### Returns

: EPVDNSPending for success, EPVDNSFailure for failure.

### 6.6.4.24 OSCL\_IMPORT\_REF int32 OsclTCPSocket::GetPeerName (OsclNetworkAddress & *aPeerName*) [inherited]

Retrieves the peer address of the socket

#### Parameters

*aPeerName*,: This will store the peer address when API returns successfully.

#### Returns

Returns OsclErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

### 6.6.4.25 OSCL\_IMPORT\_REF int32 OsclUDPSocket::GetPeerName (OsclNetworkAddress & *aPeerName*) [inherited]

Retrieves the peer address of the socket

#### Parameters

*aPeerName*,: This will store the peer address when API returns successfully.

#### Returns

Returns OsclErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

### 6.6.4.26 OSCL\_IMPORT\_REF uint8\* OsclTCPSocket::GetRecvData (int32 \* *aLength*) [inherited]

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.

**6.6.4.27 OSCL\_IMPORT\_REF uint8\* OsclUDPSocket::GetRecvData (int32 \* *aLength*)  
[inherited]**

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.

**6.6.4.28 OSCL\_IMPORT\_REF uint8\* OsclTCPSocket::GetSendData (int32 \* *aLength*)  
[inherited]**

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.

**6.6.4.29 OSCL\_IMPORT\_REF uint8\* OsclUDPSocket::GetSendData (int32 \* *aLength*)  
[inherited]**

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.

**6.6.4.30 OSCL\_IMPORT\_REF int32 OsclUDPSocket::Join (OsclNetworkAddress & *aAddress*)  
[inherited]**

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

**6.6.4.31 OSCL\_IMPORT\_REF int32 OsclUDPSocket::JoinMulticastGroup (OsclIpMReq & aMReq) [inherited]**

Join the multicast group.

**Parameters**

*aMReq*,: Multicast group information.

**Returns**

Returns: OsclErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK\_ERR\_BAD\_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

**6.6.4.32 OSCL\_IMPORT\_REF int32 OsclTCPSocket::Listen (int32 aQueueSize) [inherited]**

Listen. This is a synchronous method.

**Parameters**

*aQueueSize*,: Queue size.

**Returns**

Returns OsclErrNone for success, or a platform-specific error code.

**6.6.4.33 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::ListenAsync (int32 aQueueSize, int32 aTimeoutMsec = (-1)) [inherited]**

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.

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

oscl\_chdir changes the current directory to the path given

**Parameters**

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

**Returns**

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

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

oscl\_chdir changes the current directory to the path given

**Parameters**

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

**Returns**

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

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

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

**Parameters**

*pointer* to character buffer to receive the current directory

*size* size of buffer in characters

**Returns**

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

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

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

**Parameters**

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

*size* size of buffer in wide characters

**Returns**

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

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

oscl\_mkdir function creates a directory in the path given

**Parameters**

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

**Returns**

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

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

oscl\_mkdir function creates a directory in the path given

**Parameters**

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

**Returns**

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

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

oscl\_rename removes an empty directory in the path given

**Parameters**

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

**Returns**

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

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

oscl\_rename function renames a file or directory

**Parameters**

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

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

**Returns**

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

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

oscl\_rmdir removes an empty directory in the path given

**Parameters**

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

**Returns**

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

---

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

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

**Parameters**

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

**Returns**

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

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

oscl\_stat function can be used to determine the attributes of a file in addition to whether the file exists or not

**Parameters**

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

**Returns**

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

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

oscl\_stat function can be used to determine the attributes of a file in addition to whether the file exists or not

**Parameters**

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

**Returns**

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

**6.6.4.46 OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_stats (OSCL\_FSSTAT \* stats, const oscl\_wchar \* path)**

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

**Parameters**

*stats* pointer to structure to hold information

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

**Returns**

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

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

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

**Parameters**

*stats* pointer to structure to hold information

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

**Returns**

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

**6.6.4.48 static OSCL\_IMPORT\_REF void OsclFileManager::OsclExtractFilenameFromFullPath (const oscl\_wchar \* *aPath*, oscl\_wchar \*& *aFileName*) [static, inherited]**

**6.6.4.49 static OSCL\_IMPORT\_REF void OsclFileManager::OsclExtractFilenameFromFullPath (const char \* *aPath*, char \*& *aFileName*) [static, inherited]**

OsclExtractFilenameFromFullPath utility function provide the FileName From Path of a file.

**Parameters**

← *character* path; the full path of the file or directory

→ *character* File Name .It is assigned a pointer to file name in path itself.

**Returns**

void for all condition

**6.6.4.50 static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileAttributes (const char \* *aFileName*, uint32 & *aFileAttributes*) [static, inherited]**

OsclGetFileAttributes utility function provides the various attributes of file (or directory) like if it is hidden, read only etc. The uint32 value is to be interpreted as per the enum OSCL\_FILE\_ATTRIBUTE\_TYPE defined in [oscl\\_file\\_manager.h](#)

**Parameters**

← *character* path; the full path of the file or directory

→ *file* attributes.

**Returns**

true if successful, otherwise false.

---

**6.6.4.51 static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileAttributes (const oscl\_wchar \* aFileName, uint32 & aFileAttributes) [static, inherited]**

OsclGetFileAttributes utility function provides the various attributes of file (or directory) like if it is hidden, read only etc. The uint32 value is to be interpreted as per the enum OSCL\_FILE\_ATTRIBUTE\_TYPE defined in [oscl\\_file\\_manager.h](#)

#### Parameters

- ← *wide* character path; the full path of the file or directory
- *file* attributes.

#### Returns

true if successful, otherwise false.

---

**6.6.4.52 static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileCreationTime (const char \* aFileName, uint64 & aFileCreationTime) [static, inherited]**

OsclGetFileCreationTime utility function provides the file (or directory) creation time

#### Note

On symbian platform, this api returns last modified time.

#### Parameters

- ← *character* path; the full path of the file or directory
- *creation* time in microseconds.

#### Returns

true if successful, otherwise false.

---

**6.6.4.53 static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileCreationTime (const oscl\_wchar \* aFileName, uint64 & aFileCreationTime) [static, inherited]**

OsclGetFileCreationTime utility function provides the file (or directory) creation time

#### Note

On symbian platform, this api returns last modified time.

#### Parameters

- ← *wide* character path; the full path of the file or directory
- *creation* time in microseconds

#### Returns

true if successful, otherwise false.

**6.6.4.54 static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileLastAccessTime (const char \**aFileName*, uint64 & *aFileLastAccessTime*) [static, inherited]**

OsclGetFileLastAccessTime utility function provides the file (or directory) last access time, which might be different from last modified time.

**Note**

On symbian platform, this api returns last modified time.

**Parameters**

- ← *character* path; the full path of the file or directory
- *Last* access time in microseconds.

**Returns**

true if successful, otherwise false.

**6.6.4.55 static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileLastAccessTime (const oscl\_wchar \**aFileName*, uint64 & *aFileLastAccessTime*) [static, inherited]**

OsclGetFileLastAccessTime utility function provides the file (or directory) last access time, which might be different from last modified time.

**Note**

On symbian platform, this api returns last modified time.

**Parameters**

- ← *wide* character path; the full path of the file or directory
- *Last* access time in microseconds

**Returns**

true if successful, otherwise false.

**6.6.4.56 static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileLastWriteTime (const char \**aFileName*, uint64 & *aFileLastWriteTime*) [static, inherited]**

OsclGetFileLastWriteTime utility function provides the file (or directory) last modified time.

**Parameters**

- ← *character* path; the full path of the file or directory
- *last* modified time in microseconds

**Returns**

true if successful, otherwise false.

**6.6.4.57 static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileLastWriteTime (const oscl\_wchar \* *aFileName*, uint64 & *aFileLastWriteTime*) [static, inherited]**

OsclGetFileLastWriteTime utility function provides the file (or directory) last modified time.

**Parameters**

- ← *wide* character path; the full path of the file or directory
- *last* modified time in microseconds

**Returns**

true if successful, otherwise false.

**6.6.4.58 static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileSize (const char \* *aFileName*, uint64 & *aFileSize*) [static, inherited]**

OsclGetFileSize utility function provides the file size. For directory, this value is undefined.

**Parameters**

- ← *character* path; the full path of the file or directory
- *file* size in bytes.

**Returns**

true if successful, otherwise false.

**6.6.4.59 static OSCL\_IMPORT\_REF bool OsclFileManager::OsclGetFileSize (const oscl\_wchar \* *aFileName*, uint64 & *aFileSize*) [static, inherited]**

OsclGetFileSize utility function provides the file size. For directory, this value is undefined. creation time

**Parameters**

- ← *wide* character path; the full path of the file or directory
- *file* size in bytes

**Returns**

true if successful, otherwise false.

**6.6.4.60 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Recv (uint8 \* *aPtr*, uint32 *aMaxLen*, int32 *aTimeoutMsec* = -1) [inherited]**

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.

**6.6.4.61 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)  
 [inherited]**

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.

**6.6.4.62 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Send (const uint8 \* *aPtr*,  
 uint32 *aLen*, int32 *aTimeoutMsec* = -1) [inherited]**

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.

---

**6.6.4.63 OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::SendTo (const uint8 \*  
aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1)  
[inherited]**

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.

**6.6.4.64 OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetMulticastTTL (int32 aTTL)  
[inherited]**

Controls the number of intermediate systems through which a multicast datagram can be forwarded.

#### Parameters

*aTTL*:Specifies the time-to-live value for multicast datagrams sent through this socket.

#### Returns

Returns: OsclErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK\_ERR\_BAD\_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

**6.6.4.65 OSCL\_IMPORT\_REF int32 OsclTCPSocket::SetOptionToReuseAddress ()  
[inherited]**

Allows the server to bind to an address which is in a TIME\_WAIT state.

#### Returns

Returns: OsclErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK\_ERR\_BAD\_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

**6.6.4.66 OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetOptionToReuseAddress ()  
[inherited]**

Allows the server to bind to an address which is in a TIME\_WAIT state.

**Returns**

Returns: OsclErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK\_ERR\_BAD\_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

**6.6.4.67 OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetRecvBufferSize (uint32 *size*)  
[inherited]**

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.

**6.6.4.68 OSCL\_IMPORT\_REF int32 OsclTCPSocket::SetTOS (const OsclSocketTOS & *aTOS*)  
[inherited]**

Sets the Type of Service field of each outgoing IP datagram.

**Parameters**

*aTOS*,: Specifies the type of service requested.

**Returns**

Returns: OsclErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK\_ERR\_BAD\_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

**6.6.4.69 OSCL\_IMPORT\_REF int32 OsclUDPSocket::SetTOS (const OsclSocketTOS & *aTOS*)  
[inherited]**

Sets the Type of Service field of each outgoing IP datagram.

**Parameters**

*aTOS*,: Specifies the type of service requested.

**Returns**

Returns: OsclErrNone for success, or a platform-specific error code. or PVSOCK\_ERR\_NOT\_SUPPORTED, if underlying OS doesn't support joining multicast group PVSOCK\_ERR\_BAD\_PARAM, if config io file is not configured in accordance with underlying OS PVSOCK\_ERR\_NOT\_IMPLEMENTED, if this API is not implemented in OSCL for the underlying OS

**6.6.4.70 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::Shutdown  
[TPVSocketShutdown *aHow*, int32 *aTimeoutMsec* = -1] [inherited]**

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.

**6.6.4.71 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::ThreadLogoff ()  
[inherited]**

Thread logoff routine. This will prepare for transfer and use of the socket in another thread. All socket requests must be complete prior to calling this routine. If any requests are still active, it will return EPVSocketFailure;

**6.6.4.72 OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::ThreadLogoff ()  
[inherited]**

Thread logoff routine. This will prepare for transfer and use of the socket in another thread. All socket requests must be complete prior to calling this routine. If any requests are still active, it will return EPVSocketFailure;

**6.6.4.73 OSCL\_IMPORT\_REF TPVSocketEvent OsclTCPSocket::ThreadLogon (OsclSocketServ  
& *aServ*, OsclSocketObserver \* *aObserver*) [inherited]**

Thread logon routine. This will complete the transfer of a socket from another thread for use in the current thread. The ThreadLogoff API must be called in the original thread prior to calling ThreadLogon.

**6.6.4.74 OSCL\_IMPORT\_REF TPVSocketEvent OsclUDPSocket::ThreadLogon (OsclSocketServ  
& *aServ*, OsclSocketObserver \* *aObserver*) [inherited]**

Thread logon routine. This will complete the transfer of a socket from another thread for use in the current thread. The ThreadLogoff API must be called in the original thread prior to calling ThreadLogon.

**6.6.4.75 OSCL\_IMPORT\_REF OsclDNS::~OsclDNS () [inherited]**

Destructor.

Note: the application must de-allocate the DNS object using the same allocator that was passed in the NewL object creation call.

**6.6.4.76 virtual OsclDNSObserver::~OsclDNSObserver () [inline, virtual, inherited]**

**6.6.4.77 OSCL\_IMPORT\_REF OsclSocketServ::~OsclSocketServ () [inherited]**

Destructor. The server object must be deleted using the same allocator used in the NewL call.

**6.6.4.78 OSCL\_IMPORT\_REF OsclTCPSocket::~OsclTCPSocket () [inherited]**

Destructor. The object must be deleted using the same allocator used in the NewL call.

**6.6.4.79 OSCL\_IMPORT\_REF OsclUDPSocket::~OsclUDPSocket () [inherited]**

Destructor. The object must be deleted using the same allocator used in the NewL call.

## 6.6.5 Friends

**6.6.5.1 friend class OsclDNS [friend, inherited]**

**6.6.5.2 friend class OsclDNSRequestAO [friend, inherited]**

**6.6.5.3 friend class OsclTCPSocket [friend, inherited]**

**6.6.5.4 friend class OsclUDPSocket [friend, inherited]**

## 6.7 OSCL Proc

### Data Structures

- class [OsclAOStatus](#)
- class [OsclDoubleLink](#)
- class [OsclPriorityLink](#)
- class [OsclDoubleListBase](#)
- class [OsclDoubleList< T >](#)
- class [OsclPriorityList< T >](#)
- class [OsclDoubleRunner< T >](#)
- class [OsclScheduler](#)
- class [OsclSchedulerObserver](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclExecScheduler](#)
- class [PVSchedulerStopper](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [PVActiveBase](#)
- class [OsclReadyAlloc](#)
- class [OsclReadyCompare](#)
- class [OsclTimerCompare](#)
- class [OsclReadyQ](#)
- class [OsclTimerQ](#)
- class [TReadyQueLink](#)
- class [PVThreadContext](#)
- class [OsclExecSchedulerBase](#)

### 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](#)

*Tunable settings for Oscl Scheduler.*

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

*Scheduler common types include file.*

## 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_LOOP_STATS 0`
- `#define PV_SCHED_ENABLE_PERF_LOGGING 1`
- `#define PV_SCHED_ENABLE_THREAD_CONTEXT_CHECKS 1`
- `#define PV_SCHED_LOG_Q 0`
- `#define PV_SCHED_CHECK_Q 0`
- `#define PV_SCHED_FAIR_SCHEDULING 1`
- `#define OSCL_PERF_SUMMARY_LOGGING 0`

## Typedefs

- `typedef PVActiveBase * TOsclReady`

## Enumerations

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

## Functions

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

## Variables

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

## 6.7.1 Define Documentation

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

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

This file defines the **PVActiveBase** class, which is a common base for All PV ExecObjs on all platforms.

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

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

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

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

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

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

**6.7.1.9 #define PVEEXECNAMELEN 30**

**6.7.1.10 #define PVSCHEDNAMELEN 30**

PV Scheduler class

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

**Value:**

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

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

**Value:**

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

## 6.7.2 Typedef Documentation

### 6.7.2.1 `typedef PVActiveBase* TOsclReady`

## 6.7.3 Enumeration Type Documentation

### 6.7.3.1 `enum TPVThreadContext`

Thread context type

Enumerator:

*EPVThreadContext\_InThread*  
*EPVThreadContext\_OsclThread*  
*EPVThreadContext\_NonOsclThread*  
*EPVThreadContext\_Undetermined*

## 6.7.4 Function Documentation

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

Referenced by `OsclDoubleRunner< T >::Set()`.

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

Referenced by `OsclDoubleRunner< T >::operator T *()`, and `OsclDoubleRunner< T >::operator++()`.

## 6.7.5 Variable Documentation

### 6.7.5.1 `const int32 OSCL_REQUEST_ERR_CANCEL = (-1)`

### 6.7.5.2 `const int32 OSCL_REQUEST_ERR_GENERAL = (-2)`

### 6.7.5.3 `const int32 OSCL_REQUEST_ERR_NONE = 0`

Referenced by `CallbackTimer< Alloc >::Run()`.

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

## 6.8 OSCL Init

### Data Structures

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

### Files

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

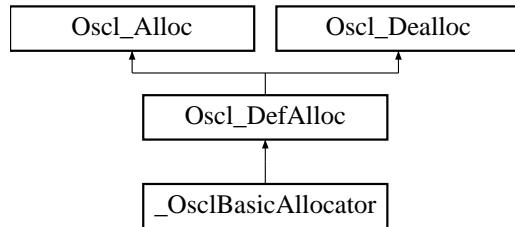
## Chapter 7

# Data Structure Documentation

### 7.1 \_OsclBasicAllocator Class Reference

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

Inheritance diagram for \_OsclBasicAllocator:



#### Public Member Functions

- `OsclAny * allocate (const uint32 size)`
- `void deallocate (OsclAny *p)`
- `virtual ~_OsclBasicAllocator ()`

#### 7.1.1 Detailed Description

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

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

## 7.1.2 Constructor & Destructor Documentation

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

## 7.1.3 Member Function Documentation

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

Implements [Oscl\\_DefAlloc](#).

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

Implements [Oscl\\_DefAlloc](#).

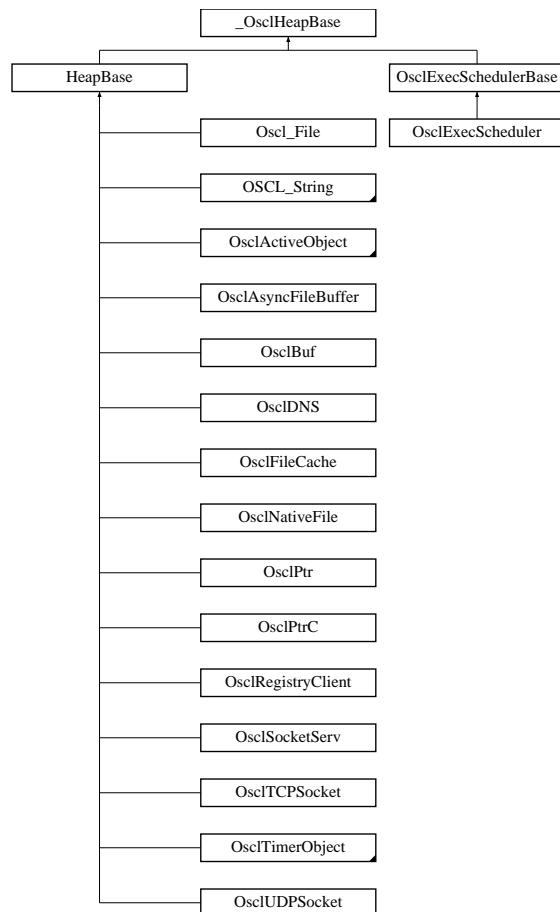
The documentation for this class was generated from the following file:

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

## 7.2 \_OsclHeapBase Class Reference

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

Inheritance diagram for \_OsclHeapBase:



### Public Member Functions

- virtual ~\_OsclHeapBase ()

### Protected Member Functions

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

### Friends

- class PVCleanupStack

### 7.2.1 Detailed Description

[\\_OsclHeapBase](#) is used as the base for cleanup stack items with virtual destructor.

### 7.2.2 Constructor & Destructor Documentation

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

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

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

### 7.2.3 Friends And Related Function Documentation

7.2.3.1 `friend class PVCleanupStack [friend]`

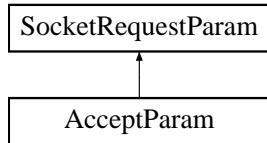
The documentation for this class was generated from the following file:

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

## 7.3 AcceptParam Class Reference

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

Inheritance diagram for AcceptParam:



### Public Member Functions

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

### Data Fields

- [OsclSocketI \\* iBlankSocket](#)

#### 7.3.1 Constructor & Destructor Documentation

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

#### 7.3.2 Field Documentation

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

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

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

## 7.4 allocator Class Reference

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

### 7.4.1 Detailed Description

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

A memory allocator class which allocates and deallocates from a fixed size memory pool; The memory pool is one block of memory and allocations are not fixed in size. The memory pool also has the capability of growing by allocating more block one at a time. This memory pool also provides the capability of returning the tail end of memory previously allocated from the memory pool

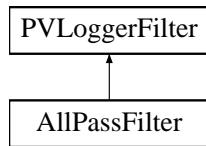
The documentation for this class was generated from the following file:

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

## 7.5 AllPassFilter Class Reference

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

Inheritance diagram for AllPassFilter:



### Public Types

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

### Public Member Functions

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

#### 7.5.1 Detailed Description

Example filter that allows all messages to be logged.

#### 7.5.2 Member Typedef Documentation

##### 7.5.2.1 `typedef PVLoggerFilter::filter_status_type AllPassFilter::filter_status_type`

Reimplemented from [PVLoggerFilter](#).

##### 7.5.2.2 `typedef PVLoggerFilter::log_level_type AllPassFilter::log_level_type`

Reimplemented from [PVLoggerFilter](#).

##### 7.5.2.3 `typedef PVLoggerFilter::message_id_type AllPassFilter::message_id_type`

Reimplemented from [PVLoggerFilter](#).

### 7.5.3 Constructor & Destructor Documentation

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

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

### 7.5.4 Member Function Documentation

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

Implements [PVLoggerFilter](#).

References OSCL\_UNUSED\_ARG, and PVLOGGER\_FILTER\_ACCEPT.

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

Implements [PVLoggerFilter](#).

References OSCL\_UNUSED\_ARG, and PVLOGGER\_FILTER\_ACCEPT.

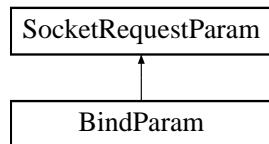
The documentation for this class was generated from the following file:

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

## 7.6 BindParam Class Reference

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

Inheritance diagram for BindParam:



### Public Member Functions

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

### Data Fields

- [OsclNetworkAddress iAddr](#)

#### 7.6.1 Constructor & Destructor Documentation

**7.6.1.1 BindParam::BindParam (OsclNetworkAddress & *anAddr*) [inline]**

#### 7.6.2 Field Documentation

**7.6.2.1 OsclNetworkAddress BindParam::iAddr**

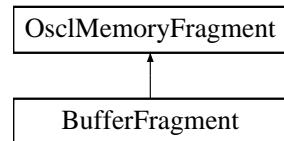
The documentation for this class was generated from the following file:

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

## 7.7 BufferFragment Class Reference

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

Inheritance diagram for BufferFragment:



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

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

## 7.8 BufferMgr Class Reference

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

### Public Member Functions

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

#### 7.8.1 Constructor & Destructor Documentation

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

#### 7.8.2 Member Function Documentation

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

Referenced by [BufferState::decrement\\_refcnt\(\)](#).

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

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

## 7.9 BufferState Class Reference

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

### Public Member Functions

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

#### 7.9.1 Constructor & Destructor Documentation

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

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

#### 7.9.2 Member Function Documentation

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

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

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

References `BufferMgr::BufferReleased()`.

Referenced by `MediaData<ChainClass, max_frags, local_bufsize>::Clear()`, and `BufFragGroup<ChainClass, max_frags>::Clear()`.

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

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

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

**7.9.2.7 `int32 BufferState::get_refcount () [inline]`**

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

Referenced by `BufFragGroup<ChainClass, max_frags>::AddFragment()`.

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

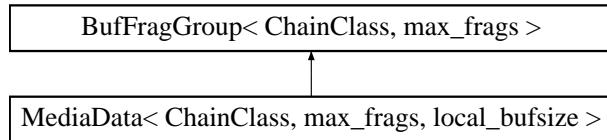
The documentation for this class was generated from the following file:

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

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

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

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



### Public Member Functions

- [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 Member Functions

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

### Protected Attributes

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

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

#### 7.10.1 Constructor & Destructor Documentation

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

References BufFragGroup< ChainClass, max\_frags >::buffer\_states, BufFragGroup< ChainClass, max\_frags >::fragments, and oscl\_memset().

---

**7.10.1.2 template<class ChainClass , uint32 max\_frags> virtual BufFragGroup< ChainClass, max\_frags >::~BufFragGroup () [inline, virtual]**

## 7.10.2 Member Function Documentation

**7.10.2.1 template<class ChainClass , uint32 max\_frags> BufFragStatusClass::status\_t BufFragGroup< ChainClass, max\_frags >::AddFragment (const BufferFragment & *frag*, BufferState \* *in\_buffer\_state*, int32 *location\_offset* = **max\_frags**) [inline, protected]**

References BufFragStatusClass::BFG\_SUCCESS, BufFragGroup< ChainClass, max\_frags >::buffer\_states, BufFragStatusClass::EMPTY\_FRAGMENT, BufFragGroup< ChainClass, max\_frags >::fragments, BufferState::increment\_refcnt(), OsclMemoryFragment::len, BufFragGroup< ChainClass, max\_frags >::length, NULL, BufFragGroup< ChainClass, max\_frags >::num\_fragments, oscl\_memmove(), OsclMemoryFragment::ptr, and BufFragStatusClass::TOO\_MANY\_FRAGS.

**7.10.2.2 template<class ChainClass , uint32 max\_frags> void BufFragGroup< ChainClass, max\_frags >::AppendNext (ChainClass \* *next\_ptr*) [inline]**

References BufFragGroup< ChainClass, max\_frags >::next.

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

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

References BufFragGroup< ChainClass, max\_frags >::buffer\_states, BufferState::decrement\_refcnt(), BufFragGroup< ChainClass, max\_frags >::fragments, BufFragGroup< ChainClass, max\_frags >::length, BufFragGroup< ChainClass, max\_frags >::num\_fragments, and oscl\_memset().

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

References BufFragGroup< ChainClass, max\_frags >::length.

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

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

References BufFragGroup< ChainClass, max\_frags >::next.

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

References BufFragGroup< ChainClass, max\_frags >::num\_fragments.

### 7.10.3 Field Documentation

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

Referenced by BufFragGroup< ChainClass, max\_frags >::AddFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), BufFragGroup< ChainClass, max\_frags >::BufFragGroup(), MediaData< ChainClass, max\_frags, local\_bufsize >::Clear(), BufFragGroup< ChainClass, max\_frags >::Clear(), and BufFragGroup< ChainClass, max\_frags >::GetBufferState().

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

Referenced by BufFragGroup< ChainClass, max\_frags >::AddFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), BufFragGroup< ChainClass, max\_frags >::BufFragGroup(), MediaData< ChainClass, max\_frags, local\_bufsize >::Clear(), BufFragGroup< ChainClass, max\_frags >::Clear(), BufFragGroup< ChainClass, max\_frags >::GetFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::GetMediaFragment(), and MediaData< ChainClass, max\_frags, local\_bufsize >::GetMediaSize().

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

Referenced by BufFragGroup< ChainClass, max\_frags >::AddFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::Clear(), BufFragGroup< ChainClass, max\_frags >::Clear(), BufFragGroup< ChainClass, max\_frags >::GetLength(), and MediaData< ChainClass, max\_frags, local\_bufsize >::GetMediaSize().

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

Referenced by BufFragGroup< ChainClass, max\_frags >::AppendNext(), and BufFragGroup< ChainClass, max\_frags >::GetNext().

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

Referenced by BufFragGroup< ChainClass, max\_frags >::AddFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::Clear(), BufFragGroup< ChainClass, max\_frags >::Clear(), BufFragGroup< ChainClass, max\_frags >::GetBufferState(), BufFragGroup< ChainClass, max\_frags >::GetFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::GetMediaFragment(), BufFragGroup< ChainClass, max\_frags >::GetNumFrags(), and MediaData< ChainClass, max\_frags, local\_bufsize >::GetNumMediaFrags().

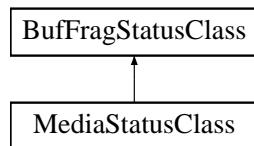
The documentation for this class was generated from the following file:

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

## 7.11 BufFragStatusClass Class Reference

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

Inheritance diagram for BufFragStatusClass:



### Public Types

- enum `status_t` {  
    *BFG\_SUCCESS* = 0, *TOO\_MANY\_FRAGS* = 1, *NOT\_ENOUGH\_SPACE* = 2, *EMPTY\_FRAGMENT* = 3,  
    *NULL\_INPUT* = 4, *FIXED\_FRAG\_LOC\_FULL* = 5, *INTERNAL\_ERROR*, *INVALID\_ID* }

#### 7.11.1 Member Enumeration Documentation

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

Enumerator:

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

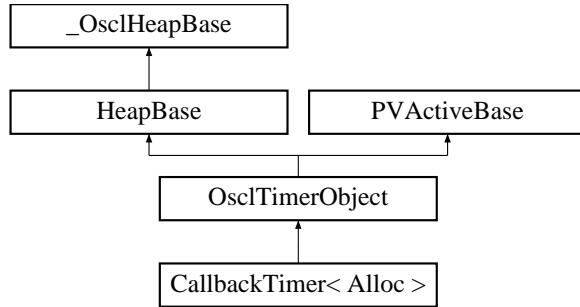
The documentation for this class was generated from the following file:

- `oscl_media_status.h`

## 7.12 CallbackTimer< Alloc > Class Template Reference

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

Inheritance diagram for CallbackTimer< Alloc >:



### Public Member Functions

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

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

#### 7.12.1 Constructor & Destructor Documentation

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

References OsclTimerObject::AddToScheduler().

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

References OsclTimerObject::RemoveFromScheduler().

#### 7.12.2 Member Function Documentation

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

Handles an active object's request completion event.

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

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

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

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

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

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

Implements [PVActiveBase](#).

References [OSCL\\_REQUEST\\_ERR\\_NONE](#), [OsclTimerObject::Status\(\)](#), and [CallbackTimerObserver::TimerBaseElapsed\(\)](#).

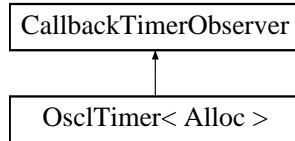
The documentation for this class was generated from the following file:

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

## 7.13 CallbackTimerObserver Class Reference

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

Inheritance diagram for CallbackTimerObserver:



### Public Member Functions

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

#### 7.13.1 Constructor & Destructor Documentation

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

#### 7.13.2 Member Function Documentation

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

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

Referenced by [CallbackTimer< Alloc >::Run\(\)](#).

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

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

## 7.14 CFastRep Class Reference

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

### Public Member Functions

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

### Data Fields

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

### 7.14.1 Detailed Description

For internal use only-- fast string representation

## 7.14.2 Constructor & Destructor Documentation

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

## 7.14.3 Member Function Documentation

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

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

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

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

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

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

## 7.14.4 Field Documentation

7.14.4.1 **OsclAny\* CFastRep::buffer**

7.14.4.2 **uint32 CFastRep::maxsize**

7.14.4.3 **bool CFastRep::overwrite**

7.14.4.4 **uint32 CFastRep::size**

7.14.4.5 **bool CFastRep::writable**

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

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

## 7.15 CHeapRep Class Reference

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

### Public Member Functions

- [CHearpRep \(\)](#)
- 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 Member Functions

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

### Data Fields

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

#### 7.15.1 Detailed Description

For internal use only-- heap string representation

## 7.15.2 Constructor & Destructor Documentation

7.15.2.1 **CHeapRep::CHearpRep () [inline]**

## 7.15.3 Member Function Documentation

7.15.3.1 **OSCL\_IMPORT\_REF void CHearpRep::add\_ref ()**

7.15.3.2 **OSCL\_IMPORT\_REF bool CHearpRep::append (uint32, const oscl\_wchar \*, uint32, const oscl\_wchar \*, Oscl\_DefAlloc &)**

7.15.3.3 **OSCL\_IMPORT\_REF bool CHearpRep::append (uint32, const char \*, uint32, const char \*, Oscl\_DefAlloc &)**

7.15.3.4 **static OSCL\_IMPORT\_REF void CHearpRep::append\_rep (CHearpRep \*&, Oscl\_DefAlloc &, const oscl\_wchar \*, uint32) [static]**

7.15.3.5 **static OSCL\_IMPORT\_REF void CHearpRep::append\_rep (CHearpRep \*&, Oscl\_DefAlloc &, const char \*, uint32) [static]**

7.15.3.6 **static OSCL\_IMPORT\_REF void CHearpRep::assign (CHearpRep \*&, CHearpRep \*, Oscl\_DefAlloc &) [static]**

Referenced by OSCL\_wHeapString< Alloc >::operator=(), OSCL\_HeapString< Alloc >::operator=(), OSCL\_HeapString< Alloc >::OSCL\_HeapString(), and OSCL\_wHeapString< Alloc >::OSCL\_wHeapString().

7.15.3.7 **OSCL\_IMPORT\_REF void CHearpRep::remove\_ref (Oscl\_DefAlloc &)**

Referenced by OSCL\_HeapString< Alloc >::~OSCL\_HeapString(), and OSCL\_wHeapString< Alloc >::~OSCL\_wHeapString().

7.15.3.8 **OSCL\_IMPORT\_REF bool CHearpRep::set (uint32, const oscl\_wchar \*, Oscl\_DefAlloc &)**

7.15.3.9 **OSCL\_IMPORT\_REF bool CHearpRep::set (uint32, const char \*, Oscl\_DefAlloc &)**

7.15.3.10 **static OSCL\_IMPORT\_REF void CHearpRep::set\_rep (CHearpRep \*&, Oscl\_DefAlloc &, const oscl\_wchar \*, uint32) [static]**

7.15.3.11 **static OSCL\_IMPORT\_REF void CHearpRep::set\_rep (CHearpRep \*&, Oscl\_DefAlloc &, const char \*, uint32) [static]**

## 7.15.4 Field Documentation

7.15.4.1 **OsclAny\* CHearpRep::buffer**

Referenced by OSCL\_wHeapString< Alloc >::get\_cstr(), OSCL\_HeapString< Alloc >::get\_cstr(), OSCL\_wHeapString< Alloc >::get\_str(), and OSCL\_HeapString< Alloc >::get\_str().

#### 7.15.4.2 uint32 CHeapRep::maxsize

Referenced by OSCL\_wHeapString< Alloc >::get\_maxsize(), and OSCL\_HeapString< Alloc >::get\_maxsize().

#### 7.15.4.3 uint32 CHeapRep::refcount

#### 7.15.4.4 uint32 CHeapRep::size

Referenced by OSCL\_wHeapString< Alloc >::get\_size(), OSCL\_HeapString< Alloc >::get\_size(), OSCL\_wHeapString< Alloc >::set(), and OSCL\_HeapString< Alloc >::set().

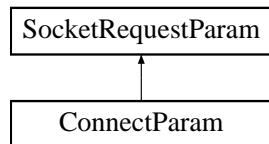
The documentation for this class was generated from the following file:

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

## 7.16 ConnectParam Class Reference

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

Inheritance diagram for ConnectParam:



### Public Member Functions

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

### Data Fields

- [OsclNetworkAddress iAddr](#)

#### 7.16.1 Constructor & Destructor Documentation

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

#### 7.16.2 Field Documentation

**7.16.2.1 OsclNetworkAddress ConnectParam::iAddr**

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

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

## 7.17 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 Member Functions

- `const_iterator()`
- `const_iterator(mapiter x)`
- `const_iterator(const const_iterator &it)`
- `reference operator*() const`
- `pointer operator->() const`
- `bool operator==(const self &x)`
- `bool operator!=(const self &x)`
- `self & operator++()`
- `self operator++(int)`
- `self & operator--()`
- `self operator--(int)`

### Data Fields

- `mapiter mapit`

---

`template<class T, class Alloc> struct Oscl_TagTree< T, Alloc >::const_iterator`

### 7.17.1 Member Typedef Documentation

- 7.17.1.1 `template<class T, class Alloc> typedef map_type::const_iterator Oscl_TagTree< T, Alloc >::const_iterator::mapiter`
- 7.17.1.2 `template<class T, class Alloc> typedef const node_type* Oscl_TagTree< T, Alloc >::const_iterator::pointer`
- 7.17.1.3 `template<class T, class Alloc> typedef const node_type& Oscl_TagTree< T, Alloc >::const_iterator::reference`
- 7.17.1.4 `template<class T, class Alloc> typedef const_iterator Oscl_TagTree< T, Alloc >::const_iterator::self`

### 7.17.2 Constructor & Destructor Documentation

- 7.17.2.1 `template<class T, class Alloc> Oscl_TagTree< T, Alloc >::const_iterator::const_iterator() [inline]`
- 7.17.2.2 `template<class T, class Alloc> Oscl_TagTree< T, Alloc >::const_iterator::const_iterator(mapiter x) [inline]`

References `Oscl_TagTree< T, Alloc >::const_iterator::mapit`.

- 7.17.2.3 `template<class T, class Alloc> Oscl_TagTree< T, Alloc >::const_iterator::const_iterator(const const_iterator & it) [inline]`

References `Oscl_TagTree< T, Alloc >::const_iterator::mapit`.

### 7.17.3 Member Function Documentation

- 7.17.3.1 `template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::const_iterator::operator!= (const self & x) [inline]`

References `Oscl_TagTree< T, Alloc >::const_iterator::mapit`.

- 7.17.3.2 `template<class T, class Alloc> reference Oscl_TagTree< T, Alloc >::const_iterator::operator* () const [inline]`

Referenced by `Oscl_TagTree< T, Alloc >::const_iterator::operator->()`.

- 7.17.3.3 `template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::const_iterator::operator++ (int) [inline]`

- 7.17.3.4 `template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::const_iterator::operator++ () [inline]`

References `Oscl_TagTree< T, Alloc >::const_iterator::mapit`.

7.17.3.5 `template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::const_iterator::operator--(int) [inline]`

7.17.3.6 `template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::const_iterator::operator--() [inline]`

References Oscl\_TagTree< T, Alloc >::const\_iterator::mapit.

7.17.3.7 `template<class T, class Alloc> pointer Oscl_TagTree< T, Alloc >::const_iterator::operator->() const [inline]`

References Oscl\_TagTree< T, Alloc >::const\_iterator::operator\*().

7.17.3.8 `template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::const_iterator::operator==(const self & x) [inline]`

References Oscl\_TagTree< T, Alloc >::const\_iterator::mapit.

## 7.17.4 Field Documentation

7.17.4.1 `template<class T, class Alloc> mapiter Oscl_TagTree< T, Alloc >::const_iterator::mapit`

Referenced by Oscl\_TagTree< T, Alloc >::const\_iterator::const\_iterator(), Oscl\_TagTree< T, Alloc >::const\_iterator::operator!=(), Oscl\_TagTree< T, Alloc >::const\_iterator::operator++(), Oscl\_TagTree< T, Alloc >::const\_iterator::operator--(), and Oscl\_TagTree< T, Alloc >::const\_iterator::operator==( ).

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

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

## 7.18 CStackRep Class Reference

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

### Public Member Functions

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

### Data Fields

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

#### 7.18.1 Detailed Description

For internal use only-- stack string representation

#### 7.18.2 Constructor & Destructor Documentation

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

#### 7.18.3 Member Function Documentation

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

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

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

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

Referenced by [OSCL\\_wStackString< MaxBufSize >::set\(\)](#), and [OSCL\\_StackString< MaxBufSize >::set\(\)](#).

#### 7.18.4 Field Documentation

##### 7.18.4.1 OsclAny\* CStackRep::buffer

Referenced by [OSCL\\_wStackString< MaxBufSize >::get\\_cstr\(\)](#), [OSCL\\_StackString< MaxBufSize >::get\\_cstr\(\)](#), [OSCL\\_wStackString< MaxBufSize >::get\\_str\(\)](#), and [OSCL\\_StackString< MaxBufSize >::get\\_str\(\)](#).

#### 7.18.4.2 uint32 CStackRep::maxsize

Referenced by OSCL\_wStackString< MaxBufSize >::get\_maxsize(), and OSCL\_StackString< MaxBufSize >::get\_maxsize().

#### 7.18.4.3 uint32 CStackRep::size

Referenced by OSCL\_wStackString< MaxBufSize >::get\_size(), OSCL\_StackString< MaxBufSize >::get\_size(), OSCL\_wStackString< MaxBufSize >::set(), and OSCL\_StackString< MaxBufSize >::set().

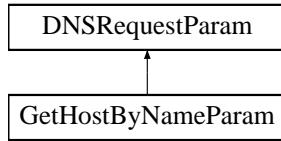
The documentation for this class was generated from the following file:

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

## 7.19 DNSRequestParam Class Reference

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

Inheritance diagram for DNSRequestParam:



### Public Member Functions

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

### Data Fields

- [TPVDNSFxnx iFxn](#)
- [OsclDNSRequest \\* iDNSRequest](#)

### Protected Member Functions

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

### Protected Attributes

- uint32 [iRefCount](#)

#### 7.19.1 Constructor & Destructor Documentation

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

**7.19.1.2** [DNSRequestParam::DNSRequestParam \(TPVDNSFxnx aFxn\) \[protected\]](#)

#### 7.19.2 Member Function Documentation

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

Implemented in [GetHostByNameParam](#).

**7.19.2.2 void DNSRequestParam::RemoveRef ()****7.19.3 Field Documentation****7.19.3.1 OsclDNSRequest\* DNSRequestParam::iDNSRequest****7.19.3.2 TPVDNSFxn DNSRequestParam::iFxn****7.19.3.3 uint32 DNSRequestParam::iRefCount [protected]**

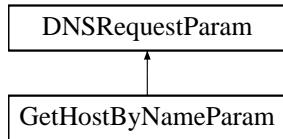
The documentation for this class was generated from the following file:

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

## 7.20 GetHostNameParam Class Reference

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

Inheritance diagram for GetHostNameParam:



### Public Types

- enum { `addressListCapacity` = 10 }

### Public Member Functions

- void `Destroy` ()
- `~GetHostNameParam` ()
- void `PersistHostAddress` (const `OsclNetworkAddress` &`addr`)
- bool `canPersistMoreHostAddresses` ()

### Static Public Member Functions

- static `GetHostNameParam` \* `Create` (const char \*`name`, `OsclNetworkAddress` \*&`addr`, `Oscl_-Vector< OsclNetworkAddress, OsclMemAllocator >` \*`aAddressList`)

### Data Fields

- `char * iName`
- `OsclNetworkAddress * iAddr`
- `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * iAddressList`

#### 7.20.1 Member Enumeration Documentation

##### 7.20.1.1 anonymous enum

Enumerator:

*addressListCapacity*

## 7.20.2 Constructor & Destructor Documentation

7.20.2.1 `GetHostNameParam::~GetHostNameParam ()`

## 7.20.3 Member Function Documentation

7.20.3.1 `bool GetHostNameParam::canPersistMoreHostAddresses () [inline]`

References `Oscl_Vector_Base::capacity()`, `iAddr`, `iAddressList`, `OsclNetworkAddress::ipAddr`, `OSCL_STATIC_CAST`, `oscl_strcmp()`, `Oscl_Vector_Base::size()`, and `OsclNameString<__len>::Str()`.

7.20.3.2 `static GetHostNameParam* GetHostNameParam::Create (const char * name, OsclNetworkAddress *& addr, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * aAddressList) [static]`

7.20.3.3 `void GetHostNameParam::Destroy () [virtual]`

Implements [DNSRequestParam](#).

7.20.3.4 `void GetHostNameParam::PersistHostAddress (const OsclNetworkAddress & addr) [inline]`

References `Oscl_Vector_Base::capacity()`, `iAddr`, `iAddressList`, `OsclNetworkAddress::ipAddr`, `OSCL_STATIC_CAST`, `oscl_strcmp()`, `Oscl_Vector< T, Alloc >::push_back()`, `OsclNameString<__len>::Set()`, `Oscl_Vector_Base::size()`, and `OsclNameString<__len>::Str()`.

## 7.20.4 Field Documentation

7.20.4.1 `OsclNetworkAddress* GetHostNameParam::iAddr`

Referenced by `canPersistMoreHostAddresses()`, and `PersistHostAddress()`.

7.20.4.2 `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >* GetHostNameParam::iAddressList`

Referenced by `canPersistMoreHostAddresses()`, and `PersistHostAddress()`.

7.20.4.3 `char* GetHostNameParam::iName`

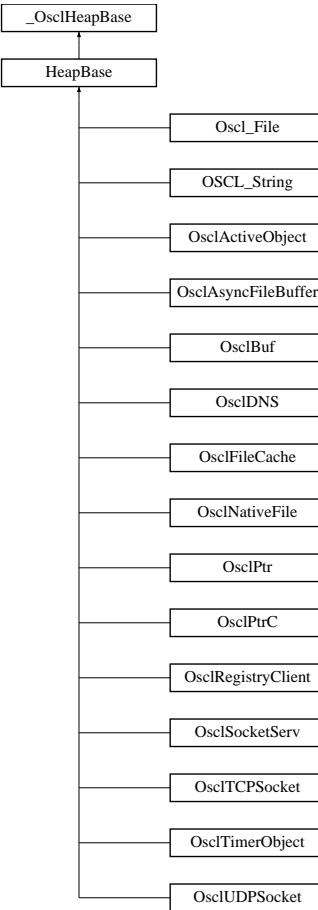
The documentation for this class was generated from the following file:

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

## 7.21 HeapBase Class Reference

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

Inheritance diagram for HeapBase:



### Public Member Functions

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

### Static Public Member Functions

- static void \* [operator new \(size\\_t aSize\)](#)
- static void \* [operator new\[\] \(size\\_t aSize\)](#)
- static void \* [operator new\[\] \(size\\_t aSize, const char \\*aFile=NULL, const int aLine=0\)](#)
- static void \* [operator new \(size\\_t aSize, void \\*aPtr\)](#)
- static void [operator delete \(void \\*aPtr\)](#)
- static void [operator delete\[\] \(void \\*aPtr\)](#)

### 7.21.1 Detailed Description

[HeapBase](#) is the base class for all classes that allocates memory.

[HeapBase](#) has overloaded new and delete operators.

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

[HeapBase](#) has a virtual destructor which calls the destructor of all the derived classes.

### 7.21.2 Constructor & Destructor Documentation

**7.21.2.1 `HeapBase::HeapBase () [inline]`**

**7.21.2.2 `virtual HeapBase::~HeapBase () [inline, virtual]`**

### 7.21.3 Member Function Documentation

**7.21.3.1 `static void HeapBase::operator delete (void * aPtr) [inline, static]`**

References [\\_oscl\\_free\(\)](#).

**7.21.3.2 `static void HeapBase::operator delete[] (void * aPtr) [inline, static]`**

References [\\_oscl\\_free\(\)](#).

**7.21.3.3 `static void* HeapBase::operator new (size_t aSize, void * aPtr) [inline, static]`**

**7.21.3.4 `static void* HeapBase::operator new (size_t aSize) [inline, static]`**

References [\\_oscl\\_default\\_new\(\)](#).

**7.21.3.5 `static void* HeapBase::operator new[] (size_t aSize, const char * aFile = NULL, const int aLine = 0) [inline, static]`**

References [\\_oscl\\_default\\_new\(\)](#), and [OSCL\\_UNUSED\\_ARG](#).

**7.21.3.6 `static void* HeapBase::operator new[] (size_t aSize) [inline, static]`**

References [\\_oscl\\_default\\_new\(\)](#).

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

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

## 7.22 internalLeave Class Reference

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

### Data Fields

- int a

#### 7.22.1 Field Documentation

##### 7.22.1.1 int internalLeave::a

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

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

## 7.23 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 Member Functions

- `iterator ()`
- `iterator (mapiter x)`
- `iterator (const iterator &it)`
- `reference operator* () const`
- `pointer operator-> () const`
- `bool operator== (const self &x)`
- `bool operator!= (const self &x)`
- `self & operator++ ()`
- `self operator++ (int)`
- `self & operator-- ()`
- `self operator-- (int)`

### Data Fields

- `mapiter mapit`

---

`template<class T, class Alloc> struct Oscl_TagTree< T, Alloc >::iterator`

### 7.23.1 Member Typedef Documentation

- 7.23.1.1 `template<class T, class Alloc> typedef map_type::iterator Oscl_TagTree< T, Alloc >::iterator::mapiter`
- 7.23.1.2 `template<class T, class Alloc> typedef node_type* Oscl_TagTree< T, Alloc >::iterator::pointer`
- 7.23.1.3 `template<class T, class Alloc> typedef node_type& Oscl_TagTree< T, Alloc >::iterator::reference`
- 7.23.1.4 `template<class T, class Alloc> typedef iterator Oscl_TagTree< T, Alloc >::iterator::self`

### 7.23.2 Constructor & Destructor Documentation

- 7.23.2.1 `template<class T, class Alloc> Oscl_TagTree< T, Alloc >::iterator::iterator () [inline]`
- 7.23.2.2 `template<class T, class Alloc> Oscl_TagTree< T, Alloc >::iterator::iterator (mapiter x) [inline]`

References `Oscl_TagTree< T, Alloc >::iterator::mapit`.

- 7.23.2.3 `template<class T, class Alloc> Oscl_TagTree< T, Alloc >::iterator::iterator (const iterator & it) [inline]`

References `Oscl_TagTree< T, Alloc >::iterator::mapit`.

### 7.23.3 Member Function Documentation

- 7.23.3.1 `template<class T, class Alloc> bool Oscl_TagTree< T, Alloc >::iterator::operator!= (const self & x) [inline]`

References `Oscl_TagTree< T, Alloc >::iterator::mapit`.

- 7.23.3.2 `template<class T, class Alloc> reference Oscl_TagTree< T, Alloc >::iterator::operator* () const [inline]`

Referenced by `Oscl_TagTree< T, Alloc >::iterator::operator->()`.

- 7.23.3.3 `template<class T, class Alloc> self Oscl_TagTree< T, Alloc >::iterator::operator++ (int) [inline]`

- 7.23.3.4 `template<class T, class Alloc> self& Oscl_TagTree< T, Alloc >::iterator::operator++ 0 [inline]`

References `Oscl_TagTree< T, Alloc >::iterator::mapit`.

7.23.3.5 **template<class T, class Alloc> self Oscl\_TagTree< T, Alloc >::iterator::operator-- (int) [inline]**

7.23.3.6 **template<class T, class Alloc> self& Oscl\_TagTree< T, Alloc >::iterator::operator-- () [inline]**

References Oscl\_TagTree< T, Alloc >::iterator::mapit.

7.23.3.7 **template<class T, class Alloc> pointer Oscl\_TagTree< T, Alloc >::iterator::operator-> () const [inline]**

References Oscl\_TagTree< T, Alloc >::iterator::operator\*().

7.23.3.8 **template<class T, class Alloc> bool Oscl\_TagTree< T, Alloc >::iterator::operator==(const self & x) [inline]**

References Oscl\_TagTree< T, Alloc >::iterator::mapit.

## 7.23.4 Field Documentation

7.23.4.1 **template<class T, class Alloc> mapiter Oscl\_TagTree< T, Alloc >::iterator::mapit**

Referenced by Oscl\_TagTree< T, Alloc >::iterator::iterator(), Oscl\_TagTree< T, Alloc >::iterator::operator!=(), Oscl\_TagTree< T, Alloc >::iterator::operator++(), Oscl\_TagTree< T, Alloc >::iterator::operator--(), and Oscl\_TagTree< T, Alloc >::iterator::operator==( ).

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

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

## 7.24 LinkedListElement< LLClass > Class Template Reference

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

### Public Member Functions

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

### Data Fields

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

#### 7.24.1 Detailed Description

**template<class LLClass> class LinkedListElement< LLClass >**

Linked List Element Class

#### 7.24.2 Constructor & Destructor Documentation

**7.24.2.1 template<class LLClass> LinkedListElement< LLClass >::LinkedListElement (LLClass in\_data) [inline]**

References [LinkedListElement< LLClass >::data](#), [LinkedListElement< LLClass >::next](#), and [NULL](#).

#### 7.24.3 Field Documentation

**7.24.3.1 template<class LLClass> LLClass LinkedListElement< LLClass >::data**

Referenced by [LinkedListElement< LLClass >::LinkedListElement\(\)](#).

**7.24.3.2 template<class LLClass> LinkedListElement<LLClass>\*> LinkedElement< LLClass >::next**

Referenced by [LinkedListElement< LLClass >::LinkedListElement\(\)](#).

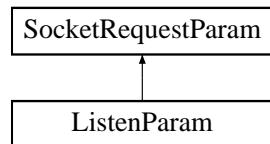
The documentation for this class was generated from the following file:

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

## 7.25 ListenParam Class Reference

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

Inheritance diagram for ListenParam:



### Public Member Functions

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

### Data Fields

- uint32 [iQSize](#)

#### 7.25.1 Constructor & Destructor Documentation

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

#### 7.25.2 Field Documentation

7.25.2.1 [uint32 ListenParam::iQSize](#)

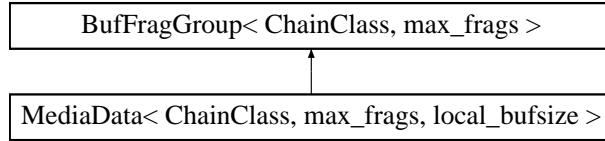
The documentation for this class was generated from the following file:

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

## 7.26 `MediaData< ChainClass, max_frags, local_bufsize >` Class Template Reference

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

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



### Public Member Functions

- `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 Member Functions

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

### Protected Attributes

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

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

### 7.26.1 Constructor & Destructor Documentation

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

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

### 7.26.2 Member Function Documentation

7.26.2.1 `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]`

References `MediaData< ChainClass, max_frags, local_bufsize >::available_localbuf`, `BuffFragStatusClass::BFG_SUCCESS`, `BuffFragGroup< ChainClass, max_frags >::buffer_states`, `BuffFragStatusClass::EMPTY_FRAGMENT`, `BuffFragGroup< ChainClass, max_frags >::fragments`, `OsclMemoryFragment::len`, `BuffFragGroup< ChainClass, max_frags >::length`, `MediaData< ChainClass, max_frags, local_bufsize >::localbuf`, `BuffFragStatusClass::NOT_ENOUGH_SPACE`, `NULL`, `BuffFragGroup< ChainClass, max_frags >::num.fragments`, `oscl_memcpy()`, `oscl_memmove()`, `OsclMemoryFragment::ptr`, and `BuffFragStatusClass::TOO_MANY_FRAGS`.

7.26.2.2 `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 >](#).

References `MediaData< ChainClass, max_frags, local_bufsize >::available_localbuf`, `BuffFragGroup< ChainClass, max_frags >::buffer_states`, `BufferState::decrement_refcnt()`, `BuffFragGroup< ChainClass, max_frags >::fragments`, `BuffFragGroup< ChainClass, max_frags >::length`, `BuffFragGroup< ChainClass, max_frags >::num.fragments`, and `oscl_memset()`.

7.26.2.3 `template<class ChainClass , uint32 max_frags, uint32 local_bufsize> uint32 MediaData< ChainClass, max_frags, local_bufsize >::GetAvailableBufferSize () const [inline]`

References `MediaData< ChainClass, max_frags, local_bufsize >::available_localbuf`.

7.26.2.4 `template<class ChainClass , uint32 max_frags, uint32 local_bufsize> uint32 MediaData< ChainClass, max_frags, local_bufsize >::GetLocalBufsize () const [inline]`

7.26.2.5 `template<class ChainClass , uint32 max_frags, uint32 local_bufsize> MediaStatusClass::status_t MediaData< ChainClass, max_frags, local_bufsize >::GetLocalFragment (BufferFragment &fragment) [inline]`

References `MediaData< ChainClass, max_frags, local_bufsize >::available_localbuf`, `BuffFragStatusClass::BFG_SUCCESS`, `OsclMemoryFragment::len`, `MediaData< ChainClass, max_frags,`

local\_bufsize >::localbuf, BufFragStatusClass::NOT\_ENOUGH\_SPACE, NULL, and OsclMemoryFragment::ptr.

**7.26.2.6 template<class ChainClass , uint32 max\_frags, uint32 local\_bufsize> BufferFragment\* MediaData< ChainClass, max\_frags, local\_bufsize >::GetMediaFragment (const uint32 *idx*) [inline]**

References BufFragGroup< ChainClass, max\_frags >::fragments, NULL, BufFragGroup< ChainClass, max\_frags >::num\_fragments, and MediaData< ChainClass, max\_frags, local\_bufsize >::num\_reserved\_fragments.

**7.26.2.7 template<class ChainClass , uint32 max\_frags, uint32 local\_bufsize> int MediaData< ChainClass, max\_frags, local\_bufsize >::GetMediaSize () const [inline]**

References BufFragGroup< ChainClass, max\_frags >::fragments, BufFragGroup< ChainClass, max\_frags >::length, and MediaData< ChainClass, max\_frags, local\_bufsize >::num\_reserved\_fragments.

**7.26.2.8 template<class ChainClass , uint32 max\_frags, uint32 local\_bufsize> uint32 MediaData< ChainClass, max\_frags, local\_bufsize >::GetNumMediaFrags (const uint32 *idx*) const [inline]**

References BufFragGroup< ChainClass, max\_frags >::num\_fragments, and MediaData< ChainClass, max\_frags, local\_bufsize >::num\_reserved\_fragments.

**7.26.2.9 template<class ChainClass , uint32 max\_frags, uint32 local\_bufsize> MediaTimestamp MediaData< ChainClass, max\_frags, local\_bufsize >::GetTimestamp () const [inline]**

References MediaData< ChainClass, max\_frags, local\_bufsize >::timestamp.

**7.26.2.10 template<class ChainClass , uint32 max\_frags, uint32 local\_bufsize> bool MediaData< ChainClass, max\_frags, local\_bufsize >::IsLocalData (const OsclMemoryFragment & *frag*) const [inline]**

References OsclMemoryFragment::len, MediaData< ChainClass, max\_frags, local\_bufsize >::localbuf, NULL, and OsclMemoryFragment::ptr.

**7.26.2.11 template<class ChainClass , uint32 max\_frags, uint32 local\_bufsize> void MediaData< ChainClass, max\_frags, local\_bufsize >::SetTimestamp (MediaTimestamp *in\_timestamp*) [inline]**

References MediaData< ChainClass, max\_frags, local\_bufsize >::timestamp.

### 7.26.3 Field Documentation

#### 7.26.3.1 template<class ChainClass , uint32 max\_frags, uint32 local\_bufsize> uint32 MediaData<ChainClass, max\_frags, local\_bufsize >::available\_localbuf [protected]

Referenced by MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::Clear(), MediaData< ChainClass, max\_frags, local\_bufsize >::GetAvailableBufferSize(), and MediaData< ChainClass, max\_frags, local\_bufsize >::GetLocalFragment().

#### 7.26.3.2 template<class ChainClass , uint32 max\_frags, uint32 local\_bufsize> uint8 MediaData<ChainClass, max\_frags, local\_bufsize >::localbuf[local\_bufsize] [protected]

Referenced by MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::GetLocalFragment(), and MediaData< ChainClass, max\_frags, local\_bufsize >::IsLocalData().

#### 7.26.3.3 template<class ChainClass , uint32 max\_frags, uint32 local\_bufsize> int MediaData<ChainClass, max\_frags, local\_bufsize >::num\_reserved.fragments [protected]

Referenced by MediaData< ChainClass, max\_frags, local\_bufsize >::GetMediaFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::GetMediaSize(), and MediaData< ChainClass, max\_frags, local\_bufsize >::GetNumMediaFrags().

#### 7.26.3.4 template<class ChainClass , uint32 max\_frags, uint32 local\_bufsize> MediaTimestamp MediaData< ChainClass, max\_frags, local\_bufsize >::timestamp [protected]

Referenced by MediaData< ChainClass, max\_frags, local\_bufsize >::GetTimestamp(), and MediaData< ChainClass, max\_frags, local\_bufsize >::SetTimestamp().

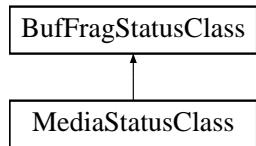
The documentation for this class was generated from the following file:

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

## 7.27 MediaStatusClass Class Reference

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

Inheritance diagram for MediaStatusClass:



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

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

## 7.28 MemAllocator< T > Class Template Reference

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

### Public Types

- `typedef T * pointer`

### Public Member Functions

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

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

#### 7.28.1 Member Typedef Documentation

##### 7.28.1.1 template<class T > `typedef T* MemAllocator< T >::pointer`

#### 7.28.2 Constructor & Destructor Documentation

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

#### 7.28.3 Member Function Documentation

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

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

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

- `oscl_media_data.h`

## 7.29 OsclMemPoolResizableAllocator::MemPoolBlockInfo Struct Reference

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

### Data Fields

- uint32 [iBlockPrefence](#)
- [MemPoolBlockInfo](#) \* [iNextFreeBlock](#)
- [MemPoolBlockInfo](#) \* [iPrevFreeBlock](#)
- uint32 [iBlockSize](#)
- uint8 \* [iBlockBuffer](#)
- [MemPoolBufferInfo](#) \* [iParentBuffer](#)
- uint32 [iBlockPostFence](#)

#### 7.29.1 Field Documentation

7.29.1.1 uint8\* [OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockBuffer](#)

7.29.1.2 uint32 [OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPostFence](#)

7.29.1.3 uint32 [OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockPrefence](#)

7.29.1.4 uint32 [OsclMemPoolResizableAllocator::MemPoolBlockInfo::iBlockSize](#)

7.29.1.5 [MemPoolBlockInfo](#)\* [OsclMemPoolResizableAllocator::MemPoolBlockInfo::iNextFreeBlock](#)

7.29.1.6 [MemPoolBufferInfo](#)\* [OsclMemPoolResizableAllocator::MemPoolBlockInfo::iParentBuffer](#)

7.29.1.7 [MemPoolBlockInfo](#)\* [OsclMemPoolResizableAllocator::MemPoolBlockInfo::iPrevFreeBlock](#)

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

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

## 7.30 OsclMemPoolResizableAllocator::MemPoolBufferInfo Struct Reference

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

### Data Fields

- uint32 [iBufferPreFence](#)
- [OsclAny](#) \* [iStartAddr](#)
- [OsclAny](#) \* [iEndAddr](#)
- uint32 [iBufferSize](#)
- uint32 [iNumOutstanding](#)
- [MemPoolBlockInfo](#) \* [iNextFreeBlock](#)
- [MemPoolBlockInfo](#) \* [iPrevAllocBlock](#)
- uint32 [iAllocatedSz](#)
- uint32 [iBufferPostFence](#)

### 7.30.1 Field Documentation

**7.30.1.1 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iAllocatedSz**

**7.30.1.2 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPostFence**

**7.30.1.3 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferPreFence**

**7.30.1.4 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iBufferSize**

**7.30.1.5 OsclAny\* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iEndAddr**

**7.30.1.6 MemPoolBlockInfo\* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iNextFreeBlock**

**7.30.1.7 uint32 OsclMemPoolResizableAllocator::MemPoolBufferInfo::iNumOutstanding**

**7.30.1.8 MemPoolBlockInfo\* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iPrevAllocBlock**

**7.30.1.9 OsclAny\* OsclMemPoolResizableAllocator::MemPoolBufferInfo::iStartAddr**

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

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

## 7.31 MM\_AllocBlockFence Struct Reference

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

### Public Member Functions

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

### Data Fields

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

#### 7.31.1 Constructor & Destructor Documentation

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

References [fill\\_fence\(\)](#).

#### 7.31.2 Member Function Documentation

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

References [FENCE\\_PATTERN](#), and [pad](#).

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

References [FENCE\\_PATTERN](#), [oscl\\_memset\(\)](#), and [pad](#).

Referenced by [MM\\_AllocBlockFence\(\)](#).

#### 7.31.3 Field Documentation

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

Referenced by [check\\_fence\(\)](#), and [fill\\_fence\(\)](#).

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

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

## 7.32 MM\_AllocBlockHdr Struct Reference

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

### Public Member Functions

- `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

- `static const uint32 ALLOC_NODE_FLAG = 0x80000000`

#### 7.32.1 Constructor & Destructor Documentation

**7.32.1.1 `MM_AllocBlockHdr::MM_AllocBlockHdr () [inline]`**

**7.32.1.2 `MM_AllocBlockHdr::MM_AllocBlockHdr (void *ptr, uint32 inSize) [inline]`**

#### 7.32.2 Member Function Documentation

**7.32.2.1 `bool MM_AllocBlockHdr::isAllocNodePtr () [inline]`**

References ALLOC\_NODE\_FLAG, and size.

**7.32.2.2 `void MM_AllocBlockHdr::setAllocNodeFlag () [inline]`**

References ALLOC\_NODE\_FLAG, and size.

#### 7.32.3 Field Documentation

**7.32.3.1 `uint32 MM_AllocBlockHdr::pad`**

**7.32.3.2 `void* MM_AllocBlockHdr::pNode`**

**7.32.3.3 `void* MM_AllocBlockHdr::pRootNode`**

**7.32.3.4 `uint32 MM_AllocBlockHdr::size`**

Referenced by `isAllocNodePtr()`, and `setAllocNodeFlag()`.

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

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

## 7.33 MM\_AllocInfo Struct Reference

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

### Public Member Functions

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

### Data Fields

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

#### 7.33.1 Constructor & Destructor Documentation

##### 7.33.1.1 MM\_AllocInfo::MM\_AllocInfo () [inline]

References oscl\_memset().

##### 7.33.1.2 MM\_AllocInfo::~MM\_AllocInfo () [inline]

References Oscl\_TAlloc< T, Alloc >::deallocate(), and pFileName.

#### 7.33.2 Member Function Documentation

##### 7.33.2.1 void MM\_AllocInfo::operator delete (void \*ptr) throw () [inline]

References Oscl\_TAlloc< T, Alloc >::deallocate().

##### 7.33.2.2 void\* MM\_AllocInfo::operator new (oscl\_memsize\_t size, MM\_AllocInfo \*ptr) [inline]

References OSCL\_UNUSED\_ARG.

##### 7.33.2.3 void\* MM\_AllocInfo::operator new (oscl\_memsize\_t size) [inline]

References Oscl\_TAlloc< T, Alloc >::allocate(), and OSCL\_UNUSED\_ARG.

### 7.33.3 Field Documentation

7.33.3.1 **uint32 MM\_AllocInfo::allocNum**

7.33.3.2 **bool MM\_AllocInfo::bSetFailure**

7.33.3.3 **uint32 MM\_AllocInfo::lineNo**

7.33.3.4 **char\* MM\_AllocInfo::pFileName**

Referenced by ~MM\_AllocInfo().

7.33.3.5 **void\* MM\_AllocInfo::pMemBlock**

7.33.3.6 **OsclMemStatsNode\* MM\_AllocInfo::pStatsNode**

7.33.3.7 **uint32 MM\_AllocInfo::size**

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

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

## 7.34 MM\_AllocNode Struct Reference

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

### Public Member Functions

- `MM_AllocNode ()`
- `~MM_AllocNode ()`
- `void * operator new (oscl_memsize_t size)`
- `void * operator new (oscl_memsize_t size, MM_AllocNode *ptr)`
- `void operator delete (void *ptr) throw ()`

### Data Fields

- `MM_AllocInfo * pAllocInfo`
- `MM_AllocNode * pPrev`
- `MM_AllocNode * pNext`

#### 7.34.1 Constructor & Destructor Documentation

##### 7.34.1.1 MM\_AllocNode::MM\_AllocNode () [inline]

References `oscl_memset()`.

##### 7.34.1.2 MM\_AllocNode::~MM\_AllocNode () [inline]

References `OSCL_DELETE`, and `pAllocInfo`.

#### 7.34.2 Member Function Documentation

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

References `Oscl_TAlloc< T, Alloc >::deallocate()`.

##### 7.34.2.2 void\* MM\_AllocNode::operator new (oscl\_memsize\_t size, MM\_AllocNode \*ptr) [inline]

References `OSCL_UNUSED_ARG`.

##### 7.34.2.3 void\* MM\_AllocNode::operator new (oscl\_memsize\_t size) [inline]

References `Oscl_TAlloc< T, Alloc >::allocate()`, and `OSCL_UNUSED_ARG`.

### 7.34.3 Field Documentation

#### 7.34.3.1 MM\_AllocInfo\* MM\_AllocNode::pAllocInfo

Referenced by ~MM\_AllocNode().

#### 7.34.3.2 MM\_AllocNode\* MM\_AllocNode::pNext

#### 7.34.3.3 MM\_AllocNode\* MM\_AllocNode::pPrev

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

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

## 7.35 MM\_AllocQueryInfo Struct Reference

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

### Data Fields

- uint32 `allocNum`
- char `fileName` [MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN]
- uint32 `lineNo`
- uint32 `size`
- const void \* `pMemBlock`
- char `tag` [MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN]

#### 7.35.1 Field Documentation

7.35.1.1 uint32 MM\_AllocQueryInfo::allocNum

7.35.1.2 char MM\_AllocQueryInfo::fileName[MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN]

7.35.1.3 uint32 MM\_AllocQueryInfo::lineNo

7.35.1.4 const void\* MM\_AllocQueryInfo::pMemBlock

7.35.1.5 uint32 MM\_AllocQueryInfo::size

7.35.1.6 char MM\_AllocQueryInfo::tag[MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN]

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

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

## 7.36 MM\_Audit\_Imp Class Reference

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

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

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

## 7.37 MM\_AuditOverheadStats Struct Reference

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

### Data Fields

- uint32 [per\\_allocation\\_overhead](#)
- uint32 [stats\\_overhead](#)

#### 7.37.1 Field Documentation

##### 7.37.1.1 uint32 MM\_AuditOverheadStats::per\_allocation\_overhead

##### 7.37.1.2 uint32 MM\_AuditOverheadStats::stats\_overhead

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

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

## 7.38 MM\_FailInsertParam Struct Reference

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

### Public Member Functions

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

### Data Fields

- [uint32 nAllocNum](#)
- [uint16 xsubi \[3\]](#)

#### 7.38.1 Constructor & Destructor Documentation

##### 7.38.1.1 MM\_FailInsertParam::MM\_FailInsertParam () [inline]

References oscl\_memset(), and xsubi.

#### 7.38.2 Member Function Documentation

##### 7.38.2.1 void MM\_FailInsertParam::operator delete (void \*ptr) throw () [inline]

References Oscl\_TAlloc< T, Alloc >::deallocate().

##### 7.38.2.2 void\* MM\_FailInsertParam::operator new (oscl\_memsize\_t size, MM\_FailInsertParam \*ptr) [inline]

References OSCL\_UNUSED\_ARG.

##### 7.38.2.3 void\* MM\_FailInsertParam::operator new (oscl\_memsize\_t size) [inline]

References Oscl\_TAlloc< T, Alloc >::allocate(), and OSCL\_UNUSED\_ARG.

##### 7.38.2.4 void MM\_FailInsertParam::reset () [inline]

References nAllocNum, oscl\_memset(), and xsubi.

Referenced by OsclMemStatsNode::reset().

### 7.38.3 Field Documentation

#### 7.38.3.1 uint32 MM\_FailInsertParam::nAllocNum

Referenced by reset().

#### 7.38.3.2 uint16 MM\_FailInsertParam::xsubi[3]

Referenced by MM\_FailInsertParam(), and reset().

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

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

## 7.39 MM\_Stats\_CB Struct Reference

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

### Public Member Functions

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

### Data Fields

- [const char \\* tag](#)
- [const MM\\_Stats\\_t \\* pStats](#)
- [uint32 num\\_child\\_nodes](#)

#### 7.39.1 Constructor & Destructor Documentation

##### 7.39.1.1 MM\_Stats\_CB::MM\_Stats\_CB () [inline]

References oscl\_memset().

#### 7.39.2 Member Function Documentation

##### 7.39.2.1 void MM\_Stats\_CB::operator delete (void \*ptr) throw () [inline]

References Oscl\_TAlloc< T, Alloc >::deallocate().

##### 7.39.2.2 void\* MM\_Stats\_CB::operator new (oscl\_memsize\_t size, MM\_Stats\_CB \*ptr) [inline]

References OSCL\_UNUSED\_ARG.

##### 7.39.2.3 void\* MM\_Stats\_CB::operator new (oscl\_memsize\_t size) [inline]

References Oscl\_TAlloc< T, Alloc >::allocate(), and OSCL\_UNUSED\_ARG.

#### 7.39.3 Field Documentation

##### 7.39.3.1 uint32 MM\_Stats\_CB::num\_child\_nodes

##### 7.39.3.2 const MM\_Stats\_t\* MM\_Stats\_CB::pStats

##### 7.39.3.3 const char\* MM\_Stats\_CB::tag

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

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

## 7.40 MM\_Stats\_t Struct Reference

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

### Public Member Functions

- [MM\\_Stats\\_t \(\)](#)
- [MM\\_Stats\\_t \(uint32 sizeIn\)](#)
- [void reset \(\)](#)
- [void update \(const MM\\_Stats\\_t &delta, bool add\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size, MM\\_Stats\\_t \\*ptr\)](#)
- [void operator delete \(void \\*ptr\) throw \(\)](#)

### Data Fields

- [uint32 numBytes](#)
- [uint32 peakNumBytes](#)
- [uint32 numAllocs](#)
- [uint32 peakNumAllocs](#)
- [uint32 numAllocFails](#)
- [uint32 totalNumAllocs](#)
- [uint32 totalNumBytes](#)

#### 7.40.1 Constructor & Destructor Documentation

##### 7.40.1.1 MM\_Stats\_t::MM\_Stats\_t () [inline]

References [oscl\\_memset\(\)](#).

##### 7.40.1.2 MM\_Stats\_t::MM\_Stats\_t (uint32 sizeIn) [inline]

References [numAllocFails](#), [numAllocs](#), [numBytes](#), [peakNumAllocs](#), [peakNumBytes](#), [totalNumAllocs](#), and [totalNumBytes](#).

#### 7.40.2 Member Function Documentation

##### 7.40.2.1 void MM\_Stats\_t::operator delete (void \* ptr) throw () [inline]

References [Oscl\\_TAlloc< T, Alloc >::deallocate\(\)](#).

##### 7.40.2.2 void\* MM\_Stats\_t::operator new (oscl\_memsize\_t size, MM\_Stats\_t \* ptr) [inline]

References [OSCL\\_UNUSED\\_ARG](#).

##### 7.40.2.3 void\* MM\_Stats\_t::operator new (oscl\_memsize\_t size) [inline]

References [Oscl\\_TAlloc< T, Alloc >::allocate\(\)](#), and [OSCL\\_UNUSED\\_ARG](#).

**7.40.2.4 void MM\_Stats\_t::reset () [inline]**

References oscl\_memset().

Referenced by OsclMemStatsNode::reset().

**7.40.2.5 void MM\_Stats\_t::update (const MM\_Stats\_t & *delta*, bool *add*) [inline]**

References numAllocFails, numAllocs, numBytes, peakNumAllocs, peakNumBytes, totalNumAllocs, and totalNumBytes.

### 7.40.3 Field Documentation

**7.40.3.1 uint32 MM\_Stats\_t::numAllocFails**

Referenced by MM\_Stats\_t(), and update().

**7.40.3.2 uint32 MM\_Stats\_t::numAllocs**

Referenced by MM\_Stats\_t(), and update().

**7.40.3.3 uint32 MM\_Stats\_t::numBytes**

Referenced by MM\_Stats\_t(), and update().

**7.40.3.4 uint32 MM\_Stats\_t::peakNumAllocs**

Referenced by MM\_Stats\_t(), and update().

**7.40.3.5 uint32 MM\_Stats\_t::peakNumBytes**

Referenced by MM\_Stats\_t(), and update().

**7.40.3.6 uint32 MM\_Stats\_t::totalNumAllocs**

Referenced by MM\_Stats\_t(), and update().

**7.40.3.7 uint32 MM\_Stats\_t::totalNumBytes**

Referenced by MM\_Stats\_t(), and update().

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

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

## 7.41 Oscl\_TagTree< T, Alloc >::Node Struct Reference

```
#include <oscl_tagtree.h>
```

### Public Types

- `typedef Oscl_Vector< Node *, Alloc > children_type`

### Public Member Functions

- `Node ()`
- `void sort_children ()`
- `tag_type::size_type depth ()`

### Data Fields

- `tag_type tag`
- `T value`
- `Node * parent`
- `children_type children`

```
template<class T, class Alloc> struct Oscl_TagTree< T, Alloc >::Node
```

#### 7.41.1 Member Typedef Documentation

7.41.1.1 `template<class T, class Alloc> typedef Oscl_Vector<Node*, Alloc> Oscl_TagTree< T, Alloc >::Node::children_type`

#### 7.41.2 Constructor & Destructor Documentation

7.41.2.1 `template<class T, class Alloc> Oscl_TagTree< T, Alloc >::Node::Node () [inline]`

#### 7.41.3 Member Function Documentation

7.41.3.1 `template<class T, class Alloc> tag_type::size_type Oscl_TagTree< T, Alloc >::Node::depth () [inline]`

References `Oscl_Tag< Alloc >::tag`, `Oscl_TagTree< T, Alloc >::Node::tag`, and `Oscl_Tag_Base::tag_depth()`.

7.41.3.2 `template<class T, class Alloc> void Oscl_TagTree< T, Alloc >::Node::sort_children () [inline]`

References `Oscl_Vector< T, Alloc >::begin()`, `Oscl_TagTree< T, Alloc >::Node::children`, `Oscl_Vector_Base::empty()`, and `Oscl_Vector< T, Alloc >::end()`.

## 7.41.4 Field Documentation

### 7.41.4.1 template<class T, class Alloc> children\_type Oscl\_TagTree< T, Alloc >::Node::children

Referenced by Oscl\_TagTree< T, Alloc >::Node::sort\_children().

### 7.41.4.2 template<class T, class Alloc> Node\* Oscl\_TagTree< T, Alloc >::Node::parent

### 7.41.4.3 template<class T, class Alloc> tag\_type Oscl\_TagTree< T, Alloc >::Node::tag

Referenced by Oscl\_TagTree< T, Alloc >::Node::depth().

### 7.41.4.4 template<class T, class Alloc> T Oscl\_TagTree< T, Alloc >::Node::value

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

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

## 7.42 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 Member Functions

- `OSCL_COND_IMPORT_REF NTPTime ()`  
*The default constructor creates an `NTPTime` instance representing the current system time.*
- `OSCL_COND_IMPORT_REF NTPTime (const NTPTime &src)`  
*Copy constructor to create a new `NTPTime` from an existing one.*
- `OSCL_COND_IMPORT_REF NTPTime (const uint32 seconds)`  
*Construct an `NTPTime` from a uint32.*
- `OSCL_COND_IMPORT_REF NTPTime (const int32 seconds)`  
*Construct an `NTPTime` from a int.*
- `OSCL_COND_IMPORT_REF NTPTime (const TimeValue &t)`  
*Construct a `NTPTime` instance from a `TimeValue` instance.*
- `OSCL_COND_IMPORT_REF NTPTime (const uint64 value)`  
*Construct a `NTPTime` instance from a uint64 value.*
- `OSCL_COND_IMPORT_REF NTPTime & operator= (uint32 newval)`  
*The assignment operator for a 32 bit integer.*
- `OSCL_COND_IMPORT_REF NTPTime & operator= (uint64 newval)`  
*The assignment operator for a 64 bit integer.*
- `OSCL_COND_IMPORT_REF NTPTime & operator+= (uint64 val)`  
*The += operator is used to add a 64 bit 32.32 value to an existing `NTPTime` value.*
- `OSCL_COND_IMPORT_REF NTPTime operator- (const NTPTime &npt) const`  
*The - operator allows subtraction of one `NTPTime` value from another. This is useful to measure an interval.*
- `void set_from_system_time (const uint32 systemtime)`  
*This method converts a 32-bit system time to NTP time.*
- `OSCL_COND_IMPORT_REF uint32 get_middle32 () const`  
*Grab the middle 32 bits of the 64 bit 32.32 representation.*
- `OSCL_COND_IMPORT_REF uint32 get_upper32 () const`  
*This method returns the upper 32 bits of the 32.32 representation.*
- `OSCL_COND_IMPORT_REF uint32 get_lower32 () const`  
*This method returns the lower 32 bits of the 32.32 representation.*

- int32 [to\\_system\\_time \(\) const](#)  
*This method converts the ntp time value to system time.*
- OSCL\_COND\_IMPORT\_REF uint64 [get\\_value \(\) const](#)  
*This method returns the 32.32 ntp representation.*
- OSCL\_IMPORT\_REF int [set\\_to\\_current\\_time \(\)](#)  
*This method sets the 32.32 representation to the current system time value.*

### 7.42.1 Detailed Description

The [NTPTime](#) class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900. The [NTPTime](#) class: Conversion to/from Unix (epoch at 0h Jan. 1, 1970) amount to addition/subtraction of 2208988800. A single 64 bit value is used to represent the time. This value represents the number of seconds since 0h (UTC) Jan. 1, 1900. There is an implied binary point between the upper 32 bits and lower 32 bits (this is referred to as a 32.32 fractional representation). For example a binary value of 00000000 00000000 00000011 10000000 00000000 00000000 00000000 represents 3.5 seconds since Jan 1, 1900.

### 7.42.2 Constructor & Destructor Documentation

#### 7.42.2.1 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime ()

The default constructor creates an [NTPTime](#) instance representing the current system time.

#### 7.42.2.2 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime (const NTPTime & src)

Copy constructor to create a new [NTPTime](#) from an existing one.

#### 7.42.2.3 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime (const uint32 seconds)

Construct an [NTPTime](#) from a uint32.

##### Parameters

**seconds** The uint32 input represents the number of seconds since Jan. 1, 1900.

#### 7.42.2.4 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime (const int32 seconds)

Construct an [NTPTime](#) from a int.

##### Parameters

**seconds** The int input represents the number of seconds since Jan. 1, 1900.

#### 7.42.2.5 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime (const TimeValue & *t*)

Construct a [NTPTime](#) instance from a [TimeValue](#) instance.

This constructor creates an [NTPTime](#) value representing the same absolute time as the [TimeValue](#) parameter.

##### Parameters

*t* A reference to a [TimeValue](#) object.

#### 7.42.2.6 OSCL\_COND\_IMPORT\_REF NTPTime::NTPTime (const uint64 *value*)

Construct a [NTPTime](#) instance from a uint64 value.

##### Parameters

*value* A 64 bit integer argument which is used as the ntp 32.32 fractional representation.

### 7.42.3 Member Function Documentation

#### 7.42.3.1 OSCL\_COND\_IMPORT\_REF uint32 NTPTime::get\_lower32 () const

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

#### 7.42.3.2 OSCL\_COND\_IMPORT\_REF uint32 NTPTime::get\_middle32 () const

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

#### 7.42.3.3 OSCL\_COND\_IMPORT\_REF uint32 NTPTime::get\_upper32 () const

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

#### 7.42.3.4 OSCL\_COND\_IMPORT\_REF uint64 NTPTime::get\_value () const

This method returns the 32.32 ntp representation.

#### 7.42.3.5 OSCL\_COND\_IMPORT\_REF NTPTime& NTPTime::operator+= (uint64 *val*)

The += operator is used to add a 64 bit 32.32 value to an existing [NTPTime](#) value.

##### Parameters

*val* The 64 bit 32.32 value to add to this object's value.

#### 7.42.3.6 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**

*ntp* A reference to the [NTPTime](#) object to be subtracted from this one.

**7.42.3.7 OSCL\_COND\_IMPORT\_REF NTPTime& NTPTime::operator= (uint64 newval)**

The assignment operator for a 64 bit integer.

**Parameters**

*newval* A 64 bit value which represents the 32.32 fractional representation of the ntp time.

**7.42.3.8 OSCL\_COND\_IMPORT\_REF NTPTime& NTPTime::operator= (uint32 newval)**

The assignment operator for a 32 bit integer.

**Parameters**

*newval* A 32 bit integer representing the upper 32 bits of the 32.32 NTP time (e.g. the number of whole seconds since Jan 1, 1900 UTC).

**7.42.3.9 void NTPTime::set\_from\_system\_time (const uint32 systemtime)**

This method converts a 32-bit system time to NTP time.

This method sets the value of the [NTPTime](#) instance to the absolute time represented by the 32 bit input argument.

**Parameters**

*systemtime* This 32-bit value is interpreted as the number of seconds since the unix epoch Jan. 1 1970.

**7.42.3.10 OSCL\_IMPORT\_REF int NTPTime::set\_to\_current\_time ()**

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

**7.42.3.11 int32 NTPTime::to\_system\_time () const**

This method converts the ntp time value to system time.

This method returns a 32 bit value representing the same absolute time as the NTP time value, but expressed as whole seconds since the unix epoch. The fractional part of the ntp value is discarded.

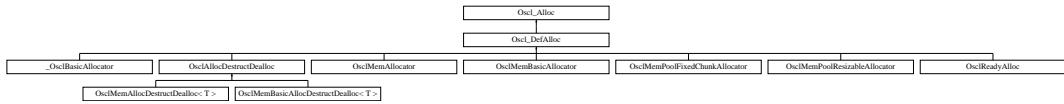
The documentation for this class was generated from the following file:

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

## 7.43 Oscl\_Alloc Class Reference

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

Inheritance diagram for Oscl\_Alloc:



### Public Member Functions

- virtual [~Oscl\\_Alloc \(\)](#)
- virtual [OsclAny \\* allocate \(const uint32 size\)=0](#)
- virtual [OsclAny \\* allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\)](#)

#### 7.43.1 Constructor & Destructor Documentation

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

#### 7.43.2 Member Function Documentation

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

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

Referenced by [allocate\\_fl\(\)](#).

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

Reimplemented in [Oscl\\_DefAlloc](#), and [OsclReadyAlloc](#).

References [allocate\(\)](#), and [OSCL\\_UNUSED\\_ARG](#).

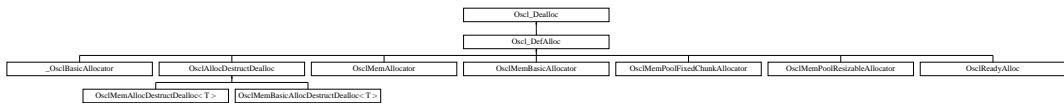
The documentation for this class was generated from the following file:

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

## 7.44 Oscl\_Dealloc Class Reference

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

Inheritance diagram for Oscl\_Dealloc:



### Public Member Functions

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

#### 7.44.1 Constructor & Destructor Documentation

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

#### 7.44.2 Member Function Documentation

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

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

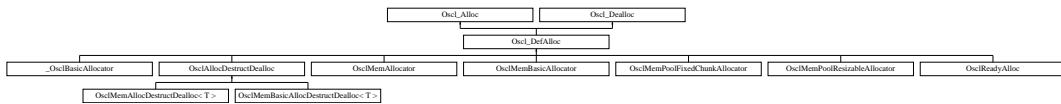
The documentation for this class was generated from the following file:

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

## 7.45 Oscl\_DefAlloc Class Reference

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

Inheritance diagram for Oscl\_DefAlloc:



### Public Member Functions

- virtual [OsclAny](#) \* [allocate](#) (const uint32 size)=0
- virtual [OsclAny](#) \* [allocate\\_fl](#) (const uint32 size, const char \*file\_name, const int line\_num)
- virtual void [deallocate](#) ([OsclAny](#) \*p)=0

#### 7.45.1 Member Function Documentation

##### 7.45.1.1 virtual [OsclAny](#)\* [Oscl\\_DefAlloc::allocate](#) (const uint32 size) [pure virtual]

Implements [Oscl\\_Alloc](#).

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

Referenced by [allocate\\_fl\(\)](#).

##### 7.45.1.2 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 [OsclReadyAlloc](#).

References [allocate\(\)](#), and [OSCL\\_UNUSED\\_ARG](#).

Referenced by [OsclMemBasicAllocDestructDealloc< T >](#)::[allocate\(\)](#), [OsclMemAllocDestructDealloc< T >](#)::[allocate\(\)](#), and [OsclMemAllocator](#)::[allocate\(\)](#).

##### 7.45.1.3 virtual void [Oscl\\_DefAlloc::deallocate](#) ([OsclAny](#) \*p) [pure virtual]

Implements [Oscl\\_Dealloc](#).

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

Referenced by [OsclErrorAllocator](#)::[deallocate\(\)](#).

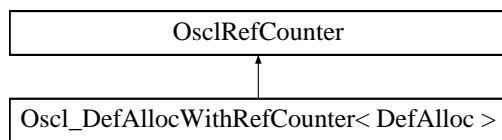
The documentation for this class was generated from the following file:

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

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

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

Inheritance diagram for Oscl\_DefAllocWithRefCounter< DefAlloc >:



### Public Member Functions

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

### Static Public Member Functions

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

#### 7.46.1 Detailed Description

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

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

#### 7.46.2 Member Function Documentation

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

Add to the reference count

Implements [OsclRefCounter](#).

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

Delete object

References [Oscl\\_DefAllocWithRefCounter< DefAlloc >::removeRef\(\)](#).

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

Gets the current number of references

Implements [OsclRefCounter](#).

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

Create object

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

Delete from reference count

Implements [OsclRefCounter](#).

Referenced by [Oscl\\_DefAllocWithRefCounter< DefAlloc >::Delete\(\)](#).

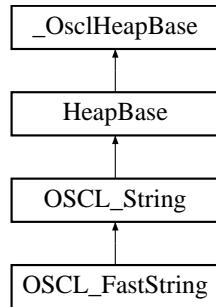
The documentation for this class was generated from the following file:

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

## 7.47 OSCL\_FastString Class Reference

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

Inheritance diagram for OSCL\_FastString:



### Public Types

- `typedef OSCL_String::chartype chartype`
- `typedef TOSCL_StringOp optype`
- `typedef OSCL_wString::chartype other_chartype`

### Public Member Functions

- `OSCL_IMPORT_REF OSCL_FastString()`
- `OSCL_IMPORT_REF OSCL_FastString(const OSCL_FastString &src)`
- `OSCL_IMPORT_REF OSCL_FastString(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_FastString(chartype *buf, uint32 maxlen)`
- `OSCL_IMPORT_REF ~OSCL_FastString()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_FastString & operator=(const OSCL_FastString &src)`
- `OSCL_IMPORT_REF OSCL_FastString & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(chartype *cstr, uint32 maxlen)`
- `OSCL_IMPORT_REF void set(const other_chartype *buf, uint32 numofbyte, optype op)`
- `OSCL_IMPORT_REF void set_length()`

### Friends

- `class OSCL_String`

#### 7.47.1 Detailed Description

`OSCL_FastString` is a simple string class, compatible with regular character array strings.

This class does not allocate internal memory for the string but acts as a container for a user-defined buffer. This means no copying of the string is done and provides a faster way of manipulating strings. Depending on initialization, this container provides either read-only or read-write access to the string.

Implementation assumes the input string is null-terminated.

#### Parameters

*C*,: type of character.

### 7.47.2 Member Typedef Documentation

#### 7.47.2.1 `typedef OSCL_String::chartype OSCL_FastString::chartype`

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

#### 7.47.2.2 `typedef TOSCL_StringOp OSCL_FastString::optype`

#### 7.47.2.3 `typedef OSCL_wString::chartype OSCL_FastString::other_chartype`

### 7.47.3 Constructor & Destructor Documentation

#### 7.47.3.1 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString()`

Default constructor.

#### 7.47.3.2 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(const OSCL_FastString & src)`

Creates a fast string that contains a copy of the input string. The string inherits the writable-ness of the source string.

#### Parameters

*src*,: input string.

#### 7.47.3.3 `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.

#### Parameters

null-terminated string.

#### 7.47.3.4 `OSCL_IMPORT_REF OSCL_FastString::OSCL_FastString(chartype * buf, uint32 maxlen)`

Create the string and initialize it to contain the input string. The string is writable.

#### Parameters

*cp*,: null-terminated string.

*maxlen*,: maximum size of storage at cp, not incl null terminator. If input string is not null-terminated, the function leaves.

#### 7.47.3.5 OSCL\_IMPORT\_REF OSCL\_FastString::~OSCL\_FastString ()

#### 7.47.4 Member Function Documentation

##### 7.47.4.1 OSCL\_IMPORT\_REF const chartype\* OSCL\_FastString::get\_cstr () const [virtual]

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

Implements [OSCL\\_String](#).

##### 7.47.4.2 OSCL\_IMPORT\_REF uint32 OSCL\_FastString::get\_maxsize () const [virtual]

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

Implements [OSCL\\_String](#).

##### 7.47.4.3 OSCL\_IMPORT\_REF uint32 OSCL\_FastString::get\_size () const [virtual]

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

Implements [OSCL\\_String](#).

##### 7.47.4.4 OSCL\_IMPORT\_REF chartype\* OSCL\_FastString::get\_str () const [virtual]

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

Implements [OSCL\\_String](#).

##### 7.47.4.5 OSCL\_IMPORT\_REF OSCL\_FastString& OSCL\_FastString::operator= (const chartype \* *cstr*)

Assignment operator

###### Parameters

null-terminated string

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

##### 7.47.4.6 OSCL\_IMPORT\_REF OSCL\_FastString& OSCL\_FastString::operator= (const OSCL\_FastString & *src*)

Assignment operators

**7.47.4.7 OSCL\_IMPORT\_REF void OSCL\_FastString::set (const other\_chartype \* buf, uint32 numofbyte, optype op)**

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

**Parameters**

*buf*,: string or character array.

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

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

**7.47.4.8 OSCL\_IMPORT\_REF void OSCL\_FastString::set (chartype \* cstr, uint32 maxlen)**

This function can be used to reassign the string to a new writable string. If input string is not null-terminated, the function leaves.

**7.47.4.9 OSCL\_IMPORT\_REF void OSCL\_FastString::set\_length ()**

This function can be used to refresh the string size in case the contents of the string buffer have been modified since the container was created.

**7.47.5 Friends And Related Function Documentation****7.47.5.1 friend class OSCL\_String [friend]**

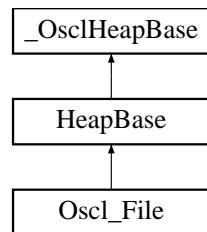
The documentation for this class was generated from the following file:

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

## 7.48 Oscl\_File Class Reference

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

Inheritance diagram for Oscl\_File:



### Data Structures

- class [OsclCacheObserver](#)
- class [OsclFixedCacheParam](#)

### 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 Member Functions

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

- OSCL\_IMPORT\_REF int32 [Seek](#) (TOsclFileOffset offset, [seek\\_type](#) origin)
- OSCL\_IMPORT\_REF TOsclFileOffset [Tell](#) ()
- OSCL\_IMPORT\_REF int32 [Close](#) ()
- OSCL\_IMPORT\_REF int32 [Flush](#) ()
- OSCL\_IMPORT\_REF int32 [SetSize](#) (uint32 size)
- OSCL\_IMPORT\_REF int32 [EndOfFile](#) ()
- OSCL\_IMPORT\_REF int32 [GetError](#) ()
- [OsclHandle \\* Handle](#) ()
- OSCL\_IMPORT\_REF TOsclFileOffset [Size](#) ()
- OSCL\_IMPORT\_REF void [SetLoggingEnable](#) (bool aEnable)
- OSCL\_IMPORT\_REF void [SetSummaryStatsLoggingEnable](#) (bool aEnable)

## Friends

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

### 7.48.1 Member Enumeration Documentation

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

**Enumerator:**

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

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

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

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

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

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

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

**Enumerator:**

**SEEKSET** Beginning of file

**SEEKCUR** Current position of file pointer

**SEEKEND** End of file

#### 7.48.1.3 enum Oscl\_File::TSymbianAccessMode

Defines mode options for SetNativeAccessMode on Symbian.

**Enumerator:**

*ESymbianAccessMode\_Rfile*

*ESymbianAccessMode\_RfileBuf*

### 7.48.2 Constructor & Destructor Documentation

#### 7.48.2.1 OSCL\_IMPORT\_REF Oscl\_File::Oscl\_File()

Constructor

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

Deprecated Constructor, present for back-compatibility.

**Parameters**

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

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

Deprecated Constructor, present for back-compatibility.

**Parameters**

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

*aFileHandle*,: open file handle.

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

Destructor

### 7.48.3 Member Function Documentation

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

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

**Parameters**

*aParam*,: Cache location and size.

References Oscl\_Vector< T, Alloc >::push\_back().

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

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

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

##### Returns

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

#### 7.48.3.3 OSCL\_IMPORT\_REF int32 Oscl\_File::EndOfFile ()

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

##### Returns

#### 7.48.3.4 OSCL\_IMPORT\_REF int32 Oscl\_File::Flush ()

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

##### Returns

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

#### 7.48.3.5 OSCL\_IMPORT\_REF int32 Oscl\_File::GetError ()

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

##### Returns

#### 7.48.3.6 OsclFileHandle\* Oscl\_File::Handle () [inline]

Retrieve the file handle.

##### Returns

file handle

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

Opens a file.

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

**Parameters**

*filename* name of file to open (Unicode)

*mode* combination of open mode flags

*fileserv* fileserv to use

**Returns**

returns 0 if successful and a non-zero value otherwise

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

Opens a file.

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

**Parameters**

*filename* name of file to open (Utf8)

*mode* combination of open mode flags

*fileserv* fileserv to use

**Returns**

returns 0 if successful and a non-zero value otherwise

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

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

**Parameters**

*buffer* pointer to buffer of type void

*size* element size in bytes

*numelements* max number of elements to read

**Returns**

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

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

RemoveFixedCache removes a fixed cache.

##### Parameters

*aPos*,: Cache location and size.

References Oscl\_Vector< T, Alloc >::push\_back().

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

The File Seek operation Sets the position for file pointer

##### Parameters

*offset* offset from the specified origin.

*origin* starting point

##### Returns

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

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

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

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

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

##### Parameters

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

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

#### 7.48.3.14 OSCL\_IMPORT\_REF int32 Oscl\_File::SetFileHandle (OsclFileHandle \* *aHandle*)

SetFileHandle adds an open file handle to the [Oscl\\_File](#) object. The [Oscl\\_File](#) object will use that handle to access the file.

This call is not available when the [Oscl\\_File](#) object is already open.

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

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

**Parameters**

*aHandle*,: container for an open file handle.

**Returns**

returns 0 if successful, non-zero if error.

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

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

**Parameters**

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

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

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

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

**Parameters**

*aMode*,: access mode.

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

SetNativeBufferSize configures the native file buffering feature, when available.

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

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

**Parameters**

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

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

SetPVCacheSize configures the read/write cache.

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

**Parameters**

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

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

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

##### Returns

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

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

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

##### Parameters

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

#### 7.48.3.21 OSCL\_IMPORT\_REF TOsclFileOffset Oscl\_File::Size ()

Get the file size in bytes.

##### Returns

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

#### 7.48.3.22 OSCL\_IMPORT\_REF TOsclFileOffset Oscl\_File::Tell ()

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

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

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

##### Parameters

*buffer* pointer to buffer of type void

*size* element size in bytes

*numelements* number of elements to write

##### Returns

The number of elements written

#### 7.48.4 Friends And Related Function Documentation

7.48.4.1 friend class `asyncfilereadcancel_test` [friend]

7.48.4.2 friend class `asyncfilereadwrite_test` [friend]

7.48.4.3 friend class `largeasyncfilereadwrite_test` [friend]

7.48.4.4 friend class `OsclFileCache` [friend]

7.48.4.5 friend class `OsclFileCacheBuffer` [friend]

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

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

## 7.49 Oscl\_FileFind Class Reference

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

### Public Types

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

### Public Member Functions

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

#### 7.49.1 Detailed Description

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

#### 7.49.2 Member Enumeration Documentation

##### 7.49.2.1 enum Oscl\_FileFind::element\_type

Enumerator:

*FILE\_TYPE*  
*DIR\_TYPE*  
*INVALID\_TYPE*

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

Enumerator:

*E\_OK*

*E\_INVALID\_STATE*  
*E\_INVALID\_ARG*  
*E\_PATH\_TOO\_LONG*  
*E\_PATH\_NOT\_FOUND*  
*E\_NO\_MATCH*  
*E\_BUFFER\_TOO\_SMALL*  
*E\_NOT\_IMPLEMENTED*  
*E\_MEMORY\_ERROR*  
*E\_OTHER*

### 7.49.3 Constructor & Destructor Documentation

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

constructor.

##### Returns

none

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

destructor. will deallocate open handles if necessary

##### Returns

none

### 7.49.4 Member Function Documentation

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

closes the handle to directory.

##### Returns

none

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

Opens a directory for reading.

##### Parameters

*directory* directory to search (utf16).

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

*buf* buffer for returned pathname (utf16).

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

### Returns

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

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

Finds first element matching the pattern.

### Parameters

*directory* directory to search (utf8).

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

*buf* buffer for returned pathname (utf8).

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

### Returns

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

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

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

### Parameters

*buf* buffer to hold directory name(utf16)

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

### Returns

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

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

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

### Parameters

*buf* buffer to hold directory name(utf8)

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

**Returns**

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

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

Returns the element type for the last element returned

**Returns**

see enumeration above for more info.

**7.49.4.7 OSCL\_IMPORT\_REF error\_type Oscl\_FileFind::GetLastError ()**

Returns the error code for the last operation.

**Returns**

see enumeration above for more info.

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

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

## 7.50 Oscl\_FileServer Class Reference

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

### Public Member Functions

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

### Friends

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

#### 7.50.1 Constructor & Destructor Documentation

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

Constructor

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

Destructor

#### 7.50.2 Member Function Documentation

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

Closes a file server.

#### Returns

returns 0 on success and a non-zero value otherwise

##### 7.50.2.2 OSCL\_IMPORT\_REF int32 Oscl\_FileServer::Connect (bool *aShareSession = false*)

Connects the server. This must be called before a file server can be used.

#### Returns

returns 0 on success and a non-zero value otherwise

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

Deletes a file from the filesystem

**Parameters**

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

**Returns**

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

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

Deletes a file from the filesystem \*

**Parameters**

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

**Returns**

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

**7.50.3 Friends And Related Function Documentation****7.50.3.1 friend class Oscl\_File [friend]****7.50.3.2 friend class OsclNativeFile [friend]**

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

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

## 7.51 oscl\_fsstat Struct Reference

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

### Data Fields

- `uint64 freebytes`
- `uint64 totalbytes`

#### 7.51.1 Field Documentation

##### 7.51.1.1 `uint64 oscl_fsstat::freebytes`

##### 7.51.1.2 `uint64 oscl_fsstat::totalbytes`

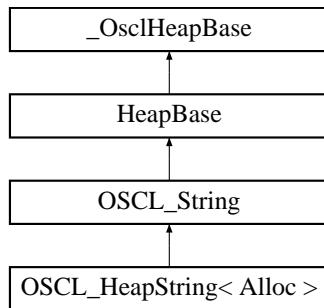
The documentation for this struct was generated from the following file:

- [oscl\\_file\\_dir\\_utils.h](#)

## 7.52 OSCL\_HeapString< Alloc > Class Template Reference

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

Inheritance diagram for OSCL\_HeapString< Alloc >:



### Public Types

- `typedef OSCL_String::chartype chartype`
- `typedef TOSCL_StringOp optype`
- `typedef OSCL_wString::chartype other_chartype`

### Public Member Functions

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

### Friends

- `class OSCL_String`

### 7.52.1 Detailed Description

**template<class Alloc> class OSCL\_HeapString< Alloc >**

[OSCL\\_HeapString](#) is a simple string class, compatible with regular character array strings.

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

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

#### Parameters

*Alloc*,: memory allocator, derived from [Oscl\\_DefAlloc](#).

### 7.52.2 Member Typedef Documentation

**7.52.2.1 template<class Alloc> typedef OSCL\_String::chartype OSCL\_HeapString< Alloc >::chartype**

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

**7.52.2.2 template<class Alloc> typedef TOSCL\_StringOp OSCL\_HeapString< Alloc >::optype**

**7.52.2.3 template<class Alloc> typedef OSCL\_wString::chartype OSCL\_HeapString< Alloc >::other\_chartype**

### 7.52.3 Friends And Related Function Documentation

**7.52.3.1 template<class Alloc> friend class OSCL\_String [friend]**

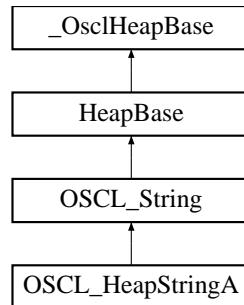
The documentation for this class was generated from the following file:

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

## 7.53 OSCL\_HeapStringA Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_HeapStringA:



### Public Types

- `typedef OSCL_String::chartype chartype`
- `typedef TOSCL_StringOp optype`
- `typedef OSCL_wString::chartype other_chartype`

### Public Member Functions

- `OSCL_IMPORT_REF OSCL_HeapStringA ()`
- `OSCL_IMPORT_REF OSCL_HeapStringA (Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const OSCL_HeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const OSCL_HeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const OSCL_String &src, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const chartype *cstr, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_HeapStringA (const chartype *buf, uint32 length, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF ~OSCL_HeapStringA ()`
- `OSCL_IMPORT_REF uint32 get_size () const`
- `OSCL_IMPORT_REF uint32 get_maxsize () const`
- `OSCL_IMPORT_REF const chartype * get_cstr () const`
- `OSCL_IMPORT_REF chartype * get_str () const`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator= (const OSCL_HeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator= (const OSCL_String &src)`
- `OSCL_IMPORT_REF OSCL_HeapStringA & operator= (const chartype *cstr)`
- `OSCL_IMPORT_REF void set (const chartype *buf, uint32 length)`
- `OSCL_IMPORT_REF void set (const other_chartype *buf, optype op)`
- `OSCL_IMPORT_REF void set (const other_chartype *buf, uint32 length, optype op)`

### Friends

- `class OSCL_String`

### 7.53.1 Detailed Description

`OSCL_HeapStringA` is a simple string class, compatible with regular character array strings. It is similar to `OSCL_HeapString`, except that the allocator is passed at run-time instead of compile-time. The allocator pointer is passed in the constructor, and may be a reference-counted object. If the allocator is not a reference-counted object then it must persist over the lifetime of all `OSCL_HeapStringA` objects that use it. If no allocator is provided, then an `OsclMemAllocator` will be used.

The string array is variable length, is allocated from the heap, and is modifiable. A copy-on-write mechanism is used to minimize unnecessary copying when multiple instances of a string are created for reading. Allocated memory is automatically freed by the class destructor when the last string referencing the memory is destroyed.

The class HAS NO thread synchronization built-in, so it is NOT MT-SAFE. External locks should be used if the class is to be shared across threads.

### 7.53.2 Member Typedef Documentation

#### 7.53.2.1 `typedef OSCL_String::chartype OSCL_HeapStringA::chartype`

Reimplemented from `OSCL_String`.

#### 7.53.2.2 `typedef TOSCL_StringOp OSCL_HeapStringA::optype`

#### 7.53.2.3 `typedef OSCL_wString::chartype OSCL_HeapStringA::other_chartype`

### 7.53.3 Constructor & Destructor Documentation

#### 7.53.3.1 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA()`

The default constructor creates an empty string.

##### Parameters

(optional) allocator or reference-counted allocator.

(optional) reference counter associated with allocator object. If no allocator is provided, this object will use an `OsclMemAllocator`.

#### 7.53.3.2 `OSCL_IMPORT_REF OSCL_HeapStringA::OSCL_HeapStringA(Oscl_DefAlloc *alloc, OsclRefCounter *ref = NULL)`

#### 7.53.3.3 `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.

(optional) allocator or reference-counted allocator.

(optional) reference counter associated with allocator object. If no allocator is provided, this object will use an `OsclMemAllocator`.

- 7.53.3.4 **OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const OSCL\_HeapStringA & *src*, Oscl\_DefAlloc \* *alloc*, OsclRefCounter \* *ref* = NULL)**
- 7.53.3.5 **OSCL\_IMPORT\_REF OSCL\_HeapStringA::OSCL\_HeapStringA (const OSCL\_String & *src*, Oscl\_DefAlloc \* *alloc* = NULL, OsclRefCounter \* *ref* = NULL)**
- 7.53.3.6 **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.  
(optional) allocator or reference-counted allocator.  
(optional) reference counter associated with allocator object. If no allocator is provided, this object will use an [OsclMemAllocator](#).

- 7.53.3.7 **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.  
(optional) allocator or reference-counted allocator.  
(optional) reference counter associated with allocator object. If no allocator is provided, this object will use an [OsclMemAllocator](#).

- 7.53.3.8 **OSCL\_IMPORT\_REF OSCL\_HeapStringA::~OSCL\_HeapStringA ()**

### 7.53.4 Member Function Documentation

- 7.53.4.1 **OSCL\_IMPORT\_REF const chartype\* OSCL\_HeapStringA::get\_cstr () const [virtual]**

This function returns the C-style string for read access.

Implements [OSCL\\_String](#).

- 7.53.4.2 **OSCL\_IMPORT\_REF uint32 OSCL\_HeapStringA::get\_maxsize () const [virtual]**

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implements [OSCL\\_String](#).

**7.53.4.3 OSCL\_IMPORT\_REF uint32 OSCL\_HeapStringA::get\_size () const [virtual]**

Pure virtuals from [OSCL\\_String](#)

Implements [OSCL\\_String](#).

**7.53.4.4 OSCL\_IMPORT\_REF chartype\* OSCL\_HeapStringA::get\_str () const [virtual]**

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implements [OSCL\\_String](#).

**7.53.4.5 OSCL\_IMPORT\_REF OSCL\_HeapStringA& OSCL\_HeapStringA::operator= (const chartype \* *cstr*)**

Assignment operator

**Parameters**

null-terminated string

Reimplemented from [OSCL\\_String](#).

**7.53.4.6 OSCL\_IMPORT\_REF OSCL\_HeapStringA& OSCL\_HeapStringA::operator= (const OSCL\_String & *src*)**

Assignment operator

Reimplemented from [OSCL\\_String](#).

**7.53.4.7 OSCL\_IMPORT\_REF OSCL\_HeapStringA& OSCL\_HeapStringA::operator= (const OSCL\_HeapStringA & *src*)**

Assignment operators

**7.53.4.8 OSCL\_IMPORT\_REF void OSCL\_HeapStringA::set (const other\_chartype \* *buf*, uint32 *length*, optype *op*)**

Set the contents of this string to a new string or character array, with conversion operation.

**Parameters**

*buf*,: string or character array.

*length*,: number of characters to copy.

*op*,: conversion operation to apply

**7.53.4.9 OSCL\_IMPORT\_REF void OSCL\_HeapStringA::set (const other\_chartype \* *buf*, optype *op*)**

Set the contents of this string to a new string, with conversion operation.

**Parameters**

*buf*,: NULL-terminated wide string.

*op*,: conversion operation to apply

**7.53.4.10 OSCL\_IMPORT\_REF void OSCL\_HeapStringA::set (const chartype \* *buf*, uint32 *length*)**

Set the contents of this string to a new string or character array.

**Parameters**

*buf*,: string or character array.

*length*,: number of characters to copy.

**7.53.5 Friends And Related Function Documentation****7.53.5.1 friend class OSCL\_String [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 7.54 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 Member Functions

- static OSCL\_IMPORT\_REF void [set\\_int64](#) (int64 &input\_value, const int32 upper, const uint32 lower)
- static OSCL\_IMPORT\_REF int32 [get\\_int64\\_upper32](#) (const int64 &input\_value)
- static OSCL\_IMPORT\_REF uint32 [get\\_int64\\_lower32](#) (const int64 &input\_value)
- static OSCL\_IMPORT\_REF uint32 [get\\_int64\\_middle32](#) (const int64 &input\_value)
- static OSCL\_IMPORT\_REF void [set\\_uint64](#) (uint64 &input\_value, const uint32 upper, const uint32 lower)
- static OSCL\_IMPORT\_REF uint32 [get\\_uint64\\_upper32](#) (const uint64 &input\_value)
- static OSCL\_IMPORT\_REF uint32 [get\\_uint64\\_lower32](#) (const uint64 &input\_value)
- static OSCL\_IMPORT\_REF uint32 [get\\_uint64\\_middle32](#) (const uint64 &input\_value)

### 7.54.1 Detailed Description

The [Oscl\\_Int64\\_Utils](#) class provides a wrapper for commonly used int64/uint64 operations. The [Oscl\\_Int64\\_Utils](#) class:

Provides a wrapper for commonly used operations to mask the differences between OSes that have an int64/uint64 class instead of a 64-bit integer.

### 7.54.2 Member Function Documentation

- 7.54.2.1 static OSCL\_IMPORT\_REF uint32 Oscl\_Int64\_Utils::get\_int64\_lower32 (const int64 & *input\_value*) [static]
- 7.54.2.2 static OSCL\_IMPORT\_REF uint32 Oscl\_Int64\_Utils::get\_int64\_middle32 (const int64 & *input\_value*) [static]
- 7.54.2.3 static OSCL\_IMPORT\_REF int32 Oscl\_Int64\_Utils::get\_int64\_upper32 (const int64 & *input\_value*) [static]
- 7.54.2.4 static OSCL\_IMPORT\_REF uint32 Oscl\_Int64\_Utils::get\_uint64\_lower32 (const uint64 & *input\_value*) [static]
- 7.54.2.5 static OSCL\_IMPORT\_REF uint32 Oscl\_Int64\_Utils::get\_uint64\_middle32 (const uint64 & *input\_value*) [static]
- 7.54.2.6 static OSCL\_IMPORT\_REF uint32 Oscl\_Int64\_Utils::get\_uint64\_upper32 (const uint64 & *input\_value*) [static]
- 7.54.2.7 static OSCL\_IMPORT\_REF void Oscl\_Int64\_Utils::set\_int64 (int64 & *input\_value*, const int32 *upper*, const uint32 *lower*) [static]
- 7.54.2.8 static OSCL\_IMPORT\_REF void Oscl\_Int64\_Utils::set\_uint64 (uint64 & *input\_value*, const uint32 *upper*, const uint32 *lower*) [static]

The documentation for this class was generated from the following file:

- [oscl\\_int64\\_utils.h](#)

## 7.55 Oscl\_Less< T > Struct Template Reference

```
#include <oscl_map.h>
```

### Public Member Functions

- bool [operator\(\)](#) (const T &x, const T &y) const

```
template<class T> struct Oscl_Less< T >
```

#### 7.55.1 Member Function Documentation

##### 7.55.1.1 template<class T > bool Oscl\_Less< T >::operator() (const T & x, const T & y) const [inline]

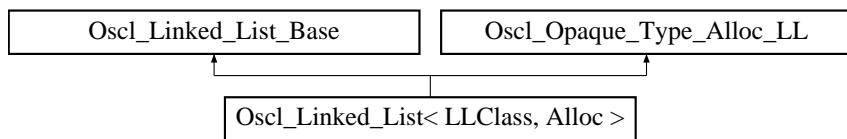
The documentation for this struct was generated from the following file:

- [oscl\\_map.h](#)

## 7.56 Oscl\_Linked\_List< LLClass, Alloc > Class Template Reference

```
#include <oscl_linked_list.h>
```

Inheritance diagram for Oscl\_Linked\_List< LLClass, Alloc >:



### Public Member Functions

- [Oscl\\_Linked\\_List \(\)](#)
- [~Oscl\\_Linked\\_List \(\)](#)
- void [clear \(\)](#)
- int32 [dequeue\\_element \(LLClass &element\)](#)
- int32 [get\\_first \(LLClass &ele\)](#)
- int32 [get\\_next \(LLClass &ele\)](#)
- int32 [check\\_list \(\)](#)
- int32 [get\\_num\\_elements \(\)](#)
- int32 [add\\_element \(LLClass &new\\_element\)](#)
- int32 [add\\_to\\_front \(const LLClass &new\\_element\)](#)
- int32 [insert\\_element \(const LLClass &new\\_element, int index\)](#)
- int32 [get\\_element \(int32 index, LLClass &element\)](#)
- int32 [remove\\_element \(const LLClass &data\\_to\\_remove\)](#)
- int32 [get\\_index \(const LLClass &data\)](#)
- int32 [remove\\_element \(const int32 index\\_to\\_remove\)](#)
- int32 [move\\_to\\_end \(const LLClass &data\\_to\\_move\)](#)
- int32 [move\\_to\\_front \(const LLClass &data\\_to\\_move\)](#)

#### 7.56.1 Detailed Description

**template<class LLClass, class Alloc> class Oscl\_Linked\_List< LLClass, Alloc >**

Oscl Linked List Class

#### 7.56.2 Constructor & Destructor Documentation

**7.56.2.1 template<class LLClass , class Alloc > Oscl\_Linked\_List< LLClass, Alloc >::Oscl\_Linked\_List () [inline]**

Initialized the protected variables of list.

References Oscl\_Linked\_List\_Base::construct(), and Oscl\_Linked\_List\_Base::sizeof\_T.

---

**7.56.2.2 template<class LLClass , class Alloc > Oscl\_Linked\_List< LLClass, Alloc >::~Oscl\_Linked\_List () [inline]**

The destructor.

References Oscl\_Linked\_List\_Base::destroy().

### 7.56.3 Member Function Documentation

**7.56.3.1 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::add\_element (LLClass & *new\_element*) [inline]**

Adds new element to the list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

#### Parameters

*new\_element* the element to be add in the list.

#### Returns

32-bit integer on the success returns 1.

**7.56.3.2 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::add\_to\_front (const LLClass & *new\_element*) [inline]**

Adds new element at the start of the list.if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

#### Parameters

*new\_element* the element to be add in the list.

#### Returns

32-bit integer on the success returns 1.

**7.56.3.3 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::check\_list () [inline]**

Debug routine: Checks the list for elements.

#### Returns

32-bit integer, if node count is equal to number of node added to the list then returns 1 otherwise returns 0.

Reimplemented from [Oscl\\_Linked\\_List\\_Base](#).

**7.56.3.4 template<class LLClass , class Alloc > void Oscl\_Linked\_List< LLClass, Alloc >::clear () [inline]**

References Oscl\_Linked\_List\_Base::destroy().

**7.56.3.5 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::dequeue\_element (LLClass & *element*) [inline]**

References Oscl\_Linked\_List< LLClass, Alloc >::get\_element(), and Oscl\_Linked\_List< LLClass, Alloc >::remove\_element().

**7.56.3.6 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_element (int32 *index*, LLClass & *element*) [inline]**

Search and returns the element in the list for passed index.

**Parameters**

*index,element* The index is the count for the node.

**Returns**

32-bit integer on success returns 1 otherwise returns 0.

Referenced by Oscl\_Linked\_List< LLClass, Alloc >::dequeue\_element().

**7.56.3.7 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_first (LLClass & *ele*) [inline]**

Return the first element of list in passed parameter,

**Parameters**

*ele* return the value of first element of list in this parameter

**Returns**

32-bit interger,If first element found, it returns 1 otherwise it returns 0

**7.56.3.8 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_index (const LLClass & *data*) [inline]**

Returns the index for requested element.

**Parameters**

*data* the element for which index to be return.

**Returns**

32-bit integer if data is found in the list it returns index otherwise it returns -1.

**7.56.3.9 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_next (LLClass & *ele*) [inline]**

Return the next element of list in passed parameter,

**Parameters**

*ele* return the value of next element of list in this parameter

**Returns**

32-bit integer ,if next element is found in list,it returns 1 otherwise it returns 0

**7.56.3.10 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::get\_num\_elements () [inline]**

Get number of elements in the list.

**Returns**

32-bit integer, number of elements in list.

References Oscl\_Linked\_List\_Base::num\_elements.

**7.56.3.11 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::insert\_element (const LLClass & *new\_element*, int *index*) [inline]**

Inserts new element in the list. If the index is past the end of the list it creates the list and add the element as first element of list.

**Parameters**

*new\_element* the element to be add in the list.

**Returns**

32-bit integer on the success returns 1.

**7.56.3.12 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::move\_to\_end (const LLClass & *data\_to\_move*) [inline]**

Moves the element to end of the list

**Parameters**

*data\_to\_move*

**Returns**

On success returns 1 otherwise returns 0.

**7.56.3.13 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::move\_to\_front (const LLClass & *data\_to\_move*) [inline]**

Moves the element to front of the list

**Parameters**

*data\_to\_move*

**Returns**

On success returns 1 otherwise returns 0.

**7.56.3.14 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::remove\_element (const int32 *index\_to\_remove*) [inline]**

Removes the element for requested index.

**Parameters**

*index\_to\_remove*

**Returns**

on success return 1 otherwise return 0.

Reimplemented from [Oscl\\_Linked\\_List\\_Base](#).

References [Oscl\\_Linked\\_List< LLClass, Alloc >::remove\\_element\(\)](#).

**7.56.3.15 template<class LLClass , class Alloc > int32 Oscl\_Linked\_List< LLClass, Alloc >::remove\_element (const LLClass & *data\_to\_remove*) [inline]**

Removes the element from the list.

**Parameters**

*data\_to\_remove*

**Returns**

32-bit integer on if element fount in the list returns 1 otherwise returns 0.

Referenced by [Oscl\\_Linked\\_List< LLClass, Alloc >::dequeue\\_element\(\)](#), and [Oscl\\_Linked\\_List< LLClass, Alloc >::remove\\_element\(\)](#).

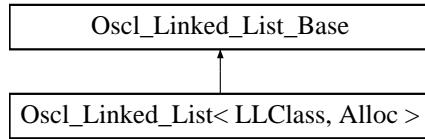
The documentation for this class was generated from the following file:

- [oscl\\_linked\\_list.h](#)

## 7.57 Oscl\_Linked\_List\_Base Class Reference

```
#include <oscl_linked_list.h>
```

Inheritance diagram for Oscl\_Linked\_List\_Base:



### Protected Member Functions

- virtual [~Oscl\\_Linked\\_List\\_Base \(\)](#)
- OSCL\_IMPORT\_REF void [construct \(Oscl\\_Opaque\\_Type\\_Alloc\\_LL \\*op\)](#)
- OSCL\_IMPORT\_REF void [destroy \(\)](#)
- OSCL\_IMPORT\_REF int32 [get\\_first \(OsclAny \\*ele\)](#)
- OSCL\_IMPORT\_REF int32 [get\\_next \(OsclAny \\*ele\)](#)
- OSCL\_IMPORT\_REF int32 [check\\_list \(\)](#)
- OSCL\_IMPORT\_REF int32 [add\\_element \(const OsclAny \\*new\\_element\)](#)
- OSCL\_IMPORT\_REF int32 [add\\_to\\_front \(const OsclAny \\*new\\_element\)](#)
- OSCL\_IMPORT\_REF int32 [insert\\_element \(const OsclAny \\*new\\_element, int index\)](#)
- OSCL\_IMPORT\_REF int32 [get\\_element \(int32 index, OsclAny \\*element\)](#)
- OSCL\_IMPORT\_REF int32 [remove\\_element \(const OsclAny \\*data\\_to\\_remove\)](#)
- OSCL\_IMPORT\_REF int32 [get\\_index \(const OsclAny \\*data\)](#)
- OSCL\_IMPORT\_REF int32 [remove\\_element \(const int32 index\\_to\\_remove\)](#)
- OSCL\_IMPORT\_REF int32 [move\\_to\\_end \(const OsclAny \\*data\\_to\\_move\)](#)
- OSCL\_IMPORT\_REF int32 [move\\_to\\_front \(const OsclAny \\*data\\_to\\_move\)](#)

### Protected Attributes

- [OsclAny \\* head](#)
- [OsclAny \\* tail](#)
- [OsclAny \\* iterator](#)
- int32 [num\\_elements](#)
- uint32 [sizeof\\_T](#)

#### 7.57.1 Detailed Description

Oscl Linked List Base Class. A non-templated base class is used to avoid large inline functions in the [Oscl\\_Linked\\_List](#) implementation.

### 7.57.2 Constructor & Destructor Documentation

**7.57.2.1 virtual Oscl\_Linked\_List\_Base::~Oscl\_Linked\_List\_Base () [inline, protected, virtual]**

### 7.57.3 Member Function Documentation

**7.57.3.1 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::add\_element (const OsclAny \* new\_element) [protected]**

Adds new element to the list.if list is already there then it adds element at end of list otherwise it create the list and add the element as first element of list.

#### Parameters

*new\_element* the element to be add in the list.

#### Returns

32-bit integer on the success returns 1.

**7.57.3.2 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::add\_to\_front (const OsclAny \* new\_element) [protected]**

Adds new element at the start of the list.if list is already exist then it adds element at start of list otherwise it create the list and add the element as first element of list.

#### Parameters

*new\_element* the element to be add in the list.

#### Returns

32-bit integer on the success returns 1.

**7.57.3.3 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::check\_list () [protected]**

Debug routine: Checks the list for elements.

#### Returns

32-bit integer, if node count is equal to number of node added to the list then returns 1 otherwise returns 0.

Reimplemented in [Oscl\\_Linked\\_List< LLClass, Alloc >](#).

**7.57.3.4 OSCL\_IMPORT\_REF void Oscl\_Linked\_List\_Base::construct (Oscl\_Opaque\_Type\_Alloc\_LL \* op) [protected]**

Referenced by [Oscl\\_Linked\\_List< LLClass, Alloc >::Oscl\\_Linked\\_List\(\)](#).

### 7.57.3.5 OSCL\_IMPORT\_REF void Oscl\_Linked\_List\_Base::destroy () [protected]

Referenced by Oscl\_Linked\_List< LLClass, Alloc >::clear(), and Oscl\_Linked\_List< LLClass, Alloc >::~Oscl\_Linked\_List().

### 7.57.3.6 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::get\_element (int32 *index*, OsclAny \* *element*) [protected]

Search and returns the element in the list for passed index.

#### Parameters

*index,element* The index is the count for the node.

#### Returns

32-bit integer on success returns 1 otherwise returns 0.

### 7.57.3.7 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 integer, If first element found, it returns 1 otherwise it returns 0

### 7.57.3.8 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::get\_index (const OsclAny \* *data*) [protected]

Returns the index for requested element.

#### Parameters

*data* the element for which index to be return.

#### Returns

32-bit integer if data is found in the list it returns index otherwise it returns -1.

### 7.57.3.9 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 integer ,if next element is found in list,it returns 1 otherwise it returns 0

**7.57.3.10 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::insert\_element (const OsclAny \* new\_element, int index) [protected]**

Inserts new element in the list. If the index is past the end of the list it creates the list and add the element as first element of list.

**Parameters**

*new\_element* the element to be add in the list.

**Returns**

32-bit integer on the success returns 1.

**7.57.3.11 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::move\_to\_end (const OsclAny \* data\_to\_move) [protected]**

Moves the element to end of the list

**Parameters**

*data\_to\_move*

**Returns**

On success returns 1 otherwise returns 0.

**7.57.3.12 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::move\_to\_front (const OsclAny \* data\_to\_move) [protected]**

Moves the element to front of the list

**Parameters**

*data\_to\_move*

**Returns**

On success returns 1 otherwise returns 0.

**7.57.3.13 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::remove\_element (const int32 index\_to\_remove) [protected]**

Removes the element for requested index.

**Parameters**

*index\_to\_remove*

**Returns**

on success return 1 otherwise return 0.

Reimplemented in [Oscl\\_Linked\\_List< LLClass, Alloc >](#).

**7.57.3.14 OSCL\_IMPORT\_REF int32 Oscl\_Linked\_List\_Base::remove\_element (const OsclAny \*  
data\_to\_remove) [protected]**

Removes the element from the list.

**Parameters**

*data\_to\_remove*

**Returns**

32-bit integer on if element fount in the list returns 1 otherwise returns 0.

**7.57.4 Field Documentation****7.57.4.1 OsclAny\* Oscl\_Linked\_List\_Base::head [protected]****7.57.4.2 OsclAny\* Oscl\_Linked\_List\_Base::iterator [protected]****7.57.4.3 int32 Oscl\_Linked\_List\_Base::num\_elements [protected]**

Referenced by Oscl\_Linked\_List< LLClass, Alloc >::get\_num\_elements().

**7.57.4.4 uint32 Oscl\_Linked\_List\_Base::sizeof\_T [protected]**

Referenced by Oscl\_Linked\_List< LLClass, Alloc >::Oscl\_Linked\_List().

**7.57.4.5 OsclAny\* Oscl\_Linked\_List\_Base::tail [protected]**

The documentation for this class was generated from the following file:

- [oscl\\_linked\\_list.h](#)

## 7.58 Oscl\_Map< Key, T, Alloc, Compare > Class Template Reference

```
#include <oscl_map.h>
```

### Data Structures

- class [value\\_compare](#)

### 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\\_criterator\\_criterator](#)

### Public Member Functions

- [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_equal_range (const key_type &x)`
- `pair_citerator_citerator_equal_range (const key_type &x) const`

### 7.58.1 Detailed Description

`template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> class Oscl_Map< Key, T, Alloc, Compare >`

**Oscl\_Map** Class. A subset of STL::Map methods. **Oscl\_Map** is a sorted associative container that associates objects of type Key with objects of type T. It is also a unique associative container, meaning that no two elements have the same key. **Oscl\_Map** uses the key to speed lookup, insertion, and deletion of elements. It is often superior to all other containers when you need to lookup an element by key value. Due to the underlying tree structure, inserts and erases can be performed in logarithmic time, where a vector would take linear time in some cases.

## 7.58.2 Member Typedef Documentation

- 7.58.2.1 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::const_iterator Oscl_Map< Key, T, Alloc, Compare >::const_iterator`
- 7.58.2.2 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::const_reference Oscl_Map< Key, T, Alloc, Compare >::const_reference`
- 7.58.2.3 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::iterator Oscl_Map< Key, T, Alloc, Compare >::iterator`
- 7.58.2.4 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Compare Oscl_Map< Key, T, Alloc, Compare >::key_compare`
- 7.58.2.5 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Key Oscl_Map< Key, T, Alloc, Compare >::key_type`
- 7.58.2.6 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Oscl_Pair<const_iterator, const_iterator> Oscl_Map< Key, T, Alloc, Compare >::pair_citerator_citerator`
- 7.58.2.7 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Oscl_Pair<iterator, bool> Oscl_Map< Key, T, Alloc, Compare >::pair_iterator_bool`
- 7.58.2.8 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Oscl_Pair<iterator, iterator> Oscl_Map< Key, T, Alloc, Compare >::pair_iterator_iterator`
- 7.58.2.9 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::pointer Oscl_Map< Key, T, Alloc, Compare >::pointer`
- 7.58.2.10 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::reference Oscl_Map< Key, T, Alloc, Compare >::reference`
- 7.58.2.11 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Oscl_Map<Key, T, Alloc, Compare> Oscl_Map< Key, T, Alloc, Compare >::self`
- 7.58.2.12 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef rep_type::size_type Oscl_Map< Key, T, Alloc, Compare >::size_type`
- 7.58.2.13 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> typedef Oscl_Pair<const Key, T> Oscl_Map< Key, T, Alloc, Compare >::value_type`

## 7.58.3 Constructor & Destructor Documentation

- 7.58.3.1 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Oscl_Map< Key, T, Alloc, Compare >::Oscl_Map (const Compare & comp = Compare ()) [inline]`

Creates an empty map using comp as the key compare object

---

**7.58.3.2 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> Oscl\_Map<Key, T, Alloc, Compare >::Oscl\_Map (const self & x) [inline]**

[Oscl\\_Map](#) copy constructor

#### 7.58.4 Member Function Documentation

**7.58.4.1 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::begin () const [inline]**

Returns a const iterator pointing to the beginning of the map

**7.58.4.2 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

Referenced by [Oscl\\_TagTree< PVLogger \\*, alloc\\_type >::begin\(\)](#).

**7.58.4.3 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void Oscl\_Map< Key, T, Alloc, Compare >::clear () [inline]**

Erases all elements

Referenced by [Oscl\\_TagTree< PVLogger \\*, alloc\\_type >::clear\(\)](#).

**7.58.4.4 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.

Referenced by [Oscl\\_TagTree< PVLogger \\*, alloc\\_type >::count\(\)](#).

**7.58.4.5 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

Referenced by [Oscl\\_TagTree< PVLogger \\*, alloc\\_type >::empty\(\)](#).

**7.58.4.6 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::end () const [inline]**

Returns a const iterator pointing to the end of the map.

**7.58.4.7 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.

Referenced by Oscl\_TagTree< PVLogger \*, alloc\_type >::end(), and Oscl\_TagTree< PVLogger \*, alloc\_type >::insert().

**7.58.4.8 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
pair\_citerator\_citerator Oscl\_Map< Key, T, Alloc, Compare >::equal\_range (const  
key\_type & x) const [inline]**

Finds a range containing all elements whose key is x

**7.58.4.9 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
pair\_iterator\_iterator Oscl\_Map< Key, T, Alloc, Compare >::equal\_range (const  
key\_type & x) [inline]**

Finds a range containing all elements whose key is x

**7.58.4.10 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void  
Oscl\_Map< Key, T, Alloc, Compare >::erase (iterator first, iterator last) [inline]**

Erases all elements in the range [first,last)

**7.58.4.11 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> size\_type  
Oscl\_Map< Key, T, Alloc, Compare >::erase (const key\_type & x) [inline]**

Erases the element with key x

**7.58.4.12 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

Referenced by Oscl\_TagTree< PVLogger \*, alloc\_type >::erase().

**7.58.4.13 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::find (const key\_type & x) const [inline]**

Finds an element whose key is x

**7.58.4.14 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

Referenced by Oscl\_TagTree< PVLogger \*, alloc\_type >::find(), and Oscl\_TagTree< PVLogger \*, alloc\_type >::insert().

---

**7.58.4.15 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> void Oscl\_Map< Key, T, Alloc, Compare >::insert (const value\_type \*first, const value\_type \*last) [inline]**

Inserts the range [first,last) into the map

**7.58.4.16 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator Oscl\_Map< Key, T, Alloc, Compare >::insert (iterator position, const value\_type & x) [inline]**

Inserts x into the map using position as a hint as to where it should be inserted

**7.58.4.17 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

Referenced by Oscl\_TagTree< PVLogger \*, alloc\_type >::insert(), Oscl\_Map< const tag\_base\_type, node\_ptr, alloc\_type, Oscl\_Tag\_Base >::operator[](), and Oscl\_TagTree< PVLogger \*, alloc\_type >::Oscl\_TagTree().

**7.58.4.18 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> key\_compare Oscl\_Map< Key, T, Alloc, Compare >::key\_comp () const [inline]**

Returns the key compare object used by the map

**7.58.4.19 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::lower\_bound (const key\_type & x) const [inline]**

Finds the first element whose key is not less than x

**7.58.4.20 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator Oscl\_Map< Key, T, Alloc, Compare >::lower\_bound (const key\_type & x) [inline]**

Finds the first element whose key is not less than x

**7.58.4.21 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> size\_type Oscl\_Map< Key, T, Alloc, Compare >::max\_size () const [inline]**

Returns the maximum possible size of the map

**7.58.4.22 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> self& Oscl\_Map< Key, T, Alloc, Compare >::operator= (const self & x) [inline]**

Oscl\_Map assignment operator

**7.58.4.23 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> T& Oscl\_Map< Key, T, Alloc, Compare >::operator[ ] (const key\_type & k) [inline]**

Returns a reference to the object that is associated with a particular key. If the map does not already contain such an object, operator[] inserts the default object [value\\_type\(\)](#).

**7.58.4.24 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

Referenced by [Oscl\\_TagTree< PVLogger \\*, alloc\\_type >::size\(\)](#).

**7.58.4.25 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> const\_iterator Oscl\_Map< Key, T, Alloc, Compare >::upper\_bound (const key\_type & x) const [inline]**

Finds the first element whose key is not greater than x

**7.58.4.26 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> iterator Oscl\_Map< Key, T, Alloc, Compare >::upper\_bound (const key\_type & x) [inline]**

Finds the first element whose key is not greater than x

**7.58.4.27 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> value\_compare Oscl\_Map< Key, T, Alloc, Compare >::value\_comp () const [inline]**

Returns the value compare object used by the map

The documentation for this class was generated from the following file:

- [oscl\\_map.h](#)

## 7.59 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock > Class Template Reference

```
#include <oscl_linked_list.h>
```

### Public Member Functions

- [Oscl\\_MTLinked\\_List \(\)](#)
- [~Oscl\\_MTLinked\\_List \(\)](#)
- int32 [dequeue\\_element \(LLClass &element\)](#)
- int32 [add\\_element \(LLClass &new\\_element\)](#)
- int32 [add\\_to\\_front \(LLClass &new\\_element\)](#)
- uint32 [get\\_element \(int32 index, LLClass &element\)](#)
- int32 [remove\\_element \(const LLClass &data\\_to\\_remove\)](#)
- int32 [get\\_index \(const LLClass &data\)](#)
- int32 [remove\\_element \(const int32 index\\_to\\_remove\)](#)
- int32 [move\\_to\\_end \(const LLClass &data\\_to\\_move\)](#)
- int32 [move\\_to\\_front \(const LLClass &data\\_to\\_move\)](#)

### Protected Attributes

- [Oscl\\_Linked\\_List< LLClass, Alloc > the\\_list](#)

#### 7.59.1 Detailed Description

**template<class LLClass, class Alloc, class TheLock> class Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >**

[Oscl\\_MTLinked\\_List](#) is a multi-threaded version of the [LinkedList](#). It has mutex protection to allow access by different threads.

#### 7.59.2 Constructor & Destructor Documentation

##### 7.59.2.1 template<class LLClass , class Alloc , class TheLock > Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::Oscl\_MTLinked\_List () [inline]

Constructor for [Oscl\\_MTLinked\\_List](#)

##### 7.59.2.2 template<class LLClass , class Alloc , class TheLock > Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::~Oscl\_MTLinked\_List () [inline]

Destructor for [Oscl\\_MTLinked\\_List](#)

### 7.59.3 Member Function Documentation

#### 7.59.3.1 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.

References Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list.

#### 7.59.3.2 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.

References Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list.

#### 7.59.3.3 template<class LLClass , class Alloc , class TheLock > int32 Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::dequeue\_element (LLClass & element) [inline]

References Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list.

#### 7.59.3.4 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.

References Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list.

**7.59.3.5 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.

References Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list.

**7.59.3.6 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.

References Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list.

**7.59.3.7 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.

References Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list.

**7.59.3.8 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.

References Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list.

**7.59.3.9 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.

References Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list.

### 7.59.4 Field Documentation

**7.59.4.1 template<class LLClass , class Alloc , class TheLock > Oscl\_Linked\_List<LLClass, Alloc> Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::the\_list [protected]**

Referenced by Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::add\_element(), Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::add\_to\_front(), Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::dequeue\_element(), Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::get\_element(), Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::get\_index(), Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::move\_to\_end(), Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::move\_to\_front(), and Oscl\_MTLinked\_List< LLClass, Alloc, TheLock >::remove\_element().

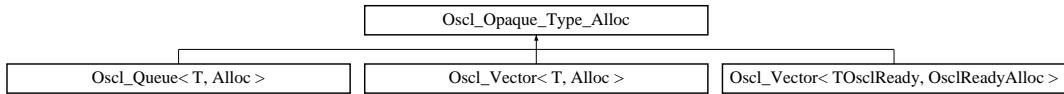
The documentation for this class was generated from the following file:

- [oscl\\_linked\\_list.h](#)

## 7.60 Oscl\_Opaque\_Type\_Alloc Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl\_Opaque\_Type\_Alloc:



### Public Member Functions

- virtual ~Oscl\_Opaque\_Type\_Alloc ()
- virtual void construct (OsclAny \*p, const OsclAny \*init\_val)=0
- virtual void destroy (OsclAny \*p)=0
- virtual OsclAny \* allocate (const uint32 size)=0
- virtual void deallocate (OsclAny \*p)=0

#### 7.60.1 Detailed Description

This class combines opaque type operations with memory allocation operations.

#### 7.60.2 Constructor & Destructor Documentation

**7.60.2.1 virtual Oscl\_Opaque\_Type\_Alloc::~Oscl\_Opaque\_Type\_Alloc () [inline, virtual]**

#### 7.60.3 Member Function Documentation

**7.60.3.1 virtual OsclAny\* Oscl\_Opaque\_Type\_Alloc::allocate (const uint32 size) [pure virtual]**

Allocate "size" bytes

**7.60.3.2 virtual void Oscl\_Opaque\_Type\_Alloc::construct (OsclAny \* p, const OsclAny \* init\_val) [pure virtual]**

Construct element at p using element at init\_val as the initial value. Both pointers must be non-NULL.

**7.60.3.3 virtual void Oscl\_Opaque\_Type\_Alloc::deallocate (OsclAny \* p) [pure virtual]**

Deallocate memory previously allocated with "allocate"

**7.60.3.4 virtual void Oscl\_Opaque\_Type\_Alloc::destroy (OsclAny \* p) [pure virtual]**

Destroy element at p.

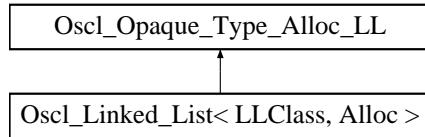
The documentation for this class was generated from the following file:

- [oscl\\_opaque\\_type.h](#)

## 7.61 Oscl\_Opaque\_Type\_Alloc\_LL Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl\_Opaque\_Type\_Alloc\_LL:



### Public Member Functions

- virtual ~Oscl\_Opaque\_Type\_Alloc\_LL()
- virtual void construct (OsclAny \*p, const OsclAny \*init\_val)=0
- virtual void destroy (OsclAny \*p)=0
- virtual OsclAny \* allocate (const uint32 size)=0
- virtual void deallocate (OsclAny \*p)=0
- virtual OsclAny \* get\_next (const OsclAny \*elem) const =0
- virtual void set\_next (OsclAny \*elem, const OsclAny \*nextelem)=0
- virtual void get\_data (OsclAny \*elem, OsclAny \*data\_val)=0
- virtual bool compare\_data (const OsclAny \*elem, const OsclAny \*data\_val) const =0

#### 7.61.1 Detailed Description

This class combines opaque type operations with memory allocation operations and linked list support

#### 7.61.2 Constructor & Destructor Documentation

**7.61.2.1 virtual Oscl\_Opaque\_Type\_Alloc\_LL::~Oscl\_Opaque\_Type\_Alloc\_LL () [inline, virtual]**

#### 7.61.3 Member Function Documentation

**7.61.3.1 virtual OsclAny\* Oscl\_Opaque\_Type\_Alloc\_LL::allocate (const uint32 size) [pure virtual]**

Allocate "size" bytes

**7.61.3.2 virtual bool Oscl\_Opaque\_Type\_Alloc\_LL::compare\_data (const OsclAny \* elem, const OsclAny \* data\_val) const [pure virtual]**

Compare data.

**7.61.3.3 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::construct (OsclAny \* p, const OsclAny \* init\_val) [pure virtual]**

Construct element at p using element at init\_val as the initial value. Both pointers must be non-NULL.

**7.61.3.4 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::deallocate (OsclAny \* *p*) [pure virtual]**

Deallocate memory previously allocated with "allocate"

**7.61.3.5 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::destroy (OsclAny \* *p*) [pure virtual]**

Destroy element at p.

**7.61.3.6 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::get\_data (OsclAny \* *elem*, OsclAny \* *data\_val*) [pure virtual]**

Get data

**7.61.3.7 virtual OsclAny\* Oscl\_Opaque\_Type\_Alloc\_LL::get\_next (const OsclAny \* *elem*) const [pure virtual]**

Get next element in linked list.

**7.61.3.8 virtual void Oscl\_Opaque\_Type\_Alloc\_LL::set\_next (OsclAny \* *elem*, const OsclAny \* *nextelem*) [pure virtual]**

Set next element in linked list.

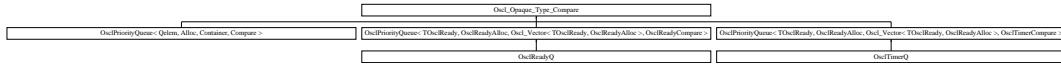
The documentation for this class was generated from the following file:

- [oscl\\_opaque\\_type.h](#)

## 7.62 Oscl\_Opaque\_Type\_Compare Class Reference

```
#include <oscl_opaque_type.h>
```

Inheritance diagram for Oscl\_Opaque\_Type\_Compare:



### Public Member Functions

- virtual ~Oscl\_Opaque\_Type\_Compare ()
- virtual void swap (OsclAny \*a, const OsclAny \*b)=0
- virtual int compare\_LT (OsclAny \*a, OsclAny \*b) const =0
- virtual int compare\_EQ (const OsclAny \*a, const OsclAny \*b) const =0

#### 7.62.1 Detailed Description

Opaque type operations with swap & comparisons.

#### 7.62.2 Constructor & Destructor Documentation

**7.62.2.1** virtual Oscl\_Opaque\_Type\_Compare::~Oscl\_Opaque\_Type\_Compare () [inline, virtual]

#### 7.62.3 Member Function Documentation

**7.62.3.1** virtual int Oscl\_Opaque\_Type\_Compare::compare\_EQ (const OsclAny \* a, const OsclAny \* b) const [pure virtual]

Return a==b.

Implemented in OsclPriorityQueue< Qelem, Alloc, Container, Compare >, OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >, and OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >.

**7.62.3.2** virtual int Oscl\_Opaque\_Type\_Compare::compare\_LT (OsclAny \* a, OsclAny \* b) const [pure virtual]

Return a<b.

Implemented in OsclPriorityQueue< Qelem, Alloc, Container, Compare >, OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >, and OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >.

**7.62.3.3 virtual void Oscl\_Opaque\_Type\_Compare::swap (OsclAny \* *a*, const OsclAny \* *b*)  
[pure virtual]**

Swap element at "a" with element at "b". Both pointers must be non-NULL.

Implemented in [OsclPriorityQueue< Qelem, Alloc, Container, Compare >](#), [OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\\_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >](#), and [OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >](#).

The documentation for this class was generated from the following file:

- [oscl\\_opaque\\_type.h](#)

## 7.63 Oscl\_Pair< T1, T2 > Struct Template Reference

```
#include <oscl_tree.h>
```

### Public Member Functions

- [Oscl\\_Pair \(\)](#)
- [Oscl\\_Pair \(const T1 &a, const T2 &b\)](#)

### Data Fields

- T1 [first](#)
- T2 [second](#)

```
template<class T1, class T2> struct Oscl_Pair< T1, T2 >
```

#### 7.63.1 Constructor & Destructor Documentation

**7.63.1.1 template<class T1, class T2> Oscl\_Pair< T1, T2 >::Oscl\_Pair () [inline]**

**7.63.1.2 template<class T1, class T2> Oscl\_Pair< T1, T2 >::Oscl\_Pair (const T1 & a, const T2 & b) [inline]**

#### 7.63.2 Field Documentation

**7.63.2.1 template<class T1, class T2> T1 Oscl\_Pair< T1, T2 >::first**

Referenced by Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::count(), Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::erase(), Oscl\_TagTree< PVLogger \*, alloc\_type >::insert(), and Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::operator()().

**7.63.2.2 template<class T1, class T2> T2 Oscl\_Pair< T1, T2 >::second**

Referenced by Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::count(), Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::erase(), and Oscl\_TagTree< PVLogger \*, alloc\_type >::insert().

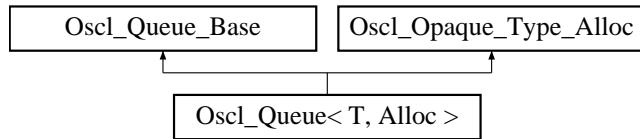
The documentation for this struct was generated from the following file:

- [oscl\\_tree.h](#)

## 7.64 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 Member Functions

- `Oscl_Queue ()`
- `Oscl_Queue (uint32 n)`
- `virtual ~Oscl_Queue ()`
- `void push (const T &x)`
- `reference front ()`
- `const_reference front () const`
- `void pop ()`
- `reference back ()`
- `const_reference back () const`
- `void clear ()`

#### 7.64.1 Detailed Description

`template<class T, class Alloc> class Oscl_Queue< T, Alloc >`

**Oscl\_Queue** Class. A subset of STL::Queue methods. **Oscl\_Queue** supports constant time insertion (at the end) and removal of elements at the front of the queue. It does not support insertion or removal of elements at the other ends or middle of the queue, nor random access to elements. \* No iteration capability is [currently] supplied. \* No assignment or copy capability is [currently] supplied. The number of elements in a queue can vary dynamically, and memory management is performed automatically.

### 7.64.2 Member Typedef Documentation

- 7.64.2.1 `template<class T, class Alloc> typedef const T& Oscl_Queue< T, Alloc >::const_reference`
- 7.64.2.2 `template<class T, class Alloc> typedef T* Oscl_Queue< T, Alloc >::pointer`
- 7.64.2.3 `template<class T, class Alloc> typedef T& Oscl_Queue< T, Alloc >::reference`
- 7.64.2.4 `template<class T, class Alloc> typedef uint32 Oscl_Queue< T, Alloc >::size_type`
- 7.64.2.5 `template<class T, class Alloc> typedef T Oscl_Queue< T, Alloc >::value_type`

### 7.64.3 Constructor & Destructor Documentation

- 7.64.3.1 `template<class T, class Alloc> Oscl_Queue< T, Alloc >::Oscl_Queue () [inline]`

Creates an empty queue.

References `Oscl_Queue_Base::construct()`, and `Oscl_Queue_Base::sizeof_T`.

- 7.64.3.2 `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.

References `Oscl_Queue_Base::construct()`, and `Oscl_Queue_Base::sizeof_T`.

- 7.64.3.3 `template<class T, class Alloc> virtual Oscl_Queue< T, Alloc >::~Oscl_Queue () [inline, virtual]`

The destructor.

References `Oscl_Queue_Base::destroy()`.

### 7.64.4 Member Function Documentation

- 7.64.4.1 `template<class T, class Alloc> const_reference Oscl_Queue< T, Alloc >::back () const [inline]`

Returns the last element (const)

References `Oscl_Queue_Base::elems`, `Oscl_Queue_Base::empty()`, `Oscl_Queue_Base::irear`, and `OSCL_ASSERT`.

**7.64.4.2 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)

References Oscl\_Queue\_Base::elems, Oscl\_Queue\_Base::empty(), Oscl\_Queue\_Base::irear, and OSCL\_ASSERT.

**7.64.4.3 template<class T, class Alloc> void Oscl\_Queue< T, Alloc >::clear () [inline]**

Removes all elements.

Reimplemented from [Oscl\\_Queue\\_Base](#).

**7.64.4.4 template<class T, class Alloc> const\_reference Oscl\_Queue< T, Alloc >::front () const [inline]**

Returns the first element (const)

References Oscl\_Queue\_Base::elems, Oscl\_Queue\_Base::empty(), Oscl\_Queue\_Base::ifront, and OSCL\_ASSERT.

**7.64.4.5 template<class T, class Alloc> reference Oscl\_Queue< T, Alloc >::front () [inline]**

Returns the first element.

Reimplemented from [Oscl\\_Queue\\_Base](#).

References Oscl\_Queue\_Base::elems, Oscl\_Queue\_Base::empty(), Oscl\_Queue\_Base::ifront, and OSCL\_ASSERT.

**7.64.4.6 template<class T, class Alloc> void Oscl\_Queue< T, Alloc >::pop () [inline]**

Removes the first element

Reimplemented from [Oscl\\_Queue\\_Base](#).

**7.64.4.7 template<class T, class Alloc> void Oscl\_Queue< T, Alloc >::push (const T & x) [inline]**

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL\_LEAVE will occur

**Parameters**

*x* new element

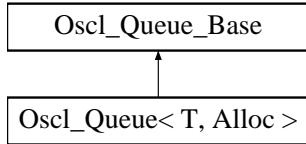
The documentation for this class was generated from the following file:

- [oscl\\_queue.h](#)

## 7.65 Oscl\_Queue\_Base Class Reference

```
#include <oscl_queue.h>
```

Inheritance diagram for Oscl\_Queue\_Base:



### Public Member Functions

- uint32 [size \(\) const](#)
- uint32 [capacity \(\) const](#)
- bool [empty \(\) const](#)
- OSCL\_IMPORT\_REF void [reserve](#) (uint32 n)

### Protected Member Functions

- OSCL\_IMPORT\_REF void [construct](#) (Oscl\_Opaque\_Type\_Alloc \*aType)
- OSCL\_IMPORT\_REF void [construct](#) (Oscl\_Opaque\_Type\_Alloc \*aType, uint32 n)
- virtual ~[Oscl\\_Queue\\_Base](#) ()
- OSCL\_IMPORT\_REF void [destroy](#) ()
- OSCL\_IMPORT\_REF void [push](#) (const OsclAny \*x)
- OSCL\_IMPORT\_REF void [pop](#) ()
- OSCL\_IMPORT\_REF void [clear](#) ()

### Protected Attributes

- uint32 [numelems](#)
- uint32 [bufsize](#)
- OsclAny \* [elems](#)
- uint32 [sizeof\\_T](#)
- uint32 [ifront](#)
- uint32 [irear](#)

#### 7.65.1 Detailed Description

[Oscl\\_Queue\\_Base](#) is a non-templatized base class for [Oscl\\_Queue](#). The purpose of this base class is to avoid large inline routines in the [Oscl\\_Queue](#) implementation. This class is not intended for direct instantiation except by [Oscl\\_Queue](#).

#### 7.65.2 Constructor & Destructor Documentation

##### 7.65.2.1 virtual Oscl\_Queue\_Base::~Oscl\_Queue\_Base () [inline, protected, virtual]

The destructor.

### 7.65.3 Member Function Documentation

#### 7.65.3.1 **uint32 Oscl\_Queue\_Base::capacity () const [inline]**

Returns the allocated memory of the queue.

References bufsize.

#### 7.65.3.2 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::clear () [protected]**

Removes all elements.

Reimplemented in [Oscl\\_Queue< T, Alloc >](#).

#### 7.65.3.3 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType, uint32 n) [protected]**

#### 7.65.3.4 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::construct (Oscl\_Opaque\_Type\_Alloc \* aType) [protected]**

Referenced by [Oscl\\_Queue< T, Alloc >::Oscl\\_Queue\(\)](#).

#### 7.65.3.5 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::destroy () [protected]**

Like an explicit destructor call.

Referenced by [Oscl\\_Queue< T, Alloc >::~Oscl\\_Queue\(\)](#).

#### 7.65.3.6 **bool Oscl\_Queue\_Base::empty () const [inline]**

True if there are no elements in the queue

References numelems.

Referenced by [Oscl\\_Queue< T, Alloc >::back\(\)](#), and [Oscl\\_Queue< T, Alloc >::front\(\)](#).

#### 7.65.3.7 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::pop () [protected]**

Removes the first element

Reimplemented in [Oscl\\_Queue< T, Alloc >](#).

#### 7.65.3.8 **OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::push (const OsclAny \* x) [protected]**

Inserts a new element at the end. Queue will be grown if necessary. If allocation fails, an OSCL\_LEAVE will occur

##### Parameters

*x* new element

**7.65.3.9 OSCL\_IMPORT\_REF void Oscl\_Queue\_Base::reserve (uint32 *n*)**

Reallocates memory if necessary to a capacity of *n* elements. The main reason for reserve is efficiency. If you know the capacity to which your vector must grow, then it is more efficient to allocate the vector all at once rather than rely on the automatic reallocation scheme. This also helps control the invalidation of iterators.

**Parameters**

*n* size of vector

**7.65.3.10 uint32 Oscl\_Queue\_Base::size () const [inline]**

Returns the size of the queue.

References numelems.

**7.65.4 Field Documentation****7.65.4.1 uint32 Oscl\_Queue\_Base::bufsize [protected]**

Referenced by capacity().

**7.65.4.2 OsclAny\* Oscl\_Queue\_Base::elems [protected]**

Referenced by Oscl\_Queue< T, Alloc >::back(), and Oscl\_Queue< T, Alloc >::front().

**7.65.4.3 uint32 Oscl\_Queue\_Base::ifront [protected]**

Referenced by Oscl\_Queue< T, Alloc >::front().

**7.65.4.4 uint32 Oscl\_Queue\_Base::irear [protected]**

Referenced by Oscl\_Queue< T, Alloc >::back().

**7.65.4.5 uint32 Oscl\_Queue\_Base::numelems [protected]**

Referenced by empty(), and size().

**7.65.4.6 uint32 Oscl\_Queue\_Base::sizeof\_T [protected]**

Referenced by Oscl\_Queue< T, Alloc >::Oscl\_Queue().

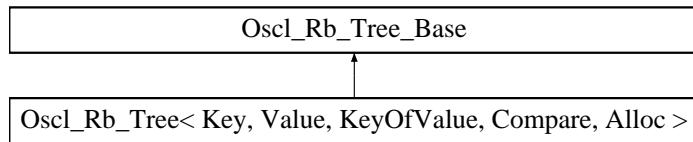
The documentation for this class was generated from the following file:

- [oscl\\_queue.h](#)

## 7.66 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 Member Functions

- `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)`

- void `clear ()`
  - `iterator find (const Key &k)`
  - `const_iterator find (const Key &k) const`
  - `size_type count (const Key &k) const`
  - `iterator lower_bound (const Key &k)`
  - `const_iterator lower_bound (const Key &k) const`
  - `iterator upper_bound (const Key &k)`
  - `const_iterator upper_bound (const Key &k) const`
  - `Oscl_Pair< iterator, iterator > equal_range (const Key &k)`
  - `Oscl_Pair< const_iterator, const_iterator > equal_range (const Key &k) const`

```
template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> class Oscl_Rb_-Tree< Key, Value, KeyOfValue, Compare, Alloc >
```

### 7.66.1 Member Typedef Documentation

- 7.66.1.1 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Oscl\_Rb\_Tree\_Const\_Iterator<value\_type> Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const\_iterator
- 7.66.1.2 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef const value\_type\* Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const\_pointer
- 7.66.1.3 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef const value\_type& Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::const\_reference
- 7.66.1.4 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef int32 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::difference\_type
- 7.66.1.5 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Oscl\_Rb\_Tree\_Iterator<value\_type> Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::iterator
- 7.66.1.6 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Key Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::key\_type
- 7.66.1.7 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Oscl\_Rb\_Tree\_Node<Value>::link\_type Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::link\_type
- 7.66.1.8 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef value\_type\* Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::pointer
- 7.66.1.9 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef value\_type& Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::reference
- 7.66.1.10 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef uint32 Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::size\_type
- 7.66.1.11 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> typedef Value Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::value\_type

### 7.66.2 Constructor & Destructor Documentation

- 7.66.2.1 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::Oscl\_Rb\_Tree (const Compare & *comp* = Compare()) [inline]
- 7.66.2.2 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]

- OSCL API
- 7.66.2.3 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::Oscl\_Rb\_Tree() [inline]

### 7.66.3 Member Function Documentation

- 7.66.3.1 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> const\_iterator Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::begin() const

`>::erase(), and Oscl_Rb_Tree< key_type, value_type, Oscl_Select1st< value_type, key_type >, key_compare, alloc_type >::insert_unique().`

**7.66.3.3 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> void Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::clear () [inline]**

Referenced by Oscl\_Map< const tag\_base\_type, node\_ptr, alloc\_type, Oscl\_Tag\_Base >::clear(), Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::erase(), Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::operator=(), and Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::~Oscl\_Rb\_Tree().

**7.66.3.4 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> size\_type Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::count (const Key & k) const [inline]**

Referenced by Oscl\_Map< const tag\_base\_type, node\_ptr, alloc\_type, Oscl\_Tag\_Base >::count().

**7.66.3.5 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> bool Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::empty () const [inline]**

Referenced by Oscl\_Map< const tag\_base\_type, node\_ptr, alloc\_type, Oscl\_Tag\_Base >::empty().

**7.66.3.6 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> const\_iterator Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::end () const [inline]**

**7.66.3.7 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> iterator Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::end () [inline]**

Referenced by Oscl\_Map< const tag\_base\_type, node\_ptr, alloc\_type, Oscl\_Tag\_Base >::end(), Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::erase(), and Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::find().

**7.66.3.8 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl\_Pair<const\_iterator, const\_iterator> Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::equal\_range (const Key & k) const [inline]**

**7.66.3.9 template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl\_Pair<iterator, iterator> Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::equal\_range (const Key & k) [inline]**

Referenced by Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::count(), Oscl\_Map< const tag\_base\_type, node\_ptr, alloc\_type, Oscl\_Tag\_Base >::equal\_range(), and Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::erase().

- 7.66.3.10 **template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> void Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::erase (const key\_type \* *first*, const key\_type \* *last*) [inline]**
- 7.66.3.11 **template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> void Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::erase (iterator *first*, iterator *last*) [inline]**
- 7.66.3.12 **template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> size\_type Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::erase (const key\_type & *x*) [inline]**
- 7.66.3.13 **template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> void Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::erase (iterator *position*) [inline]**

Referenced by Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::erase(), and Oscl\_Map< const tag\_base\_type, node\_ptr, alloc\_type, Oscl\_Tag\_Base >::erase().

- 7.66.3.14 **template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> const\_iterator Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::find (const Key & *k*) const [inline]**
- 7.66.3.15 **template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> iterator Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::find (const Key & *k*) [inline]**

Referenced by Oscl\_Map< const tag\_base\_type, node\_ptr, alloc\_type, Oscl\_Tag\_Base >::find().

- 7.66.3.16 **template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> void Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::insert\_unique (const value\_type \* *first*, const value\_type \* *last*) [inline]**
- 7.66.3.17 **template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> void Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::insert\_unique (const\_iterator *first*, const\_iterator *last*) [inline]**
- 7.66.3.18 **template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> iterator Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::insert\_unique (iterator *position*, const value\_type & *v*) [inline]**
- 7.66.3.19 **template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> Oscl\_Pair<iterator, bool> Oscl\_Rb\_Tree< Key, Value, KeyOfValue, Compare, Alloc >::insert\_unique (const value\_type & *v*) [inline]**

Referenced by Oscl\_Map< const tag\_base\_type, node\_ptr, alloc\_type, Oscl\_Tag\_Base >::insert(), and Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::insert\_unique().

- 7.66.3.20** `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc>`  
`const_iterator Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc`  
`>::lower_bound (const Key & k) const [inline]`
- 7.66.3.21** `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> iterator`  
`Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::lower_bound (const Key &`  
`k) [inline]`

Referenced by `Oscl_Rb_Tree< key_type, value_type, Oscl_Select1st< value_type, key_type >, key_compare, alloc_type >::equal_range()`, and `Oscl_Map< const tag_base_type, node_ptr, alloc_type, Oscl_Tag_Base >::lower_bound()`.

- 7.66.3.22** `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc>`  
`size_type Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::max_size () const`  
`[inline]`

Referenced by `Oscl_Map< const tag_base_type, node_ptr, alloc_type, Oscl_Tag_Base >::max_size()`.

- 7.66.3.23** `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc>`  
`Oscl_Rb_Tree<Key, Value, KeyOfValue, Compare, Alloc>& Oscl_Rb_Tree< Key,`  
`Value, KeyOfValue, Compare, Alloc >::operator= (const Oscl_Rb_Tree< Key, Value,`  
`KeyOfValue, Compare, Alloc > & x) [inline]`

- 7.66.3.24** `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc>`  
`size_type Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::size () const`  
`[inline]`

Referenced by `Oscl_Rb_Tree< key_type, value_type, Oscl_Select1st< value_type, key_type >, key_compare, alloc_type >::insert_unique()`, and `Oscl_Map< const tag_base_type, node_ptr, alloc_type, Oscl_Tag_Base >::size()`.

- 7.66.3.25** `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc>`  
`const_iterator Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc`  
`>::upper_bound (const Key & k) const [inline]`

- 7.66.3.26** `template<class Key, class Value, class KeyOfValue, class Compare, class Alloc> iterator`  
`Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >::upper_bound (const Key`  
`& k) [inline]`

Referenced by `Oscl_Rb_Tree< key_type, value_type, Oscl_Select1st< value_type, key_type >, key_compare, alloc_type >::equal_range()`, and `Oscl_Map< const tag_base_type, node_ptr, alloc_type, Oscl_Tag_Base >::upper_bound()`.

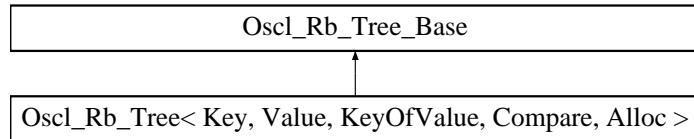
The documentation for this class was generated from the following file:

- [oscl\\_tree.h](#)

## 7.67 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 Member Functions

- `OSCL_IMPORT_REF void rotate_left (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF void rotate_right (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF void rebalance (base_link_type x, base_link_type &root)`
- `OSCL_IMPORT_REF base_link_type rebalance_for_erase (base_link_type z, base_link_type &root, base_link_type &leftmost, base_link_type &rightmost)`

#### 7.67.1 Member Typedef Documentation

##### 7.67.1.1 `typedef Oscl_Rb_Tree_Node_Base::base_link_type Oscl_Rb_Tree_Base::base_link_type`

#### 7.67.2 Member Function Documentation

##### 7.67.2.1 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rebalance (base_link_type x, base_link_type & root)`

##### 7.67.2.2 `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)`

Referenced by `Oscl_Rb_Tree< key_type, value_type, Oscl_Select1st< value_type, key_type >, key_compare, alloc_type >::erase()`.

##### 7.67.2.3 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rotate_left (base_link_type x, base_link_type & root)`

##### 7.67.2.4 `OSCL_IMPORT_REF void Oscl_Rb_Tree_Base::rotate_right (base_link_type x, base_link_type & root)`

The documentation for this class was generated from the following file:

- `oscl_tree.h`

## 7.68 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 Member Functions

- `Oscl_Rb_Tree_Const_Iterator()`
- `Oscl_Rb_Tree_Const_Iterator(link_type x)`
- `Oscl_Rb_Tree_Const_Iterator(const const_iterator &it)`
- `reference operator*() const`
- `pointer operator->() const`
- `bool operator==(const self &x)`
- `bool operator!=(const self &x)`
- `self & operator++()`
- `self operator++(int)`
- `self & operator--()`
- `self operator--(int)`

### Data Fields

- `base_link_type node`

---

```
template<class Value> struct Oscl_Rb_Tree_Const_Iterator< Value >
```

### 7.68.1 Member Typedef Documentation

- 7.68.1.1 template<class Value > typedef Oscl\_Rb\_Tree\_Node\_Base\* Oscl\_Rb\_Tree\_Const\_Iterator< Value >::base\_link\_type
- 7.68.1.2 template<class Value > typedef Oscl\_Rb\_Tree\_Const\_Iterator<Value> Oscl\_Rb\_Tree\_Const\_Iterator< Value >::const\_iterator
- 7.68.1.3 template<class Value > typedef Oscl\_Rb\_Tree\_Node<Value>\* Oscl\_Rb\_Tree\_Const\_Iterator< Value >::link\_type
- 7.68.1.4 template<class Value > typedef const value\_type\* Oscl\_Rb\_Tree\_Const\_Iterator< Value >::pointer
- 7.68.1.5 template<class Value > typedef const value\_type& Oscl\_Rb\_Tree\_Const\_Iterator< Value >::reference
- 7.68.1.6 template<class Value > typedef Oscl\_Rb\_Tree\_Const\_Iterator<Value> Oscl\_Rb\_Tree\_Const\_Iterator< Value >::self
- 7.68.1.7 template<class Value > typedef Value Oscl\_Rb\_Tree\_Const\_Iterator< Value >::value\_type

### 7.68.2 Constructor & Destructor Documentation

- 7.68.2.1 template<class Value > Oscl\_Rb\_Tree\_Const\_Iterator< Value >::Oscl\_Rb\_Tree\_Const\_Iterator () [inline]
- 7.68.2.2 template<class Value > Oscl\_Rb\_Tree\_Const\_Iterator< Value >::Oscl\_Rb\_Tree\_Const\_Iterator (link\_type x) [inline]

References Oscl\_Rb\_Tree\_Const\_Iterator< Value >::node.

- 7.68.2.3 template<class Value > Oscl\_Rb\_Tree\_Const\_Iterator< Value >::Oscl\_Rb\_Tree\_Const\_Iterator (const const\_iterator & it) [inline]

References Oscl\_Rb\_Tree\_Const\_Iterator< Value >::node.

### 7.68.3 Member Function Documentation

- 7.68.3.1 template<class Value > bool Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator!= (const self & x) [inline]

References Oscl\_Rb\_Tree\_Const\_Iterator< Value >::node.

- 7.68.3.2 template<class Value > reference Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator\* () const [inline]

References Oscl\_Rb\_Tree\_Const\_Iterator< Value >::node.

Referenced by Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator->().

**7.68.3.3 template<class Value > self Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator++ (int) [inline]**

**7.68.3.4 template<class Value > self& Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator++ () [inline]**

References Oscl\_Rb\_Tree\_Node\_Base::left, Oscl\_Rb\_Tree\_Const\_Iterator< Value >::node, Oscl\_Rb\_Tree\_Node\_Base::parent, and Oscl\_Rb\_Tree\_Node\_Base::right.

**7.68.3.5 template<class Value > self Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator-- (int) [inline]**

**7.68.3.6 template<class Value > self& Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator-- () [inline]**

References Oscl\_Rb\_Tree\_Node\_Base::color, Oscl\_Rb\_Tree\_Node\_Base::left, Oscl\_Rb\_Tree\_Const\_Iterator< Value >::node, Oscl\_Rb\_Tree\_Node\_Base::parent, Oscl\_Rb\_Tree\_Node\_Base::red, and Oscl\_Rb\_Tree\_Node\_Base::right.

**7.68.3.7 template<class Value > pointer Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator-> () const [inline]**

References Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator\*().

**7.68.3.8 template<class Value > bool Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator==(const self & x) [inline]**

References Oscl\_Rb\_Tree\_Const\_Iterator< Value >::node.

## 7.68.4 Field Documentation

**7.68.4.1 template<class Value > base\_link\_type Oscl\_Rb\_Tree\_Const\_Iterator< Value >::node**

Referenced by Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::find(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator!=(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator\*(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator++(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator--(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator==( ), and Oscl\_Rb\_Tree\_Const\_Iterator< Value >::Oscl\_Rb\_Tree\_Const\_Iterator().

The documentation for this struct was generated from the following file:

- [oscl\\_tree.h](#)

## 7.69 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 Member Functions

- `Oscl_Rb_Tree_Iterator ()`
- `Oscl_Rb_Tree_Iterator (link_type x)`
- `Oscl_Rb_Tree_Iterator (const iterator &it)`
- `reference operator* () const`
- `pointer operator-> () const`
- `bool operator== (const self &x)`
- `bool operator!= (const self &x)`
- `self & operator++ ()`
- `self operator++ (int)`
- `self & operator-- ()`
- `self operator-- (int)`

### Data Fields

- `base_link_type node`

---

```
template<class Value> struct Oscl_Rb_Tree_Iterator< Value >
```

### 7.69.1 Member Typedef Documentation

- 7.69.1.1 template<class Value > typedef Oscl\_Rb\_Tree\_Node\_Base\* Oscl\_Rb\_Tree\_Iterator< Value >::base\_link\_type
- 7.69.1.2 template<class Value > typedef Oscl\_Rb\_Tree\_Iterator<Value> Oscl\_Rb\_Tree\_Iterator< Value >::iterator
- 7.69.1.3 template<class Value > typedef Oscl\_Rb\_Tree\_Node<Value>\* Oscl\_Rb\_Tree\_Iterator< Value >::link\_type
- 7.69.1.4 template<class Value > typedef value\_type\* Oscl\_Rb\_Tree\_Iterator< Value >::pointer
- 7.69.1.5 template<class Value > typedef value\_type& Oscl\_Rb\_Tree\_Iterator< Value >::reference
- 7.69.1.6 template<class Value > typedef Oscl\_Rb\_Tree\_Iterator<Value> Oscl\_Rb\_Tree\_Iterator< Value >::self
- 7.69.1.7 template<class Value > typedef Value Oscl\_Rb\_Tree\_Iterator< Value >::value\_type

### 7.69.2 Constructor & Destructor Documentation

- 7.69.2.1 template<class Value > Oscl\_Rb\_Tree\_Iterator< Value >::Oscl\_Rb\_Tree\_Iterator () [inline]
- 7.69.2.2 template<class Value > Oscl\_Rb\_Tree\_Iterator< Value >::Oscl\_Rb\_Tree\_Iterator (link\_type *x*) [inline]

References Oscl\_Rb\_Tree\_Iterator< Value >::node.

- 7.69.2.3 template<class Value > Oscl\_Rb\_Tree\_Iterator< Value >::Oscl\_Rb\_Tree\_Iterator (const iterator & *it*) [inline]

References Oscl\_Rb\_Tree\_Iterator< Value >::node.

### 7.69.3 Member Function Documentation

- 7.69.3.1 template<class Value > bool Oscl\_Rb\_Tree\_Iterator< Value >::operator!= (const self & *x*) [inline]

References Oscl\_Rb\_Tree\_Iterator< Value >::node.

- 7.69.3.2 template<class Value > reference Oscl\_Rb\_Tree\_Iterator< Value >::operator\* () const [inline]

References Oscl\_Rb\_Tree\_Iterator< Value >::node.

Referenced by Oscl\_Rb\_Tree\_Iterator< Value >::operator->().

**7.69.3.3 template<class Value > self Oscl\_Rb\_Tree\_Iterator< Value >::operator++ (int) [inline]**

**7.69.3.4 template<class Value > self& Oscl\_Rb\_Tree\_Iterator< Value >::operator++ () [inline]**

References Oscl\_Rb\_Tree\_Node\_Base::left, Oscl\_Rb\_Tree\_Iterator< Value >::node, Oscl\_Rb\_Tree\_Node\_Base::parent, and Oscl\_Rb\_Tree\_Node\_Base::right.

**7.69.3.5 template<class Value > self Oscl\_Rb\_Tree\_Iterator< Value >::operator-- (int) [inline]**

**7.69.3.6 template<class Value > self& Oscl\_Rb\_Tree\_Iterator< Value >::operator-- () [inline]**

References Oscl\_Rb\_Tree\_Node\_Base::color, Oscl\_Rb\_Tree\_Node\_Base::left, Oscl\_Rb\_Tree\_Iterator< Value >::node, Oscl\_Rb\_Tree\_Node\_Base::parent, Oscl\_Rb\_Tree\_Node\_Base::red, and Oscl\_Rb\_Tree\_Node\_Base::right.

**7.69.3.7 template<class Value > pointer Oscl\_Rb\_Tree\_Iterator< Value >::operator-> () const [inline]**

References Oscl\_Rb\_Tree\_Iterator< Value >::operator\*().

**7.69.3.8 template<class Value > bool Oscl\_Rb\_Tree\_Iterator< Value >::operator==(const self & x) [inline]**

References Oscl\_Rb\_Tree\_Iterator< Value >::node.

## 7.69.4 Field Documentation

**7.69.4.1 template<class Value > base\_link\_type Oscl\_Rb\_Tree\_Iterator< Value >::node**

Referenced by Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::erase(), Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::find(), Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::insert\_unique(), Oscl\_Rb\_Tree\_Iterator< Value >::operator!=(), Oscl\_Rb\_Tree\_Iterator< Value >::operator\*(), Oscl\_Rb\_Tree\_Iterator< Value >::operator++(), Oscl\_Rb\_Tree\_Iterator< Value >::operator--(), Oscl\_Rb\_Tree\_Iterator< Value >::operator==(), and Oscl\_Rb\_Tree\_Iterator< Value >::Oscl\_Rb\_Tree\_Iterator().

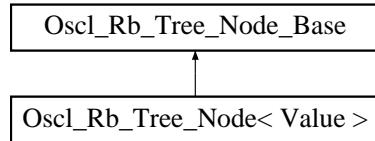
The documentation for this struct was generated from the following file:

- [oscl\\_tree.h](#)

## 7.70 Oscl\_Rb\_Tree\_Node< Value > Struct Template Reference

#include <oscl\_tree.h>

Inheritance diagram for Oscl\_Rb\_Tree\_Node< Value >:



### Public Types

- typedef Value [value\\_type](#)
- typedef [Oscl\\_Rb\\_Tree\\_Node< Value > \\*](#) [link\\_type](#)

### Data Fields

- [value\\_type value](#)

**template<class Value> struct Oscl\_Rb\_Tree\_Node< Value >**

#### 7.70.1 Member Typedef Documentation

**7.70.1.1 template<class Value> typedef Oscl\_Rb\_Tree\_Node<Value>\*> Oscl\_Rb\_Tree\_Node< Value >::link\_type**

**7.70.1.2 template<class Value> typedef Value Oscl\_Rb\_Tree\_Node< Value >::value\_type**

#### 7.70.2 Field Documentation

**7.70.2.1 template<class Value> value\_type Oscl\_Rb\_Tree\_Node< Value >::value**

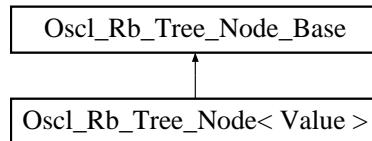
The documentation for this struct was generated from the following file:

- [oscl\\_tree.h](#)

## 7.71 Oscl\_Rb\_Tree\_Node\_Base Struct Reference

```
#include <oscl_tree.h>
```

Inheritance diagram for Oscl\_Rb\_Tree\_Node\_Base:



### Public Types

- enum RedBl { red, black }
- typedef Oscl\_Rb\_Tree\_Node\_Base \* base\_link\_type
- typedef enum Oscl\_Rb\_Tree\_Node\_Base::RedBl color\_type

### Static Public Member Functions

- static base\_link\_type minimum (base\_link\_type x)
- static base\_link\_type maximum (base\_link\_type x)

### Data Fields

- color\_type color
- base\_link\_type parent
- base\_link\_type left
- base\_link\_type right

#### 7.71.1 Member Typedef Documentation

7.71.1.1 `typedef Oscl_Rb_Tree_Node_Base* Oscl_Rb_Tree_Node_Base::base_link_type`

7.71.1.2 `typedef enum Oscl_Rb_Tree_Node_Base::RedBl Oscl_Rb_Tree_Node_Base::color_type`

#### 7.71.2 Member Enumeration Documentation

7.71.2.1 `enum Oscl_Rb_Tree_Node_Base::RedBl`

Enumerator:

*red*

*black*

### 7.71.3 Member Function Documentation

**7.71.3.1 static base\_link\_type Oscl\_Rb\_Tree\_Node\_Base::maximum (base\_link\_type x)**  
**[inline, static]**

References right.

**7.71.3.2 static base\_link\_type Oscl\_Rb\_Tree\_Node\_Base::minimum (base\_link\_type x)**  
**[inline, static]**

References left.

### 7.71.4 Field Documentation

**7.71.4.1 color\_type Oscl\_Rb\_Tree\_Node\_Base::color**

Referenced by Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator--(), Oscl\_Rb\_Tree\_Iterator< Value >::operator--(), and Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::Oscl\_Rb\_Tree().

**7.71.4.2 base\_link\_type Oscl\_Rb\_Tree\_Node\_Base::left**

Referenced by Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::erase(), Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::insert\_unique(), minimum(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator++(), Oscl\_Rb\_Tree\_Iterator< Value >::operator++(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator--(), and Oscl\_Rb\_Tree\_Iterator< Value >::operator--().

**7.71.4.3 base\_link\_type Oscl\_Rb\_Tree\_Node\_Base::parent**

Referenced by Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::erase(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator++(), Oscl\_Rb\_Tree\_Iterator< Value >::operator++(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator--(), and Oscl\_Rb\_Tree\_Iterator< Value >::operator--().

**7.71.4.4 base\_link\_type Oscl\_Rb\_Tree\_Node\_Base::right**

Referenced by Oscl\_Rb\_Tree< key\_type, value\_type, Oscl\_Select1st< value\_type, key\_type >, key\_compare, alloc\_type >::erase(), maximum(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator++(), Oscl\_Rb\_Tree\_Iterator< Value >::operator++(), Oscl\_Rb\_Tree\_Const\_Iterator< Value >::operator--(), and Oscl\_Rb\_Tree\_Iterator< Value >::operator--().

The documentation for this struct was generated from the following file:

- [oscl\\_tree.h](#)

## 7.72 Oscl\_Select1st< V, U > Struct Template Reference

```
#include <oscl_map.h>
```

### Public Member Functions

- const U & **operator()** (const V &x) const

```
template<class V, class U> struct Oscl_Select1st< V, U >
```

#### 7.72.1 Member Function Documentation

```
7.72.1.1 template<class V , class U > const U& Oscl_Select1st< V, U >::operator() (const V & x)
      const [inline]
```

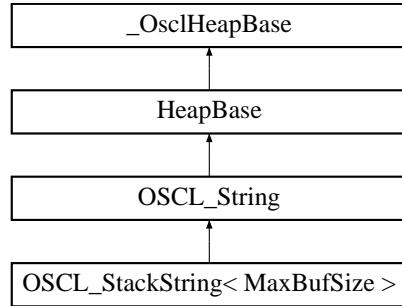
The documentation for this struct was generated from the following file:

- [oscl\\_map.h](#)

## 7.73 OSCL\_StackString< MaxBufSize > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_StackString< MaxBufSize >:



### Public Types

- [typedef OSCL\\_String::chartype chartype](#)
- [typedef TOSCL\\_StringOp optype](#)
- [typedef OSCL\\_wString::chartype other\\_chartype](#)

### Public Member Functions

- [OSCL\\_StackString \(\)](#)
- [OSCL\\_StackString \(const OSCL\\_StackString &src\)](#)
- [OSCL\\_StackString \(const OSCL\\_String &src\)](#)
- [OSCL\\_StackString \(const chartype \\*cstr\)](#)
- [OSCL\\_StackString \(const chartype \\*buf, uint32 length\)](#)
- [~OSCL\\_StackString \(\)](#)
- [uint32 get\\_size \(\) const](#)
- [uint32 get\\_maxsize \(\) const](#)
- [const chartype \\* get\\_cstr \(\) const](#)
- [chartype \\* get\\_str \(\) const](#)
- [OSCL\\_StackString & operator= \(const OSCL\\_StackString &src\)](#)
- [OSCL\\_StackString & operator= \(const OSCL\\_String &src\)](#)
- [OSCL\\_StackString & operator= \(const chartype \\*cstr\)](#)
- [void set \(const chartype \\*buf, uint32 length\)](#)
- [void set \(const other\\_chartype \\*buf, optype op\)](#)
- [void set \(const other\\_chartype \\*buf, uint32 length, optype op\)](#)

### Friends

- [class OSCL\\_String](#)

### 7.73.1 Detailed Description

**template<uint32 MaxBufSize> class OSCL\_StackString< MaxBufSize >**

[OSCL\\_StackString](#) is a simple string class, compatible with regular character array strings.

The string array is fixed length, is allocated from the stack, and is modifiable. Operations that update the string will automatically truncate it to fit the fixed size storage. This is recommended for use for short strings (<255). Use [OSCL\\_HeapString](#) for very large strings to avoid stack overflow.

#### Parameters

*C*,: type of character.

*MaxBufSize*,: maximum string length not including null terminator.

### 7.73.2 Member Typedef Documentation

**7.73.2.1 template<uint32 MaxBufSize> typedef OSCL\_String::chartype OSCL\_StackString< MaxBufSize >::chartype**

Reimplemented from [OSCL\\_String](#).

**7.73.2.2 template<uint32 MaxBufSize> typedef TOSCL\_StringOp OSCL\_StackString< MaxBufSize >::optype**

**7.73.2.3 template<uint32 MaxBufSize> typedef OSCL\_wString::chartype OSCL\_StackString< MaxBufSize >::other\_chartype**

### 7.73.3 Friends And Related Function Documentation

**7.73.3.1 template<uint32 MaxBufSize> friend class OSCL\_String [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 7.74 oscl\_stat\_buf Struct Reference

```
#include <oscl_file_dir_utils.h>
```

### Data Fields

- uint32 [mode](#)
- uint32 [perms](#)

#### 7.74.1 Field Documentation

##### 7.74.1.1 uint32 oscl\_stat\_buf::mode

##### 7.74.1.2 uint32 oscl\_stat\_buf::perms

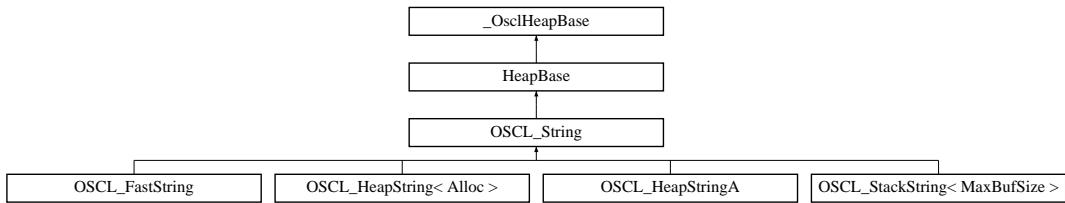
The documentation for this struct was generated from the following file:

- [oscl\\_file\\_dir\\_utils.h](#)

## 7.75 OSCL\_String Class Reference

```
#include <oscl_string.h>
```

Inheritance diagram for OSCL\_String:



### Public Types

- `typedef char chartype`

### Public Member Functions

- `virtual uint32 get_size () const =0`
- `virtual uint32 get_maxsize () const =0`
- `virtual const chartype * get_cstr () const =0`
- `virtual bool is_writable () const`
- `virtual chartype * get_str () const =0`
- `OSCL_String & operator=(const OSCL_String &src)`
- `OSCL_String & operator=(const chartype *cstr)`
- `OSCL_String & operator+=(const OSCL_String &src)`
- `OSCL_String & operator+=(const chartype *cstr)`
- `OSCL_String & operator+=(const chartype c)`
- `bool operator==(const OSCL_String &src) const`
- `bool operator!=(const OSCL_String &src) const`
- `bool operator<(const OSCL_String &src) const`
- `bool operator<=(const OSCL_String &src) const`
- `bool operator>(const OSCL_String &src) const`
- `bool operator>=(const OSCL_String &src) const`
- `bool operator==(const chartype *cstr) const`
- `chartype operator[ ](uint32 index) const`
- `virtual chartype read (uint32 index) const`
- `virtual uint32 setrep_to_char (const oscl_wchar *src, uint32 len, TOSCL_StringOp op, Oscl_DefAlloc *aAlloc)`
- `virtual int8 hash () const`
- `virtual void write (uint32 index, chartype c)`
- `virtual void write (uint32 offset, uint32 length, const chartype *buf)`

## Protected Member Functions

- `OSCL_String()`
- virtual `~OSCL_String()`
- virtual void `set_rep(const chartype *cstr)=0`
- virtual void `append_rep(const chartype *cstr)=0`
- virtual void `set_rep(const OSCL_String &src)=0`
- virtual void `append_rep(const OSCL_String &src)=0`
- virtual void `set_len(uint32 len)=0`

### 7.75.1 Detailed Description

A common base class for string classes with "char" character format

### 7.75.2 Member Typedef Documentation

#### 7.75.2.1 `typedef char OSCL_String::chartype`

Reimplemented in `OSCL_HeapString< Alloc >`, `OSCL_HeapStringA`, `OSCL_StackString< MaxBufSize >`, `OSCL_FastString`, and `OSCL_HeapString< OsclMemAllocator >`.

### 7.75.3 Constructor & Destructor Documentation

#### 7.75.3.1 `OSCL_String::OSCL_String()` [protected]

#### 7.75.3.2 `virtual OSCL_String::~OSCL_String()` [protected, virtual]

### 7.75.4 Member Function Documentation

#### 7.75.4.1 `virtual void OSCL_String::append_rep(const OSCL_String & src)` [protected, pure virtual]

Append the input string to the current string. The string may be truncated to fit the available storage.

#### 7.75.4.2 `virtual void OSCL_String::append_rep(const chartype * cstr)` [protected, pure virtual]

Append the input null-terminated string to the current string. The string may be truncated to fit the available storage.

#### 7.75.4.3 `virtual const chartype* OSCL_String::get_cstr() const` [pure virtual]

This function returns the C-style string for read access.

Implemented in `OSCL_HeapString< Alloc >`, `OSCL_HeapStringA`, `OSCL_StackString< MaxBufSize >`, `OSCL_FastString`, and `OSCL_HeapString< OsclMemAllocator >`.

**7.75.4.4 virtual uint32 OSCL\_String::get\_maxsize () const [pure virtual]**

This function returns the maximum available storage size, not including null terminator. The maximum size may be larger than the current string size.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

**7.75.4.5 virtual uint32 OSCL\_String::get\_size () const [pure virtual]**

This function returns the string size not including the null-terminator.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

**7.75.4.6 virtual chartype\* OSCL\_String::get\_str () const [pure virtual]**

This function returns the C-style string for write access. If the string is not writable it returns NULL.

Implemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

**7.75.4.7 virtual int8 OSCL\_String::hash () const [virtual]**

This function performs a hash operation on the string. If the string is not writable, the function leaves.

**7.75.4.8 virtual bool OSCL\_String::is\_writable () const [virtual]**

This function returns true if the string is writable.

**7.75.4.9 bool OSCL\_String::operator!= (const OSCL\_String & src) const****7.75.4.10 OSCL\_String& OSCL\_String::operator+= (const chartype c)**

Append operator. This operator appends the input character to this object. The string may be truncated to fit available storage.

**7.75.4.11 OSCL\_String& OSCL\_String::operator+= (const chartype \* cstr)**

Append operator. This operator appends the input string to this object. The string may be truncated to fit available storage.

**Parameters**

null-terminated string

**7.75.4.12 OSCL\_String& OSCL\_String::operator+= (const OSCL\_String & src)**

Append operator. This operator appends the input string to this object. The string may be truncated to fit available storage.

**7.75.4.13 bool OSCL\_String::operator< (const OSCL\_String & src) const**

**7.75.4.14 bool OSCL\_String::operator<= (const OSCL\_String & src) const**

**7.75.4.15 OSCL\_String& OSCL\_String::operator= (const chartype \* cstr)**

Assignment operator

#### Parameters

null-terminated string

Reimplemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), [OSCL\\_FastString](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

**7.75.4.16 OSCL\_String& OSCL\_String::operator= (const OSCL\_String & src)**

Assignment operator

Reimplemented in [OSCL\\_HeapString< Alloc >](#), [OSCL\\_HeapStringA](#), [OSCL\\_StackString< MaxBufSize >](#), and [OSCL\\_HeapString< OsclMemAllocator >](#).

**7.75.4.17 bool OSCL\_String::operator== (const chartype \* cstr) const**

Comparison operator

#### Parameters

null-terminated string

**7.75.4.18 bool OSCL\_String::operator== (const OSCL\_String & src) const**

Comparison operators

**7.75.4.19 bool OSCL\_String::operator> (const OSCL\_String & src) const**

**7.75.4.20 bool OSCL\_String::operator>= (const OSCL\_String & src) const**

**7.75.4.21 chartype OSCL\_String::operator[ ] (uint32 index) const**

This is subscript notation to access a character at the given position. If the index is outside the current size range then the function leaves.

**7.75.4.22 virtual chartype OSCL\_String::read (uint32 index) const [virtual]**

This function returns the character at the given position. If the index is outside the current size range then the function leaves.

**7.75.4.23 virtual void OSCL\_String::set\_len (uint32 len) [protected, pure virtual]**

Update the length of the string. This function will only be called when the string is writable.

**7.75.4.24 virtual void OSCL\_String::set\_rep (const OSCL\_String & src) [protected, pure virtual]**

Set string representation to input string.

**7.75.4.25 virtual void OSCL\_String::set\_rep (const chartype \* cstr) [protected, pure virtual]**

Each representation class must implement these pure virtuals. Set string representation to input null-terminated string.

**7.75.4.26 virtual uint32 OSCL\_String::setrep\_to\_char (const oscl\_wchar \* src, uint32 len, TOSCL\_StringOp op, Oscl\_DefAlloc \* aAlloc) [virtual]**

This function allocates a temp storage for performing one of the following operations based on TOSCL\_StringOp

- compress src string from oscl\_wchar to utf8.
- convert src string from oscl\_wchar to utf8. call parent [set\\_rep\(\)](#) to copy resulting string.

**Parameters**

*src*,: reference input string  
*len*,: length of string to operate on  
*op*,: type operation mentioned above  
*aAlloc*,: optional, memory allocator if available

**Returns**

length of compressed or converted string exclude terminated ”.

Referenced by OSCL\_StackString< MaxBufSize >::set(), and OSCL\_HeapString< Alloc >::set().

**7.75.4.27 virtual void OSCL\_String::write (uint32 offset, uint32 length, const chartype \* buf) [virtual]**

This function replaces characters at the specified offset within the current string. If the string is not writable, the function leaves. The characters may be truncated to fit the current storage.

**Parameters**

*offset*,: the offset into the existing string buffer  
*length*,: number of characters to copy.  
*ptr*,: character buffer, not necessarily null-terminated.

**7.75.4.28 virtual void OSCL\_String::write (uint32 index, chartype c) [virtual]**

This function stores a character at the specified position. If the string is not writable, the function leaves. If the index is outside the current size range then the function leaves.

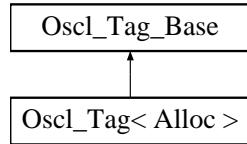
The documentation for this class was generated from the following file:

- [oscl\\_string.h](#)

## 7.76 Oscl\_Tag< Alloc > Struct Template Reference

```
#include <oscl_tagtree.h>
```

Inheritance diagram for Oscl\_Tag< Alloc >:



### Public Member Functions

- `Oscl_Tag (const Oscl_Tag< Alloc > &t)`
- `Oscl_Tag (const tag_base_type &t)`
- `~Oscl_Tag ()`
- `bool operator< (const Oscl_Tag< Alloc > &x) const`

### Data Fields

- `Oscl_TAlloc< tag_base_unit, Alloc > tagAllocator`
- `tag_base_type tag`

`template<class Alloc> struct Oscl_Tag< Alloc >`

#### 7.76.1 Constructor & Destructor Documentation

##### 7.76.1.1 `template<class Alloc> Oscl_Tag< Alloc >::Oscl_Tag (const Oscl_Tag< Alloc > & t) [inline]`

References `Oscl_Tag< Alloc >::tag`, `Oscl_Tag_Base::tag_copy()`, `Oscl_Tag_Base::tag_len()`, and `Oscl_Tag< Alloc >::tagAllocator`.

##### 7.76.1.2 `template<class Alloc> Oscl_Tag< Alloc >::Oscl_Tag (const tag_base_type & t) [inline]`

References `Oscl_Tag< Alloc >::tag`, `Oscl_Tag_Base::tag_copy()`, `Oscl_Tag_Base::tag_len()`, and `Oscl_Tag< Alloc >::tagAllocator`.

##### 7.76.1.3 `template<class Alloc> Oscl_Tag< Alloc >::~Oscl_Tag () [inline]`

References `Oscl_TAlloc< T, Alloc >::deallocate()`, `Oscl_Tag< Alloc >::tag`, and `Oscl_Tag< Alloc >::tagAllocator`.

## 7.76.2 Member Function Documentation

### 7.76.2.1 template<class Alloc> bool Oscl\_Tag< Alloc >::operator<(const Oscl\_Tag< Alloc > & x) const [inline]

References Oscl\_Tag< Alloc >::tag, and Oscl\_Tag\_Base::tag\_cmp().

## 7.76.3 Field Documentation

### 7.76.3.1 template<class Alloc> tag\_base\_type Oscl\_Tag< Alloc >::tag

Referenced by Oscl\_TagTree< T, Alloc >::Node::depth(), Oscl\_TagTree< PVLogger \*, alloc\_type >::insert(), Oscl\_Tag< Alloc >::operator<(), Oscl\_Tag< Alloc >::Oscl\_Tag(), and Oscl\_Tag< Alloc >::~Oscl\_Tag().

### 7.76.3.2 template<class Alloc> Oscl\_TAlloc<tag\_base\_unit, Alloc> Oscl\_Tag< Alloc >::tagAllocator

Referenced by Oscl\_Tag< Alloc >::Oscl\_Tag(), and Oscl\_Tag< Alloc >::~Oscl\_Tag().

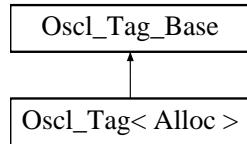
The documentation for this struct was generated from the following file:

- [oscl\\_tagtree.h](#)

## 7.77 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 Member Functions

- `bool operator() (const tag_base_type &x, const tag_base_type &y) const`
- `size_type tag_len (const tag_base_type &t) const`
- `tag_base_type tag_copy (tag_base_type &dest, const tag_base_type &src) const`
- `int32 tag_cmp (const tag_base_type &x, const tag_base_type &y) const`
- `OSCL_IMPORT_REF tag_base_type tag_ancestor (tag_base_type &dest, const tag_base_type &src) const`
- `OSCL_IMPORT_REF size_type tag_depth (const tag_base_type &t) const`

#### 7.77.1 Member Typedef Documentation

**7.77.1.1 `typedef uint32 Oscl_Tag_Base::size_type`**

**7.77.1.2 `typedef tag_base_unit* Oscl_Tag_Base::tag_base_type`**

**7.77.1.3 `typedef char Oscl_Tag_Base::tag_base_unit`**

#### 7.77.2 Member Function Documentation

**7.77.2.1 `bool Oscl_Tag_Base::operator() (const tag_base_type & x, const tag_base_type & y) const [inline]`**

References `tag_cmp()`.

**7.77.2.2 `OSCL_IMPORT_REF tag_base_type Oscl_Tag_Base::tag_ancestor (tag_base_type & dest, const tag_base_type & src) const`**

Referenced by `Oscl_TagTree< PVLogger *, alloc_type >::insert()`.

**7.77.2.3 int32 Oscl\_Tag\_Base::tag\_cmp (const tag\_base\_type & x, const tag\_base\_type & y) const [inline]**

References oscl\_strlen(), and oscl\_strncmp().

Referenced by operator()(), and Oscl\_Tag< Alloc >::operator<().

**7.77.2.4 tag\_base\_type Oscl\_Tag\_Base::tag\_copy (tag\_base\_type & dest, const tag\_base\_type & src) const [inline]**

References oscl\_strlen(), and oscl\_strncpy().

Referenced by Oscl\_Tag< Alloc >::Oscl\_Tag().

**7.77.2.5 OSCL\_IMPORT\_REF size\_type Oscl\_Tag\_Base::tag\_depth (const tag\_base\_type & t) const**

Referenced by Oscl\_TagTree< T, Alloc >::Node::depth(), and Oscl\_TagTree< PVLogger \*, alloc\_type >::insert().

**7.77.2.6 size\_type Oscl\_Tag\_Base::tag\_len (const tag\_base\_type & t) const [inline]**

References oscl\_strlen().

Referenced by Oscl\_Tag< Alloc >::Oscl\_Tag().

The documentation for this struct was generated from the following file:

- [oscl\\_tagtree.h](#)

## 7.78 Oscl\_TagTree< T, Alloc > Class Template Reference

```
#include <oscl_tagtree.h>
```

### Data Structures

- struct [const\\_iterator](#)
- struct [iterator](#)
- struct [Node](#)

### 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 Member Functions

- [Oscl\\_TagTree \(size\\_type max\\_depth=0\)](#)
- [Oscl\\_TagTree \(const Oscl\\_TagTree< T, Alloc > &x\)](#)
- [Oscl\\_TagTree< T, Alloc > & operator= \(const Oscl\\_TagTree< T, Alloc > &x\)](#)
- [~Oscl\\_TagTree \(\)](#)
- [iterator begin \(\)](#)
- [const\\_iterator begin \(\) const](#)
- [iterator end \(\)](#)
- [const\\_iterator end \(\) const](#)
- [bool empty \(\) const](#)
- [size\\_type size \(\) const](#)
- [T & operator\[\] \(const tag\\_base\\_type &t\)](#)
- [pair\\_iterator\\_bool insert \(const tag\\_base\\_type &t, const T &x\)](#)
- [void erase \(iterator position\)](#)
- [size\\_type erase \(const tag\\_base\\_type &x\)](#)
- [void clear \(\)](#)
- [iterator find \(const tag\\_base\\_type &x\)](#)
- [size\\_type count \(const tag\\_base\\_type &x\) const](#)

#### 7.78.1 Detailed Description

**template<class T, class Alloc> class Oscl\_TagTree< T, Alloc >**

[Oscl\\_TagTree](#) Class.

### 7.78.2 Member Typedef Documentation

- 7.78.2.1 `template<class T, class Alloc> typedef Oscl_Vector<Node*, Alloc> Oscl_TagTree< T, Alloc >::children_type`
- 7.78.2.2 `template<class T, class Alloc> typedef Oscl_Map<const tag_base_type, node_ptr, Alloc, Oscl_Tag_Base> Oscl_TagTree< T, Alloc >::map_type`
- 7.78.2.3 `template<class T, class Alloc> typedef node_type* Oscl_TagTree< T, Alloc >::node_ptr`
- 7.78.2.4 `template<class T, class Alloc> typedef Node Oscl_TagTree< T, Alloc >::node_type`
- 7.78.2.5 `template<class T, class Alloc> typedef Oscl_Pair<iterator, bool> Oscl_TagTree< T, Alloc >::pair_iterator_bool`
- 7.78.2.6 `template<class T, class Alloc> typedef map_type::size_type Oscl_TagTree< T, Alloc >::size_type`
- 7.78.2.7 `template<class T, class Alloc> typedef tag_type::tag_base_type Oscl_TagTree< T, Alloc >::tag_base_type`
- 7.78.2.8 `template<class T, class Alloc> typedef Oscl_Tag<Alloc> Oscl_TagTree< T, Alloc >::tag_type`
- 7.78.2.9 `template<class T, class Alloc> typedef map_type::value_type Oscl_TagTree< T, Alloc >::value_type`

### 7.78.3 Constructor & Destructor Documentation

- 7.78.3.1 `template<class T, class Alloc> Oscl_TagTree< T, Alloc >::Oscl_TagTree (size_type max_depth = 0) [inline]`

Creates a tag tree with only a root node with tag ""

- 7.78.3.2 `template<class T, class Alloc> Oscl_TagTree< T, Alloc >::Oscl_TagTree (const Oscl_TagTree< T, Alloc > & x) [inline]`

Copy constructor

- 7.78.3.3 `template<class T, class Alloc> Oscl_TagTree< T, Alloc >::~Oscl_TagTree () [inline]`

Destructor

### 7.78.4 Member Function Documentation

- 7.78.4.1 `template<class T, class Alloc> const_iterator Oscl_TagTree< T, Alloc >::begin () const [inline]`

Returns an iterator pointing to the first node in the tree.

**7.78.4.2 template<class T, class Alloc> iterator Oscl\_TagTree< T, Alloc >::begin () [inline]**

Returns an iterator pointing to the first node in the tree.

Referenced by Oscl\_TagTree< PVLogger \*, alloc\_type >::clear(), Oscl\_TagTree< PVLogger \*, alloc\_type >::operator=(), Oscl\_TagTree< PVLogger \*, alloc\_type >::Oscl\_TagTree(), and Oscl\_TagTree< PVLogger \*, alloc\_type >::~Oscl\_TagTree().

**7.78.4.3 template<class T, class Alloc> void Oscl\_TagTree< T, Alloc >::clear () [inline]**

Erases the entire tag tree.

Referenced by Oscl\_TagTree< PVLogger \*, alloc\_type >::operator=().

**7.78.4.4 template<class T, class Alloc> size\_type Oscl\_TagTree< T, Alloc >::count (const tag\_base\_type & x) const [inline]**

Finds an element whose key is x Returns the number of elements with key x. This can only be 0 or 1..

**7.78.4.5 template<class T, class Alloc> bool Oscl\_TagTree< T, Alloc >::empty () const [inline]**

Returns true if tree size is 0

**7.78.4.6 template<class T, class Alloc> const\_iterator Oscl\_TagTree< T, Alloc >::end () const [inline]**

Returns a const iterator pointing to the end of the tree.

**7.78.4.7 template<class T, class Alloc> iterator Oscl\_TagTree< T, Alloc >::end () [inline]**

Returns an iterator pointing to the end of the tree.

Referenced by Oscl\_TagTree< PVLogger \*, alloc\_type >::clear(), Oscl\_TagTree< PVLogger \*, alloc\_type >::erase(), Oscl\_TagTree< PVLogger \*, alloc\_type >::insert(), Oscl\_TagTree< PVLogger \*, alloc\_type >::operator=(), Oscl\_TagTree< PVLogger \*, alloc\_type >::Oscl\_TagTree(), and Oscl\_TagTree< PVLogger \*, alloc\_type >::~Oscl\_TagTree().

**7.78.4.8 template<class T, class Alloc> size\_type Oscl\_TagTree< T, Alloc >::erase (const tag\_base\_type & x) [inline]**

Erases the node with tag x. If the node has children, then the node will not be erased from the tree. It will be replaced with the default node value

**Parameters**

*x* Tag of node to erase

**Returns**

Returns the number of nodes erased. Since one-to-one mapping between nodes and tags, this will be either 0 or 1

**7.78.4.9 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

Referenced by Oscl\_TagTree< PVLogger \*, alloc\_type >::erase().

**7.78.4.10 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\(\)](#)

Referenced by Oscl\_TagTree< PVLogger \*, alloc\_type >::erase().

**7.78.4.11 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*.

Referenced by Oscl\_TagTree< PVLogger \*, alloc\_type >::operator=(), Oscl\_TagTree< PVLogger \*, alloc\_type >::operator[ ](), and Oscl\_TagTree< PVLogger \*, alloc\_type >::Oscl\_TagTree().

**7.78.4.12 template<class T, class Alloc> Oscl\_TagTree< T, Alloc >& Oscl\_TagTree< T, Alloc >::operator= (const Oscl\_TagTree< T, Alloc > & *x*) [inline]**

Assignment operator

**7.78.4.13 template<class T, class Alloc> T& Oscl\_TagTree< T, Alloc >::operator[ ] (const tag\_base\_type & *t*) [inline]**

Returns a reference to the object that is associated with a particular tag. If the map does not already contain such an object, operator[] inserts the default object T().

**7.78.4.14 template<class T, class Alloc> size\_type Oscl\_TagTree< T, Alloc >::size () const  
[inline]**

Returns the number of nodes stored in the tree

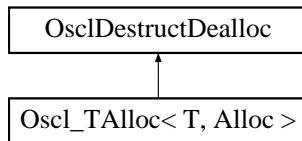
The documentation for this class was generated from the following file:

- [oscl\\_tagtree.h](#)

## 7.79 Oscl\_TAlloc< T, Alloc > Class Template Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for Oscl\_TAlloc< T, Alloc >:



## Data Structures

- struct [rebind](#)

## 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 Member Functions

- virtual [~Oscl\\_TAlloc \(\)](#)
- [pointer allocate\\_fl \(uint32 size, const char \\*file\\_name, const int line\\_num\)](#)
- [pointer allocate \(uint32 size\)](#)
- [pointer alloc\\_and\\_construct\\_fl \(const\\_reference val, const char \\*file\\_name, const int line\\_num\)](#)
- [pointer alloc\\_and\\_construct \(const\\_reference val\)](#)
- [void deallocate \(OsclAny \\*p\)](#)
- [void deallocate \(OsclAny \\*p, size\\_type n\)](#)
- [void destruct\\_and\\_dealloc \(OsclAny \\*p\)](#)
- [pointer address \(reference r\)](#)
- [const\\_pointer address \(const\\_reference r\) const](#)
- [void construct \(pointer p, const\\_reference val\)](#)
- [void destroy \(pointer p\)](#)

---

`template<class T, class Alloc> class Oscl_TAlloc< T, Alloc >`

### 7.79.1 Member Typedef Documentation

- 7.79.1.1 `template<class T, class Alloc> typedef const T* Oscl_TAlloc< T, Alloc >::const_pointer`
- 7.79.1.2 `template<class T, class Alloc> typedef const T& Oscl_TAlloc< T, Alloc >::const_reference`
- 7.79.1.3 `template<class T, class Alloc> typedef T* Oscl_TAlloc< T, Alloc >::pointer`
- 7.79.1.4 `template<class T, class Alloc> typedef T& Oscl_TAlloc< T, Alloc >::reference`
- 7.79.1.5 `template<class T, class Alloc> typedef uint32 Oscl_TAlloc< T, Alloc >::size_type`
- 7.79.1.6 `template<class T, class Alloc> typedef T Oscl_TAlloc< T, Alloc >::value_type`

### 7.79.2 Constructor & Destructor Documentation

- 7.79.2.1 `template<class T, class Alloc> virtual Oscl_TAlloc< T, Alloc >::~Oscl_TAlloc () [inline, virtual]`

### 7.79.3 Member Function Documentation

- 7.79.3.1 `template<class T, class Alloc> const_pointer Oscl_TAlloc< T, Alloc >::address (const_reference r) const [inline]`
- 7.79.3.2 `template<class T, class Alloc> pointer Oscl_TAlloc< T, Alloc >::address (reference r) [inline]`
- 7.79.3.3 `template<class T, class Alloc> pointer Oscl_TAlloc< T, Alloc >::alloc_and_construct (const_reference val) [inline]`
- 7.79.3.4 `template<class T, class Alloc> pointer Oscl_TAlloc< T, Alloc >::alloc_and_construct_fl (const_reference val, const char *file_name, const int line_num) [inline]`
- 7.79.3.5 `template<class T, class Alloc> pointer Oscl_TAlloc< T, Alloc >::allocate (uint32 size) [inline]`

Referenced by MM\_AllocNode::operator new(), MM\_AllocInfo::operator new(), MM\_Stats\_CB::operator new(), OsclMemStatsNode::operator new(), MM\_FailInsertParam::operator new(), and MM\_Stats\_t::operator new().

- 7.79.3.6 `template<class T, class Alloc> pointer Oscl_TAlloc< T, Alloc >::allocate_fl (uint32 size, const char *file_name, const int line_num) [inline]`
- 7.79.3.7 `template<class T, class Alloc> void Oscl_TAlloc< T, Alloc >::construct (pointer p, const_reference val) [inline]`

Referenced by Oscl\_TAlloc< node\_type, Alloc >::alloc\_and\_construct(), and Oscl\_TAlloc< node\_type, Alloc >::alloc\_and\_construct\_fl().

**7.79.3.8 template<class T, class Alloc> void Oscl\_TAlloc< T, Alloc >::deallocate (OsclAny \* *p*, size\_type *n*) [inline]**

**7.79.3.9 template<class T, class Alloc> void Oscl\_TAlloc< T, Alloc >::deallocate (OsclAny \* *p*) [inline]**

Referenced by OsclTimer< Alloc >::Cancel(), OsclTimer< Alloc >::Clear(), Oscl\_TAlloc< node\_type, Alloc >::destruct\_and\_dealloc(), MM\_AllocNode::operator delete(), MM\_AllocInfo::operator delete(), MM\_Stats\_CB::operator delete(), OsclMemStatsNode::operator delete(), MM\_FailInsertParam::operator delete(), MM\_Stats\_t::operator delete(), OsclTimer< Alloc >::TimerBaseElapsed(), MM\_AllocInfo::~MM\_AllocInfo(), Oscl\_Tag< Alloc >::~Oscl\_Tag(), OsclMemStatsNode::~OsclMemStatsNode(), OsclTimer< Alloc >::~OsclTimer(), and PVLogger::~PVLogger().

**7.79.3.10 template<class T, class Alloc> void Oscl\_TAlloc< T, Alloc >::destroy (pointer *p*) [inline]**

Referenced by Oscl\_TAlloc< node\_type, Alloc >::destruct\_and\_dealloc().

**7.79.3.11 template<class T, class Alloc> void Oscl\_TAlloc< T, Alloc >::destruct\_and\_dealloc (OsclAny \* *p*) [inline, virtual]**

Implements [OsclDestructDealloc](#).

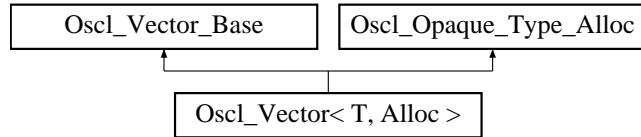
The documentation for this class was generated from the following file:

- [oscl\\_defalloc.h](#)

## 7.80 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 Member Functions

- `Oscl_Vector ()`
- `Oscl_Vector (uint32 n)`
- `Oscl_Vector (const Oscl_Vector< T, Alloc > &x)`
- `virtual ~Oscl_Vector ()`
- `Oscl_Vector< T, Alloc > & operator= (const Oscl_Vector< T, Alloc > &x)`
- `void push_back (const T &x)`
- `void push_front (const T &x)`
- `iterator insert (iterator pos, const T &x)`
- `T & operator[] (uint32 n)`
- `const T & operator[] (uint32 n) const`
- `T & front ()`
- `const T & front () const`
- `T & back ()`
- `const T & back () const`
- `void pop_back ()`
- `void clear ()`
- `void destroy ()`
- `iterator begin () const`
- `iterator end () const`
- `iterator erase (iterator pos)`
- `iterator erase (iterator first, iterator last)`

### 7.80.1 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.

### 7.80.2 Member Typedef Documentation

**7.80.2.1 template<class T, class Alloc> typedef const T\* Oscl\_Vector< T, Alloc >::const\_iterator**

**7.80.2.2 template<class T, class Alloc> typedef const T& Oscl\_Vector< T, Alloc >::const\_reference**

**7.80.2.3 template<class T, class Alloc> typedef T\* Oscl\_Vector< T, Alloc >::iterator**

**7.80.2.4 template<class T, class Alloc> typedef T\* Oscl\_Vector< T, Alloc >::pointer**

**7.80.2.5 template<class T, class Alloc> typedef T& Oscl\_Vector< T, Alloc >::reference**

**7.80.2.6 template<class T, class Alloc> typedef T Oscl\_Vector< T, Alloc >::value\_type**

### 7.80.3 Constructor & Destructor Documentation

**7.80.3.1 template<class T, class Alloc> Oscl\_Vector< T, Alloc >::Oscl\_Vector () [inline]**

Creates an empty vector.

**7.80.3.2 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.

**7.80.3.3 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.

---

**7.80.3.4 template<class T, class Alloc> virtual Oscl\_Vector< T, Alloc >::~Oscl\_Vector ()  
[inline, virtual]**

The destructor.

#### 7.80.4 Member Function Documentation

**7.80.4.1 template<class T, class Alloc> const T& Oscl\_Vector< T, Alloc >::back () const  
[inline]**

Returns the last element.

**7.80.4.2 template<class T, class Alloc> T& Oscl\_Vector< T, Alloc >::back () [inline]**

Returns the last element.

**7.80.4.3 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](#).

Referenced by OsclTimer< Alloc >::Cancel(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::clear(), OsclTimer< Alloc >::Clear(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::front(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::operator[](), PVLogger::RemoveAppender(), Oscl\_TagTree< T, Alloc >::Node::sort\_children(), OsclTimer< Alloc >::TimerBaseElapsed(), and OsclTimer< Alloc >::~OsclTimer().

**7.80.4.4 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::clear () [inline]**

Removes all elements.

Referenced by OsclTimer< Alloc >::Clear(), and OsclTimer< Alloc >::TimerBaseElapsed().

**7.80.4.5 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](#).

Referenced by Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::destroy(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::pop\_back(), and Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::~Oscl\_Vector().

**7.80.4.6 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](#).

Referenced by Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::back(), OsclTimer< Alloc >::Cancel(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::clear(), OsclTimer< Alloc >::Clear(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::pop\_back(), PVLogger::RemoveAppender(), Oscl\_TagTree< T, Alloc >::Node::sort\_children(), OsclTimer< Alloc >::TimerBaseElapsed(), and OsclTimer< Alloc >::~OsclTimer().

#### 7.80.4.7 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

#### 7.80.4.8 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

Referenced by OsclTimer< Alloc >::Cancel(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::clear(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::erase(), PVLogger::RemoveAppender(), and OsclTimer< Alloc >::TimerBaseElapsed().

#### 7.80.4.9 template<class T, class Alloc> const T& Oscl\_Vector< T, Alloc >::front () const [inline]

Returns the first element.

#### 7.80.4.10 template<class T, class Alloc> T& Oscl\_Vector< T, Alloc >::front () [inline]

Returns the first element.

#### 7.80.4.11 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

Referenced by Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::insert().

**7.80.4.12 template<class T, class Alloc> Oscl\_Vector<T, Alloc>& Oscl\_Vector< T, Alloc >::operator= (const Oscl\_Vector< T, Alloc > &x) [inline]**

The assignment operator

**7.80.4.13 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

**7.80.4.14 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

**7.80.4.15 template<class T, class Alloc> void Oscl\_Vector< T, Alloc >::pop\_back () [inline]**

Removes the last element.

Reimplemented from [Oscl\\_Vector\\_Base](#).

**7.80.4.16 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

Referenced by PVLogger::AddAppender(), PVLogger::AddFilter(), Oscl\_File::AddFixedCache(), OsclTimer< Alloc >::Cancel(), GetHostNameParam::PersistHostAddress(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::push\_back(), Oscl\_File::RemoveFixedCache(), and OsclTimer< Alloc >::Request().

**7.80.4.17 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

Referenced by Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::push\_front().

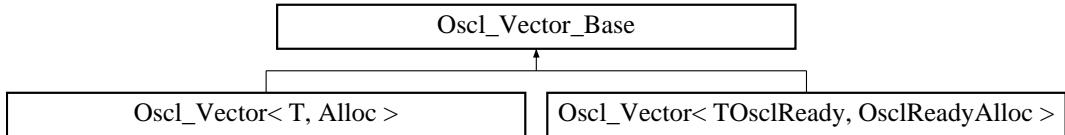
The documentation for this class was generated from the following file:

- [oscl\\_vector.h](#)

## 7.81 Oscl\_Vector\_Base Class Reference

```
#include <oscl_vector.h>
```

Inheritance diagram for Oscl\_Vector\_Base:



### Public Member Functions

- uint32 `size () const`
- uint32 `capacity () const`
- bool `empty () const`
- OSCL\_IMPORT\_REF void `reserve (uint32 n)`

### Protected Member Functions

- OSCL\_IMPORT\_REF void `construct (Oscl_Opaque_Type_Alloc *aType)`
- OSCL\_IMPORT\_REF void `construct (Oscl_Opaque_Type_Alloc *aType, uint32 n)`
- OSCL\_IMPORT\_REF void `construct (Oscl_Opaque_Type_Alloc *aType, const Oscl_Vector_Base &x)`
- virtual `~Oscl_Vector_Base ()`
- OSCL\_IMPORT\_REF void `push_back (const OsclAny *x)`
- OSCL\_IMPORT\_REF void `pop_back ()`
- OSCL\_IMPORT\_REF void `push_front (const OsclAny *x)`
- OSCL\_IMPORT\_REF `OsclAny * insert (OsclAny *pos, const OsclAny *x)`
- OSCL\_IMPORT\_REF `OsclAny * erase (OsclAny *pos)`
- OSCL\_IMPORT\_REF `OsclAny * erase (OsclAny *first, OsclAny *last)`
- OSCL\_IMPORT\_REF void `assign_vector (const Oscl_Vector_Base &x)`
- OSCL\_IMPORT\_REF void `destroy ()`

### Protected Attributes

- uint32 `numelems`
- uint32 `bufsize`
- `OsclAny * elems`
- uint32 `sizeof_T`

### Friends

- class `OsclPriorityQueueBase`

### 7.81.1 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`.

### 7.81.2 Constructor & Destructor Documentation

#### 7.81.2.1 `virtual Oscl_Vector_Base::~Oscl_Vector_Base () [inline, protected, virtual]`

The destructor.

### 7.81.3 Member Function Documentation

#### 7.81.3.1 `OSCL_IMPORT_REF void Oscl_Vector_Base::assign_vector (const Oscl_Vector_Base & x) [protected]`

Referenced by `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >::operator=()`.

#### 7.81.3.2 `uint32 Oscl_Vector_Base::capacity () const [inline]`

Returns the allocated memory of the vector in units of number of elements.

References bufsize.

Referenced by `GetHostNameParam::canPersistMoreHostAddresses()`, and `GetHostNameParam::PersistHostAddress()`.

#### 7.81.3.3 `OSCL_IMPORT_REF void Oscl_Vector_Base::construct (Oscl_Opaque_Type_Alloc * aType, const Oscl_Vector_Base & x) [protected]`

#### 7.81.3.4 `OSCL_IMPORT_REF void Oscl_Vector_Base::construct (Oscl_Opaque_Type_Alloc * aType, uint32 n) [protected]`

#### 7.81.3.5 `OSCL_IMPORT_REF void Oscl_Vector_Base::construct (Oscl_Opaque_Type_Alloc * aType) [protected]`

Referenced by `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >::Oscl_Vector()`.

#### 7.81.3.6 `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< OsclFixedCacheParam, OsclMemAllocator >`, `Oscl_Vector< OsclSocketRequest *, OsclMemAllocator >`, `Oscl_Vector< entry_type *, Alloc >`, `Oscl_Vector< OSCL_HeapString< OsclMemAllocator >, OsclMemAllocator >`, `Oscl_Vector< OsclAsyncFileBuffer *, OsclMemAllocator >`, `Oscl_Vector< TOsclFileOffset, OsclMemAllocator >`, `Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator >`, `Oscl_Vector< OsclSharedPtr< PVLoggerFilter >, alloc_type >`, `Oscl_Vector< TOsclReady, OsclReadyAlloc >`, `Oscl_Vector< OsclFileCacheBuffer, OsclMemAllocator >`, `Oscl_Vector< OsclSharedPtr< PVLoggerAppender`

`>, alloc_type >, Oscl_Vector< OsclAny *, OsclMemAllocator >, and Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >.`

#### 7.81.3.7 `bool Oscl_Vector_Base::empty () const [inline]`

True if the vector's size is 0.

References numelems.

Referenced by `Oscl_TagTree< T, Alloc >::Node::sort_children()`, and `OsclTimer< Alloc >::TimerBaseElapsed()`.

#### 7.81.3.8 `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

#### 7.81.3.9 `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

#### 7.81.3.10 `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

#### 7.81.3.11 `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< OsclFixedCacheParam, OsclMemAllocator >`, `Oscl_Vector< OsclSocketRequest *, OsclMemAllocator >`, `Oscl_Vector<`

`entry_type *, Alloc >, Oscl_Vector< OSCL_HeapString< OsclMemAllocator >, OsclMemAllocator >, Oscl_Vector< OsclAsyncFileBuffer *, OsclMemAllocator >, Oscl_Vector< TOsclFileOffset, OsclMemAllocator >, Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator >, Oscl_Vector< OsclSharedPtr< PVLoggerFilter >, alloc_type >, Oscl_Vector< TOsclReady, OsclReadyAlloc >, Oscl_Vector< OsclFileCacheBuffer, OsclMemAllocator >, Oscl_Vector< OsclSharedPtr< PVLoggerAppender >, alloc_type >, Oscl_Vector< OsclAny *, OsclMemAllocator >, and Oscl_Vector< OsclNetworkAddress, OsclMemAllocator >.`

#### 7.81.3.12 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

#### 7.81.3.13 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

#### 7.81.3.14 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

#### 7.81.3.15 uint32 Oscl\_Vector\_Base::size () const [inline]

Returns the size of the vector in units of number of elements.

References numelems.

Referenced by GetHostByNameParam::canPersistMoreHostAddresses(), PVLogger::GetNumAppenders(), and GetHostByNameParam::PersistHostAddress().

## 7.81.4 Friends And Related Function Documentation

### 7.81.4.1 friend class OsclPriorityQueueBase [friend]

## 7.81.5 Field Documentation

### 7.81.5.1 uint32 Oscl\_Vector\_Base::bufsize [protected]

Referenced by capacity().

### 7.81.5.2 OsclAny\* Oscl\_Vector\_Base::elems [protected]

Referenced by Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::begin(), and Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::end().

### 7.81.5.3 uint32 Oscl\_Vector\_Base::numelems [protected]

Referenced by empty(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::end(), Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::pop\_back(), and size().

### 7.81.5.4 uint32 Oscl\_Vector\_Base::sizeof\_T [protected]

Referenced by Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator >::Oscl\_Vector().

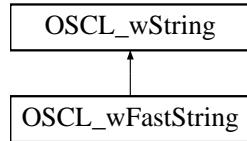
The documentation for this class was generated from the following file:

- [oscl\\_vector.h](#)

## 7.82 OSCL\_wFastString Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_wFastString:



### Public Types

- `typedef OSCL_wString::chartype chartype`
- `typedef TOSCL_wStringOp optype`
- `typedef OSCL_String::chartype other_chartype`

### Public Member Functions

- `OSCL_IMPORT_REF OSCL_wFastString()`
- `OSCL_IMPORT_REF OSCL_wFastString(const OSCL_wFastString &src)`
- `OSCL_IMPORT_REF OSCL_wFastString(const chartype *cstr)`
- `OSCL_IMPORT_REF OSCL_wFastString(chartype *buf, uint32 maxlen)`
- `OSCL_IMPORT_REF ~OSCL_wFastString()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_wFastString & operator=(const OSCL_wFastString &src)`
- `OSCL_IMPORT_REF OSCL_wFastString & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(chartype *cstr, uint32 maxlen)`
- `OSCL_IMPORT_REF void set(const other_chartype *buf, uint32 numofbyte, optype op)`
- `OSCL_IMPORT_REF void set_length()`

### Friends

- class `OSCL_wString`

#### 7.82.1 Detailed Description

`OSCL_wFastString` is identical to `OSCL_FastString` except that it uses wide-character format. For descriptions, see `OSCL_FastString`.

#### 7.82.2 Member Typedef Documentation

##### 7.82.2.1 `typedef OSCL_wString::chartype OSCL_wFastString::chartype`

Reimplemented from `OSCL_wString`.

7.82.2.2 **typedef TOSCL\_wStringOp OSCL\_wFastString::optype**

7.82.2.3 **typedef OSCL\_String::chartype OSCL\_wFastString::other\_chartype**

### 7.82.3 Constructor & Destructor Documentation

7.82.3.1 **OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString ()**

7.82.3.2 **OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString (const OSCL\_wFastString & src)**

7.82.3.3 **OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString (const chartype \* cstr)**

7.82.3.4 **OSCL\_IMPORT\_REF OSCL\_wFastString::OSCL\_wFastString (chartype \* buf, uint32 maxlen)**

7.82.3.5 **OSCL\_IMPORT\_REF OSCL\_wFastString::~OSCL\_wFastString ()**

### 7.82.4 Member Function Documentation

7.82.4.1 **OSCL\_IMPORT\_REF const chartype\* OSCL\_wFastString::get\_cstr () const [virtual]**

Implements [OSCL\\_wString](#).

7.82.4.2 **OSCL\_IMPORT\_REF uint32 OSCL\_wFastString::get\_maxsize () const [virtual]**

Implements [OSCL\\_wString](#).

7.82.4.3 **OSCL\_IMPORT\_REF uint32 OSCL\_wFastString::get\_size () const [virtual]**

Implements [OSCL\\_wString](#).

7.82.4.4 **OSCL\_IMPORT\_REF chartype\* OSCL\_wFastString::get\_str () const [virtual]**

Implements [OSCL\\_wString](#).

7.82.4.5 **OSCL\_IMPORT\_REF OSCL\_wFastString& OSCL\_wFastString::operator= (const chartype \* cstr)**

Reimplemented from [OSCL\\_wString](#).

- 7.82.4.6 **OSCL\_IMPORT\_REF OSCL\_wFastString& OSCL\_wFastString::operator= (const OSCL\_wFastString & src)**
- 7.82.4.7 **OSCL\_IMPORT\_REF void OSCL\_wFastString::set (const other\_chartype \* buf, uint32 numofbyte, optype op)**
- 7.82.4.8 **OSCL\_IMPORT\_REF void OSCL\_wFastString::set (chartype \* cstr, uint32 maxlen)**
- 7.82.4.9 **OSCL\_IMPORT\_REF void OSCL\_wFastString::set\_length ()**

## 7.82.5 Friends And Related Function Documentation

- 7.82.5.1 **friend class OSCL\_wString [friend]**

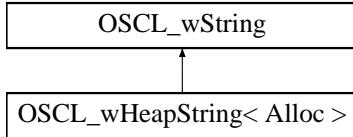
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 7.83 OSCL\_wHeapString< Alloc > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_wHeapString< Alloc >:



### Public Types

- `typedef OSCL_wString::chartype chartype`
- `typedef TOSCL_wStringOp optype`
- `typedef OSCL_String::chartype other_chartype`

### Public Member Functions

- `OSCL_wHeapString()`
- `OSCL_wHeapString(const OSCL_wHeapString &src)`
- `OSCL_wHeapString(const OSCL_wString &src)`
- `OSCL_wHeapString(const chartype *cstr)`
- `OSCL_wHeapString(const chartype *buf, uint32 length)`
- `~OSCL_wHeapString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype * get_cstr() const`
- `chartype * get_str() const`
- `OSCL_wHeapString & operator=(const OSCL_wHeapString &src)`
- `OSCL_wHeapString & operator=(const OSCL_wString &src)`
- `OSCL_wHeapString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`
- `void set(const other_chartype *buf, optype op)`
- `void set(const other_chartype *buf, uint32 length, optype op)`

### Friends

- class `OSCL_wString`

#### 7.83.1 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`.

### 7.83.2 Member Typedef Documentation

**7.83.2.1 template<class Alloc> typedef OSCL\_wString::chartype OSCL\_wHeapString< Alloc >::chartype**

Reimplemented from [OSCL\\_wString](#).

**7.83.2.2 template<class Alloc> typedef TOSCL\_wStringOp OSCL\_wHeapString< Alloc >::optype**

**7.83.2.3 template<class Alloc> typedef OSCL\_String::chartype OSCL\_wHeapString< Alloc >::other\_chartype**

### 7.83.3 Friends And Related Function Documentation

**7.83.3.1 template<class Alloc> friend class OSCL\_wString [friend]**

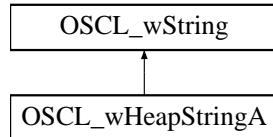
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 7.84 OSCL\_wHeapStringA Class Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_wHeapStringA:



### Public Types

- `typedef OSCL_wString::chartype chartype`
- `typedef TOSCL_wStringOp optype`
- `typedef OSCL_String::chartype other_chartype`

### Public Member Functions

- `OSCL_IMPORT_REF OSCL_wHeapStringA()`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wHeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wHeapStringA &src, Oscl_DefAlloc *alloc, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const OSCL_wString &src, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const chartype *cstr, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA(const chartype *buf, uint32 length, Oscl_DefAlloc *alloc=NULL, OsclRefCounter *ref=NULL)`
- `OSCL_IMPORT_REF ~OSCL_wHeapStringA()`
- `OSCL_IMPORT_REF uint32 get_size() const`
- `OSCL_IMPORT_REF uint32 get_maxsize() const`
- `OSCL_IMPORT_REF const chartype * get_cstr() const`
- `OSCL_IMPORT_REF chartype * get_str() const`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const OSCL_wHeapStringA &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const OSCL_wString &src)`
- `OSCL_IMPORT_REF OSCL_wHeapStringA & operator=(const chartype *cstr)`
- `OSCL_IMPORT_REF void set(const chartype *buf, uint32 length)`
- `OSCL_IMPORT_REF void set(const other_chartype *buf, optype op)`
- `OSCL_IMPORT_REF void set(const other_chartype *buf, uint32 length, optype op)`

### Friends

- class `OSCL_wString`

### 7.84.1 Detailed Description

`OSCL_wHeapStringA` is identical to `OSCL_HeapStringA` except that it uses wide-character format. For descriptions, see `OSCL_HeapStringA`.

### 7.84.2 Member Typedef Documentation

#### 7.84.2.1 `typedef OSCL_wString::chartype OSCL_wHeapStringA::chartype`

Reimplemented from `OSCL_wString`.

#### 7.84.2.2 `typedef TOSCL_wStringOp OSCL_wHeapStringA::optype`

#### 7.84.2.3 `typedef OSCL_String::chartype OSCL_wHeapStringA::other_chartype`

### 7.84.3 Constructor & Destructor Documentation

#### 7.84.3.1 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA ()`

#### 7.84.3.2 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (Oscl_DefAlloc * alloc, OsclRefCounter * ref = NULL)`

#### 7.84.3.3 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL_wHeapStringA & src)`

#### 7.84.3.4 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL_wHeapStringA & src, Oscl_DefAlloc * alloc, OsclRefCounter * ref = NULL)`

#### 7.84.3.5 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const OSCL_wString & src, Oscl_DefAlloc * alloc = NULL, OsclRefCounter * ref = NULL)`

#### 7.84.3.6 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const chartype * cstr, Oscl_DefAlloc * alloc = NULL, OsclRefCounter * ref = NULL)`

#### 7.84.3.7 `OSCL_IMPORT_REF OSCL_wHeapStringA::OSCL_wHeapStringA (const chartype * buf, uint32 length, Oscl_DefAlloc * alloc = NULL, OsclRefCounter * ref = NULL)`

#### 7.84.3.8 `OSCL_IMPORT_REF OSCL_wHeapStringA::~OSCL_wHeapStringA ()`

### 7.84.4 Member Function Documentation

#### 7.84.4.1 `OSCL_IMPORT_REF const chartype* OSCL_wHeapStringA::get_cstr () const [virtual]`

Implements `OSCL_wString`.

#### 7.84.4.2 `OSCL_IMPORT_REF uint32 OSCL_wHeapStringA::get_maxsize () const [virtual]`

Implements `OSCL_wString`.

**7.84.4.3 OSCL\_IMPORT\_REF uint32 OSCL\_wHeapStringA::get\_size () const [virtual]**

Implements [OSCL\\_wString](#).

**7.84.4.4 OSCL\_IMPORT\_REF chartype\* OSCL\_wHeapStringA::get\_str () const [virtual]**

Implements [OSCL\\_wString](#).

**7.84.4.5 OSCL\_IMPORT\_REF OSCL\_wHeapStringA& OSCL\_wHeapStringA::operator= (const chartype \* *cstr*)**

Reimplemented from [OSCL\\_wString](#).

**7.84.4.6 OSCL\_IMPORT\_REF OSCL\_wHeapStringA& OSCL\_wHeapStringA::operator= (const OSCL\_wString & *src*)**

Reimplemented from [OSCL\\_wString](#).

**7.84.4.7 OSCL\_IMPORT\_REF OSCL\_wHeapStringA& OSCL\_wHeapStringA::operator= (const OSCL\_wHeapStringA & *src*)****7.84.4.8 OSCL\_IMPORT\_REF void OSCL\_wHeapStringA::set (const other\_chartype \* *buf*, uint32 *length*, optype *op*)****7.84.4.9 OSCL\_IMPORT\_REF void OSCL\_wHeapStringA::set (const other\_chartype \* *buf*, optype *op*)****7.84.4.10 OSCL\_IMPORT\_REF void OSCL\_wHeapStringA::set (const chartype \* *buf*, uint32 *length*)**

## 7.84.5 Friends And Related Function Documentation

**7.84.5.1 friend class OSCL\_wString [friend]**

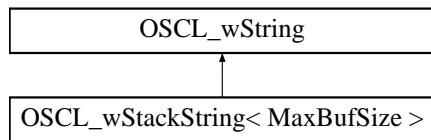
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 7.85 OSCL\_wStackString< MaxBufSize > Class Template Reference

```
#include <oscl_string_containers.h>
```

Inheritance diagram for OSCL\_wStackString< MaxBufSize >:



### Public Types

- `typedef OSCL_wString::chartype chartype`
- `typedef TOSCL_wStringOp optype`
- `typedef OSCL_String::chartype other_chartype`

### Public Member Functions

- `OSCL_wStackString()`
- `OSCL_wStackString(const OSCL_wStackString &src)`
- `OSCL_wStackString(const OSCL_wString &src)`
- `OSCL_wStackString(const chartype *cstr)`
- `OSCL_wStackString(const chartype *buf, uint32 length)`
- `~OSCL_wStackString()`
- `uint32 get_size() const`
- `uint32 get_maxsize() const`
- `const chartype * get_cstr() const`
- `chartype * get_str() const`
- `OSCL_wStackString & operator=(const OSCL_wStackString &src)`
- `OSCL_wStackString & operator=(const OSCL_wString &src)`
- `OSCL_wStackString & operator=(const chartype *cstr)`
- `void set(const chartype *buf, uint32 length)`
- `void set(const other_chartype *buf, optype op)`
- `void set(const other_chartype *buf, uint32 length, optype op)`

### Friends

- class `OSCL_wString`

#### 7.85.1 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`.

## 7.85.2 Member Typedef Documentation

7.85.2.1 `template<uint32 MaxBufSize> typedef OSCL_wString::chartype OSCL_wStackString<MaxBufSize >::chartype`

Reimplemented from [OSCL\\_wString](#).

7.85.2.2 `template<uint32 MaxBufSize> typedef TOSCL_wStringOp OSCL_wStackString<MaxBufSize >::optype`

7.85.2.3 `template<uint32 MaxBufSize> typedef OSCL_String::chartype OSCL_wStackString<MaxBufSize >::other_chartype`

## 7.85.3 Friends And Related Function Documentation

7.85.3.1 `template<uint32 MaxBufSize> friend class OSCL_wString [friend]`

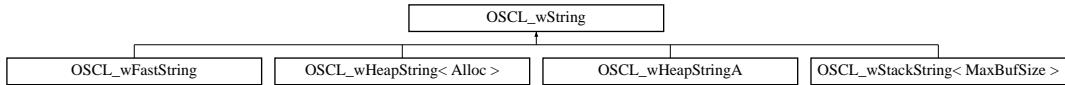
The documentation for this class was generated from the following file:

- [oscl\\_string\\_containers.h](#)

## 7.86 OSCL\_wString Class Reference

```
#include <oscl_string.h>
```

Inheritance diagram for OSCL\_wString:



### Public Types

- `typedef oscl_wchar chartype`

### Public Member Functions

- `virtual uint32 get_size () const =0`
- `virtual uint32 get_maxsize () const =0`
- `virtual const chartype * get_cstr () const =0`
- `virtual bool is_writable () const`
- `virtual chartype * get_str () const =0`
- `OSCL_wString & operator= (const OSCL_wString &src)`
- `OSCL_wString & operator= (const chartype *cstr)`
- `OSCL_wString & operator+= (const OSCL_wString &src)`
- `OSCL_wString & operator+= (const chartype *cstr)`
- `OSCL_wString & operator+= (const chartype c)`
- `bool operator== (const OSCL_wString &src) const`
- `bool operator!= (const OSCL_wString &src) const`
- `bool operator< (const OSCL_wString &src) const`
- `bool operator<= (const OSCL_wString &src) const`
- `bool operator> (const OSCL_wString &src) const`
- `bool operator>= (const OSCL_wString &src) const`
- `bool operator== (const chartype *cstr) const`
- `chartype operator[ ] (uint32 index) const`
- `virtual chartype read (uint32 index) const`
- `virtual uint32 setrep_to_wide_char (const char *src, uint32 len, TOSCL_wStringOp op, Oscl_DefAlloc *aAlloc)`
- `virtual int8 hash () const`
- `virtual void write (uint32 index, chartype c)`
- `virtual void write (uint32 offset, uint32 length, const chartype *buf)`

### Protected Member Functions

- `OSCL_wString ()`
- `virtual ~OSCL_wString ()`
- `virtual void set_rep (const chartype *cstr)=0`
- `virtual void append_rep (const chartype *cstr)=0`
- `virtual void set_rep (const OSCL_wString &src)=0`
- `virtual void append_rep (const OSCL_wString &src)=0`
- `virtual void set_len (uint32 len)=0`

### 7.86.1 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](#).

### 7.86.2 Member Typedef Documentation

#### 7.86.2.1 `typedef oscl_wchar OSCL_wString::chartype`

Reimplemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

### 7.86.3 Constructor & Destructor Documentation

#### 7.86.3.1 `OSCL_wString::OSCL_wString () [protected]`

#### 7.86.3.2 `virtual OSCL_wString::~OSCL_wString () [protected, virtual]`

### 7.86.4 Member Function Documentation

#### 7.86.4.1 `virtual void OSCL_wString::append_rep (const OSCL_wString & src) [protected, pure virtual]`

#### 7.86.4.2 `virtual void OSCL_wString::append_rep (const chartype * cstr) [protected, pure virtual]`

#### 7.86.4.3 `virtual const chartype* OSCL_wString::get_cstr () const [pure virtual]`

Implemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

#### 7.86.4.4 `virtual uint32 OSCL_wString::get_maxsize () const [pure virtual]`

Implemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

#### 7.86.4.5 `virtual uint32 OSCL_wString::get_size () const [pure virtual]`

Implemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

#### 7.86.4.6 `virtual chartype* OSCL_wString::get_str () const [pure virtual]`

Implemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

- 7.86.4.7 **virtual int8 OSCL\_wString::hash () const [virtual]**
- 7.86.4.8 **virtual bool OSCL\_wString::is\_writable () const [virtual]**
- 7.86.4.9 **bool OSCL\_wString::operator!= (const OSCL\_wString & src) const**
- 7.86.4.10 **OSCL\_wString& OSCL\_wString::operator+= (const chartype c)**
- 7.86.4.11 **OSCL\_wString& OSCL\_wString::operator+= (const chartype \* cstr)**
- 7.86.4.12 **OSCL\_wString& OSCL\_wString::operator+= (const OSCL\_wString & src)**
- 7.86.4.13 **bool OSCL\_wString::operator< (const OSCL\_wString & src) const**
- 7.86.4.14 **bool OSCL\_wString::operator<= (const OSCL\_wString & src) const**
- 7.86.4.15 **OSCL\_wString& OSCL\_wString::operator= (const chartype \* cstr)**

Reimplemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), [OSCL\\_wStackString< MaxBufSize >](#), and [OSCL\\_wFastString](#).

- 7.86.4.16 **OSCL\_wString& OSCL\_wString::operator= (const OSCL\_wString & src)**

Reimplemented in [OSCL\\_wHeapString< Alloc >](#), [OSCL\\_wHeapStringA](#), and [OSCL\\_wStackString< MaxBufSize >](#).

- 7.86.4.17 **bool OSCL\_wString::operator== (const chartype \* cstr) const**
- 7.86.4.18 **bool OSCL\_wString::operator== (const OSCL\_wString & src) const**
- 7.86.4.19 **bool OSCL\_wString::operator> (const OSCL\_wString & src) const**
- 7.86.4.20 **bool OSCL\_wString::operator>= (const OSCL\_wString & src) const**
- 7.86.4.21 **chartype OSCL\_wString::operator[] (uint32 index) const**
- 7.86.4.22 **virtual chartype OSCL\_wString::read (uint32 index) const [virtual]**
- 7.86.4.23 **virtual void OSCL\_wString::set\_len (uint32 len) [protected, pure virtual]**
- 7.86.4.24 **virtual void OSCL\_wString::set\_rep (const OSCL\_wString & src) [protected, pure virtual]**
- 7.86.4.25 **virtual void OSCL\_wString::set\_rep (const chartype \* cstr) [protected, pure virtual]**

Referenced by [OSCL\\_wStackString< MaxBufSize >::operator=\(\)](#), [OSCL\\_StackString< MaxBufSize >::operator=\(\)](#), [OSCL\\_wHeapString< Alloc >::operator=\(\)](#), [OSCL\\_HeapString< Alloc >::operator=\(\)](#), [OSCL\\_HeapString< Alloc >::OSCL\\_HeapString\(\)](#), [OSCL\\_StackString< MaxBufSize >::OSCL\\_StackString\(\)](#), [OSCL\\_wHeapString< Alloc >::OSCL\\_wHeapString\(\)](#), [OSCL\\_wStackString< MaxBufSize >::OSCL\\_wStackString\(\)](#), [OSCL\\_wHeapString< Alloc >::set\(\)](#), and [OSCL\\_HeapString< Alloc >::set\(\)](#).

**7.86.4.26 virtual uint32 OSCL\_wString::setrep\_to\_wide\_char (const char \* *src*, uint32 *len*,  
TOSCL\_wStringOp *op*, Oscl\_DefAlloc \* *aAlloc*) [virtual]**

Referenced by OSCL\_wStackString< MaxBufSize >::set(), and OSCL\_wHeapString< Alloc >::set().

**7.86.4.27 virtual void OSCL\_wString::write (uint32 *offset*, uint32 *length*, const chartype \* *buf*)  
[virtual]**

**7.86.4.28 virtual void OSCL\_wString::write (uint32 *index*, chartype *c*) [virtual]**

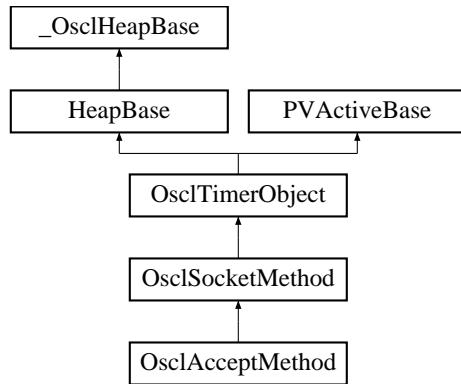
The documentation for this class was generated from the following file:

- [oscl\\_string.h](#)

## 7.87 OsclAcceptMethod Class Reference

```
#include <oscl_socket_accept.h>
```

Inheritance diagram for OsclAcceptMethod:



### Public Member Functions

- [~OsclAcceptMethod \(\)](#)
- [TPVSocketEvent Accept \(int32 aTimeout\)](#)
- [void DiscardAcceptedSocket \(\)](#)
- [OsclSocketI \\* GetAcceptedSocket \(\)](#)
- [OsclAcceptRequest \\* AcceptRequest \(\)](#)

### Static Public Member Functions

- static [OsclAcceptMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 7.87.1 Constructor & Destructor Documentation

##### 7.87.1.1 OsclAcceptMethod::~OsclAcceptMethod ()

#### 7.87.2 Member Function Documentation

##### 7.87.2.1 TPVSocketEvent OsclAcceptMethod::Accept (int32 aTimeout)

Referenced by [OsclTCPSocketI::Accept\(\)](#).

##### 7.87.2.2 OsclAcceptRequest\* OsclAcceptMethod::AcceptRequest () [inline]

References [OsclSocketMethod::iSocketRequestAO](#).

**7.87.2.3 void OsclAcceptMethod::DiscardAcceptedSocket ()**

**7.87.2.4 OsclSocketI\* OsclAcceptMethod::GetAcceptedSocket ()**

**7.87.2.5 static OsclAcceptMethod\* OsclAcceptMethod::NewL (OsclIPSocketI & c) [static]**

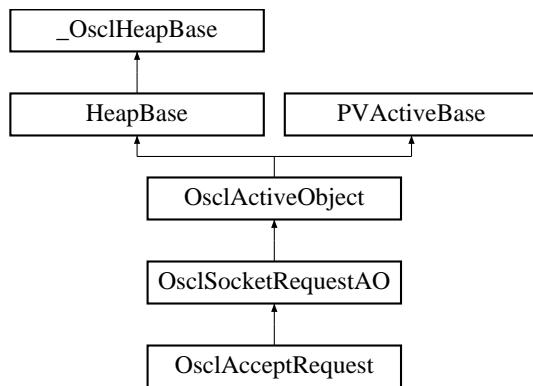
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_accept.h](#)

## 7.88 OsclAcceptRequest Class Reference

```
#include <oscl_socket_accept.h>
```

Inheritance diagram for OsclAcceptRequest:



### Public Member Functions

- [OsclAcceptRequest \(OsclSocketMethod &c\)](#)
- [void Accept \(OsclSocketI &aSocket\)](#)

#### 7.88.1 Constructor & Destructor Documentation

**7.88.1.1 OsclAcceptRequest::OsclAcceptRequest (OsclSocketMethod & c) [inline]**

#### 7.88.2 Member Function Documentation

**7.88.2.1 void OsclAcceptRequest::Accept (OsclSocketI & aSocket)**

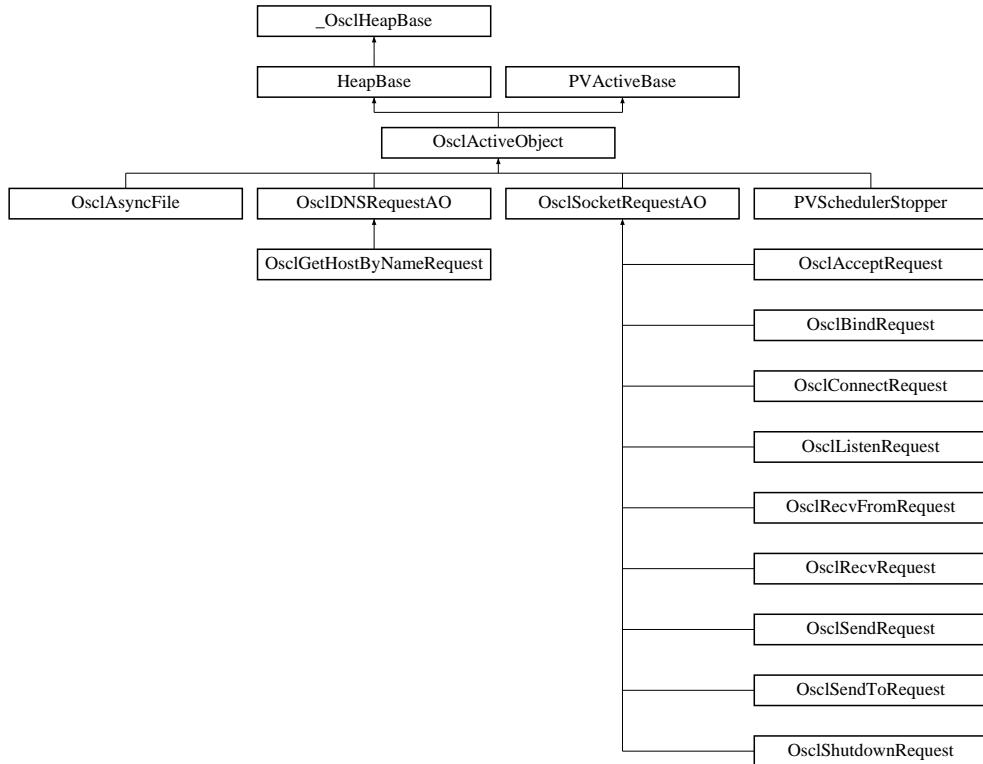
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_accept.h](#)

## 7.89 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 Member Functions

- [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 Member Functions

- virtual OSCL\_IMPORT\_REF void `DoCancel` ()
- virtual OSCL\_IMPORT\_REF int32 `RunError` (int32 aError)

### 7.89.1 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.

### 7.89.2 Member Enumeration Documentation

#### 7.89.2.1 enum OsclActiveObject::OsclActivePriority

Scheduling priorities.

##### Enumerator:

- 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 EPriorityHigh.

### 7.89.3 Constructor & Destructor Documentation

#### 7.89.3.1 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.

#### 7.89.3.2 virtual OSCL\_IMPORT\_REF OsclActiveObject::~OsclActiveObject () [virtual]

Destructor.

## 7.89.4 Member Function Documentation

### 7.89.4.1 OSCL\_IMPORT\_REF void OsclActiveObject::AddToScheduler ()

Add this exec object to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

### 7.89.4.2 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](#).

Referenced by [OsclSocketRequestAO::Abort\(\)](#), and [OsclDNSRequestAO::Abort\(\)](#).

### 7.89.4.3 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](#).

### 7.89.4.4 OSCL\_IMPORT\_REF bool OsclActiveObject::IsBusy () const

Return true if this AO is pending, false otherwise.

### 7.89.4.5 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.

### 7.89.4.6 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.

### 7.89.4.7 OSCL\_IMPORT\_REF int32 OsclActiveObject::Priority () const

Return scheduling priority of this exec object.

**7.89.4.8 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](#).

Referenced by [OsclDNSRequestAO::Abort\(\)](#).

**7.89.4.9 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](#).

**7.89.4.10 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 added to any scheduler, or if the calling thread context does not match the scheduling thread.

**7.89.4.11 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.

**7.89.4.12 OSCL\_IMPORT\_REF void OsclActiveObject::SetStatus (int32)****7.89.4.13 OSCL\_IMPORT\_REF int32 OsclActiveObject::Status () const**

Request status access

**7.89.4.14 OSCL\_IMPORT\_REF OsclAOStatus& OsclActiveObject::StatusRef ()**

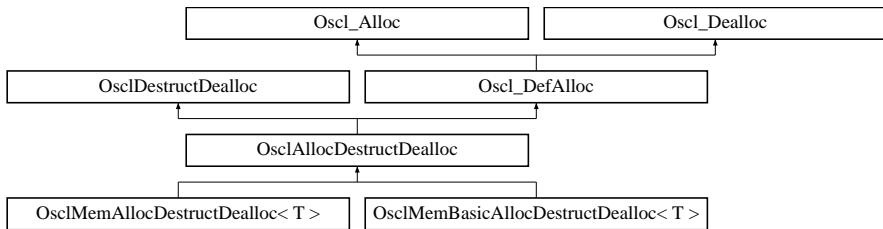
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_ao.h](#)

## 7.90 OsclAllocDestructDealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for OsclAllocDestructDealloc:



### Public Member Functions

- virtual [~OsclAllocDestructDealloc \(\)](#)

#### 7.90.1 Constructor & Destructor Documentation

##### 7.90.1.1 virtual OsclAllocDestructDealloc::~OsclAllocDestructDealloc () [inline, virtual]

The documentation for this class was generated from the following file:

- [oscl\\_defalloc.h](#)

## 7.91 OsclAOStatus Class Reference

```
#include <oscl_aostatus.h>
```

### Public Member Functions

- OSCL\_INLINE OsclAOStatus ()
- OSCL\_INLINE OsclAOStatus (int32 aStatus)
- OSCL\_INLINE int32 operator= (int32 aStatus)
- OSCL\_INLINE int32 operator== (int32 aStatus) const
- OSCL\_INLINE int32 operator!= (int32 aStatus) const
- OSCL\_INLINE int32 operator>= (int32 aStatus) const
- OSCL\_INLINE int32 operator<= (int32 aStatus) const
- OSCL\_INLINE int32 operator> (int32 aStatus) const
- OSCL\_INLINE int32 operator< (int32 aStatus) const
- OSCL\_INLINE int32 Value () const

#### 7.91.1 Constructor & Destructor Documentation

7.91.1.1 OSCL\_INLINE OsclAOStatus::OsclAOStatus ()

7.91.1.2 OSCL\_INLINE OsclAOStatus::OsclAOStatus (int32 *aStatus*)

#### 7.91.2 Member Function Documentation

7.91.2.1 OSCL\_INLINE int32 OsclAOStatus::operator!= (int32 *aStatus*) const

7.91.2.2 OSCL\_INLINE int32 OsclAOStatus::operator< (int32 *aStatus*) const

7.91.2.3 OSCL\_INLINE int32 OsclAOStatus::operator<= (int32 *aStatus*) const

7.91.2.4 OSCL\_INLINE int32 OsclAOStatus::operator= (int32 *aStatus*)

7.91.2.5 OSCL\_INLINE int32 OsclAOStatus::operator== (int32 *aStatus*) const

7.91.2.6 OSCL\_INLINE int32 OsclAOStatus::operator> (int32 *aStatus*) const

7.91.2.7 OSCL\_INLINE int32 OsclAOStatus::operator>= (int32 *aStatus*) const

7.91.2.8 OSCL\_INLINE int32 OsclAOStatus::Value () const

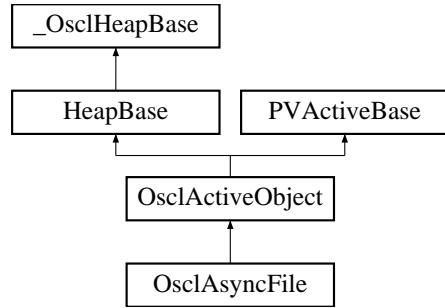
The documentation for this class was generated from the following file:

- [oscl\\_aostatus.h](#)

## 7.92 OsclAsyncFile Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclAsyncFile:



### Public Member Functions

- [~OsclAsyncFile \(\)](#)
- [int32 Open \(const oscl\\_wchar \\*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\\_FileServer &fileserv\)](#)
- [int32 Open \(const char \\*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\\_FileServer &fileserv\)](#)
- [int32 Seek \(TOsclFileOffset offset, Oscl\\_File::seek\\_type origin\)](#)
- [TOsclFileOffset Tell \(\)](#)
- [uint32 Read \(OsclAny \\*aBuffer1, uint32 aDataSize, uint32 aNumElements\)](#)
- [int32 EndOfFile \(\)](#)
- [TOsclFileOffset Size \(\)](#)
- [int32 Close \(\)](#)
- [uint32 Write \(const OsclAny \\*aBuffer1, uint32 aDataSize, uint32 aNumElements\)](#)
- [uint32 Flush \(\)](#)

### Static Public Member Functions

- [static OsclAsyncFile \\* NewL \(OsclNativeFile &aAsyncFile, int32 aCacheSize, PVLogger \\*\)](#)
- [static void Delete \(OsclAsyncFile \\*\)](#)

### Data Fields

- [uint32 iNumOfRun](#)
- [uint32 iNumOfRunErr](#)

#### 7.92.1 Detailed Description

[OsclAsyncFile](#)

## 7.92.2 Constructor & Destructor Documentation

### 7.92.2.1 OsclAsyncFile::~OsclAsyncFile ()

Destructor.

## 7.92.3 Member Function Documentation

### 7.92.3.1 int32 OsclAsyncFile::Close ()

### 7.92.3.2 static void OsclAsyncFile::Delete (OsclAsyncFile \*) [static]

### 7.92.3.3 int32 OsclAsyncFile::EndOfFile ()

### 7.92.3.4 uint32 OsclAsyncFile::Flush () [inline]

### 7.92.3.5 static OsclAsyncFile\* OsclAsyncFile::NewL (OsclNativeFile & aSyncFile, int32 aCacheSize, PVLogger \*) [static]

Two-phased constructor.

#### Parameters

*aSyncFile*,: open handle for async file read. Note: it is the caller's job to open/close this file handle.

*aSyncFile*,: duplicate open handle for sync file read. Note: it is the caller's job to open this file handle, but this class will close the handle.

*aCacheSize*,: size of one of the individual cache buffers. The total cached data size will be larger, since multiple buffers are used.

*aStartAsyncRead*,: When true, async file read will start immediately. When false, read will not begin until StartAsyncRead is called.

### 7.92.3.6 int32 OsclAsyncFile::Open (const char \*filename, uint32 mode, const OsclNativeFileParams & params, Oscl\_FileServer & fileserv)

### 7.92.3.7 int32 OsclAsyncFile::Open (const oscl\_wchar \*filename, uint32 mode, const OsclNativeFileParams & params, Oscl\_FileServer & fileserv)

### 7.92.3.8 uint32 OsclAsyncFile::Read (OsclAny \*aBuffer1, uint32 aDataSize, uint32 aNumElements)

### 7.92.3.9 int32 OsclAsyncFile::Seek (TOsclFileOffset offset, Oscl\_File::seek\_type origin)

### 7.92.3.10 TOsclFileOffset OsclAsyncFile::Size ()

### 7.92.3.11 TOsclFileOffset OsclAsyncFile::Tell ()

### 7.92.3.12 uint32 OsclAsyncFile::Write (const OsclAny \*aBuffer1, uint32 aDataSize, uint32 aNumElements) [inline]

References OSCL\_UNUSED\_ARG.

## 7.92.4 Field Documentation

### 7.92.4.1 uint32 OsclAsyncFile::iNumOfRun

### 7.92.4.2 uint32 OsclAsyncFile::iNumOfRunErr

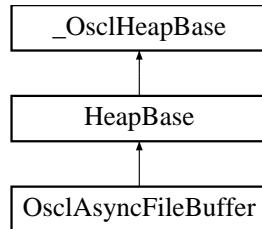
The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 7.93 OsclAsyncFileBuffer Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclAsyncFileBuffer:



### Public Member Functions

- [`~OsclAsyncFileBuffer \(\)`](#)
- [`void CleanInUse \(\)`](#)
- [`void SetInUse \(\)`](#)
- [`bool IsInUse \(\)`](#)
- [`bool IsValid \(\)`](#)
- [`TOsclFileOffset Offset \(\)`](#)
- [`void SetOffset \(TOsclFileOffset aOffset\)`](#)
- [`int32 Length \(\)`](#)
- [`bool HasThisOffset \(TOsclFileOffset aOffset\)`](#)
- [`int32 Id \(\)`](#)
- [`OsclBuf \* Buffer \(\)`](#)
- [`void UpdateData \(\)`](#)
- [`void StartAsyncRead \(bool aStartAsyncRead\)`](#)

### Static Public Member Functions

- [`static OsclAsyncFileBuffer \* NewL \(int32 aBufferSize, int32 aId\)`](#)

#### 7.93.1 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.

### 7.93.2 Constructor & Destructor Documentation

7.93.2.1 OsclAsyncFileBuffer::~OsclAsyncFileBuffer ()

### 7.93.3 Member Function Documentation

7.93.3.1 OsclBuf\* OsclAsyncFileBuffer::Buffer ()

7.93.3.2 void OsclAsyncFileBuffer::CleanInUse () [inline]

7.93.3.3 bool OsclAsyncFileBuffer::HasThisOffset (TOsclFileOffset *aOffset*)

7.93.3.4 int32 OsclAsyncFileBuffer::Id () [inline]

7.93.3.5 bool OsclAsyncFileBuffer::IsInUse () [inline]

7.93.3.6 bool OsclAsyncFileBuffer::IsValid () [inline]

7.93.3.7 int32 OsclAsyncFileBuffer::Length () [inline]

7.93.3.8 static OsclAsyncFileBuffer\* OsclAsyncFileBuffer::NewL (int32 *aBufferSize*, int32 *aId*)  
[static]

7.93.3.9 TOsclFileOffset OsclAsyncFileBuffer::Offset () [inline]

7.93.3.10 void OsclAsyncFileBuffer::SetInUse () [inline]

7.93.3.11 void OsclAsyncFileBuffer::SetOffset (TOsclFileOffset *aOffset*) [inline]

7.93.3.12 void OsclAsyncFileBuffer::StartAsyncRead (bool *aStartAsyncRead*)

7.93.3.13 void OsclAsyncFileBuffer::UpdateData ()

The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 7.94 OsclAuditCB Class Reference

```
#include <oscl_mem.h>
```

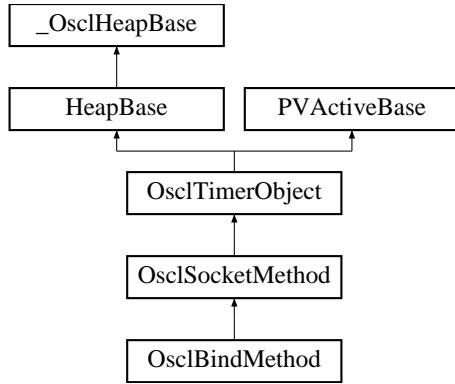
The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 7.95 OsclBindMethod Class Reference

```
#include <oscl_socket_bind.h>
```

Inheritance diagram for OsclBindMethod:



### Public Member Functions

- [~OsclBindMethod \(\)](#)
- [TPVSocketEvent Bind \(OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [OsclBindRequest \\* BindRequest \(\)](#)

### Static Public Member Functions

- static [OsclBindMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 7.95.1 Constructor & Destructor Documentation

##### 7.95.1.1 OsclBindMethod::~OsclBindMethod ()

#### 7.95.2 Member Function Documentation

##### 7.95.2.1 TPVSocketEvent OsclBindMethod::Bind (OsclNetworkAddress & aAddress, int32 aTimeout)

Referenced by OsclUDPSocketI::BindAsync(), and OsclTCPSocketI::BindAsync().

##### 7.95.2.2 OsclBindRequest\* OsclBindMethod::BindRequest () [inline]

References OsclSocketMethod::iSocketRequestAO.

##### 7.95.2.3 static OsclBindMethod\* OsclBindMethod::NewL (OsclIPSocketI &c) [static]

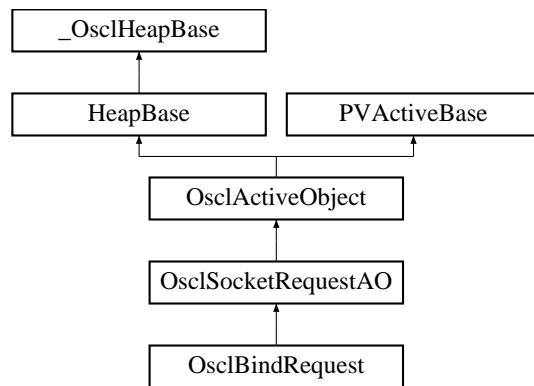
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_bind.h](#)

## 7.96 OsclBindRequest Class Reference

```
#include <oscl_socket_bind.h>
```

Inheritance diagram for OsclBindRequest:



### Public Member Functions

- [OsclBindRequest \(OsclSocketMethod &c\)](#)
- [void Bind \(OsclNetworkAddress &aAddress\)](#)

#### 7.96.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.96.2 Constructor & Destructor Documentation

**7.96.2.1 OsclBindRequest::OsclBindRequest (OsclSocketMethod & c) [inline]**

#### 7.96.3 Member Function Documentation

**7.96.3.1 void OsclBindRequest::Bind (OsclNetworkAddress & aAddress)**

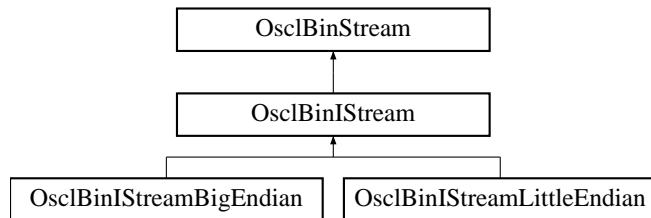
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_bind.h](#)

## 7.97 OsclBinIStream Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStream:



### Public Member Functions

- [OsclBinIStream \(\)](#)
- [~OsclBinIStream \(\)](#)
- [uint8 Read\\_uint8 \(\)](#)

*This method reads an unsigned short from the stream.*

- [OsclBinIStream & get \(int8 \\*data, int32 size\)](#)

*This method reads 'length' number of bytes from the stream and places them in 'data'.*

#### 7.97.1 Constructor & Destructor Documentation

[7.97.1.1 OsclBinIStream::OsclBinIStream \(\) \[inline\]](#)

[7.97.1.2 OsclBinIStream::~OsclBinIStream \(\) \[inline\]](#)

#### 7.97.2 Member Function Documentation

[7.97.2.1 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

[7.97.2.2 uint8 OsclBinIStream::Read\\_uint8 \(\)](#)

This method reads an unsigned short from the stream.

##### Returns

Unsigned short read from the stream.

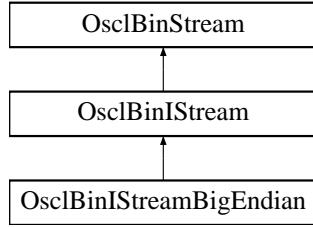
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 7.98 OsclBinIStreamBigEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStreamBigEndian:



### Public Member Functions

- [OsclBinIStreamBigEndian \(\)](#)
- [void Read \(int8 &data\)](#)
- [void Read \(uint8 &data\)](#)
- [void Read \(int16 &data\)](#)
- [void Read \(uint16 &data\)](#)
- [void Read \(int32 &data\)](#)
- [void Read \(uint32 &data\)](#)
- [OsclBinIStreamBigEndian & operator>> \(int8 &data\)](#)

*This method reads a int8 from the stream and stores it in 'data'.*

- [OsclBinIStream & operator>> \(uint8 &data\)](#)

*This method reads a uint8 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(int16 &data\)](#)

*This method reads a int16 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(uint16 &data\)](#)

*This method reads a uint16 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(int32 &data\)](#)

*This method reads a int32 from the stream and stores it in 'data'.*

- [OsclBinIStreamBigEndian & operator>> \(uint32 &data\)](#)

*This method reads a uint32 from the stream and stores it in 'data'.*

- [uint16 Read\\_uint16 \(\)](#)

*This method reads an unsigned short from the stream.*

- [uint32 Read\\_uint32 \(\)](#)

*This method reads an unsigned long from the stream.*

### 7.98.1 Constructor & Destructor Documentation

**7.98.1.1 OsclBinIStreamBigEndian::OsclBinIStreamBigEndian () [inline]**

### 7.98.2 Member Function Documentation

**7.98.2.1 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (uint32 & data)**

This method reads a uint32 from the stream and stores it in 'data'.

**7.98.2.2 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int32 & data)**

This method reads a int32 from the stream and stores it in 'data'.

**7.98.2.3 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (uint16 & data)**

This method reads a uint16 from the stream and stores it in 'data'.

**7.98.2.4 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int16 & data)**

This method reads a int16 from the stream and stores it in 'data'.

**7.98.2.5 OsclBinIStream& OsclBinIStreamBigEndian::operator>> (uint8 & data)**

This method reads a uint8 from the stream and stores it in 'data'.

**7.98.2.6 OsclBinIStreamBigEndian& OsclBinIStreamBigEndian::operator>> (int8 & data)**

This method reads a int8 from the stream and stores it in 'data'.

**7.98.2.7 void OsclBinIStreamBigEndian::Read (uint32 & data)**

**7.98.2.8 void OsclBinIStreamBigEndian::Read (int32 & data)**

**7.98.2.9 void OsclBinIStreamBigEndian::Read (uint16 & data)**

**7.98.2.10 void OsclBinIStreamBigEndian::Read (int16 & data)**

**7.98.2.11 void OsclBinIStreamBigEndian::Read (uint8 & data)**

**7.98.2.12 void OsclBinIStreamBigEndian::Read (int8 & data)**

**7.98.2.13 uint16 OsclBinIStreamBigEndian::Read\_uint16 ()**

This method reads an unsigned short from the stream.

#### Returns

Unsigned short read from the stream.

**7.98.2.14 uint32 OsclBinIStreamBigEndian::Read\_uint32 ()**

This method reads an unsigned long from the stream.

**Returns**

unsigned long read from the stream.

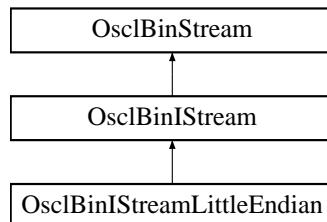
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 7.99 OsclBinIStreamLittleEndian Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinIStreamLittleEndian:



### Public Member Functions

- [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 Member Functions

- uint16 [Read\\_uint16 \(\)](#)
- uint32 [Read\\_uint32 \(\)](#)

### 7.99.1 Constructor & Destructor Documentation

**7.99.1.1 OsclBinIStreamLittleEndian::OsclBinIStreamLittleEndian () [inline]**

### 7.99.2 Member Function Documentation

**7.99.2.1 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint32 & data)**

This method reads a uint32 from the stream and stores it in 'data'.

**7.99.2.2 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int32 & data)**

This method reads a int32 from the stream and stores it in 'data'.

**7.99.2.3 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint16 & data)**

This method reads a uint16 from the stream and stores it in 'data'.

**7.99.2.4 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int16 & data)**

This method reads a int16 from the stream and stores it in 'data'.

**7.99.2.5 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (uint8 & data)**

This method reads a uint8 from the stream and stores it in 'data'.

**7.99.2.6 OsclBinIStreamLittleEndian& OsclBinIStreamLittleEndian::operator>> (int8 & data)**

This method reads a int8 from the stream and stores it in 'data'.

**7.99.2.7 uint16 OsclBinIStreamLittleEndian::Read\_uint16 () [protected]**

**7.99.2.8 uint32 OsclBinIStreamLittleEndian::Read\_uint32 () [protected]**

The documentation for this class was generated from the following file:

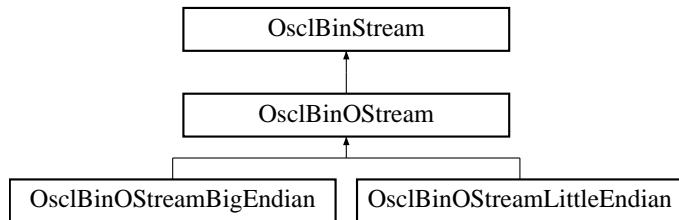
- [oscl\\_bin\\_stream.h](#)

## 7.100 OsclBinOStream Class Reference

Class [OsclBinOStream](#) implements the basic stream functions for an output stream.

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinOStream:



### Public Member Functions

- [OsclBinOStream \(\)](#)
- virtual [~OsclBinOStream \(\)](#)
- [OsclBinOStream & write \(const int8 \\*data, int32 size\)](#)

*This method writes 'length' number of bytes stored in 'data' to the stream.*

#### 7.100.1 Detailed Description

Class [OsclBinOStream](#) implements the basic stream functions for an output stream.

#### 7.100.2 Constructor & Destructor Documentation

**7.100.2.1 [OsclBinOStream::OsclBinOStream \(\) \[inline\]](#)**

**7.100.2.2 [virtual OsclBinOStream::~OsclBinOStream \(\) \[inline, virtual\]](#)**

#### 7.100.3 Member Function Documentation

**7.100.3.1 [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:

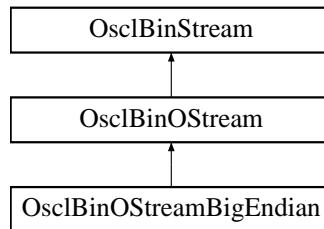
- [oscl\\_bin\\_stream.h](#)

## 7.101 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 Member Functions

- [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 Member Functions

- void [WriteUnsignedShort \(const uint16 data\)](#)
- void [WriteUnsignedLong \(const uint32 data\)](#)

#### 7.101.1 Detailed Description

Class [OsclBinOStreamBigEndian](#) implements a binary output stream using big endian byte ordering.

### 7.101.2 Constructor & Destructor Documentation

**7.101.2.1 OsclBinOStreamBigEndian::OsclBinOStreamBigEndian () [inline]**

### 7.101.3 Member Function Documentation

**7.101.3.1 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint32 & data)**

This method writes a uint32 from 'data' to the stream.

**7.101.3.2 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int32 & data)**

This method writes a int32 from 'data' to the stream.

**7.101.3.3 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint16 & data)**

This method writes a uint16 from 'data' to the stream.

**7.101.3.4 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int16 & data)**

This method writes a int16 from 'data' to the stream.

**7.101.3.5 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const uint8 & data)**

This method writes a uint8 from 'data' to the stream.

**7.101.3.6 OsclBinOStreamBigEndian& OsclBinOStreamBigEndian::operator<< (const int8 & data)**

This method writes a int8 from 'data' to the stream.

**7.101.3.7 void OsclBinOStreamBigEndian::WriteUnsignedLong (const uint32 data) [protected]**

**7.101.3.8 void OsclBinOStreamBigEndian::WriteUnsignedShort (const uint16 data) [protected]**

The documentation for this class was generated from the following file:

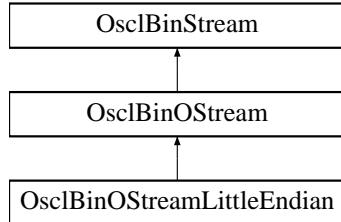
- [oscl\\_bin\\_stream.h](#)

## 7.102 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 Member Functions

- [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 Member Functions

- void [WriteUnsignedShort \(const uint16 data\)](#)  
*This method writes 'data' (unsigned short) to the stream.*
- void [WriteUnsignedLong \(const uint32 data\)](#)  
*This method writes 'data' (unsigned long) to the stream.*

#### 7.102.1 Detailed Description

Class [OsclBinOStreamLittleEndian](#) implements a binary output stream using little endian byte ordering.

### 7.102.2 Constructor & Destructor Documentation

**7.102.2.1 OsclBinOStreamLittleEndian::OsclBinOStreamLittleEndian () [inline]**

### 7.102.3 Member Function Documentation

**7.102.3.1 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint32 & data)**

This method writes a uint32 from 'data' to the stream.

**7.102.3.2 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int32 & data)**

This method writes a int32 from 'data' to the stream.

**7.102.3.3 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint16 & data)**

This method writes a uint16 from 'data' to the stream.

**7.102.3.4 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int16 & data)**

This method writes a int16 from 'data' to the stream.

**7.102.3.5 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const uint8 & data)**

This method writes a uint8 from 'data' to the stream.

**7.102.3.6 OsclBinOStreamLittleEndian& OsclBinOStreamLittleEndian::operator<< (const int8 & data)**

This method writes a int8 from 'data' to the stream.

**7.102.3.7 void OsclBinOStreamLittleEndian::WriteUnsignedLong (const uint32 data) [protected]**

This method writes 'data' (unsigned long) to the stream.

**7.102.3.8 void OsclBinOStreamLittleEndian::WriteUnsignedShort (const uint16 data) [protected]**

This method writes 'data' (unsigned short) to the stream.

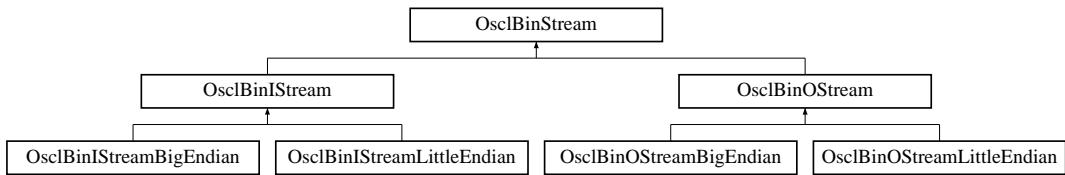
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 7.103 OsclBinStream Class Reference

```
#include <oscl_bin_stream.h>
```

Inheritance diagram for OsclBinStream:



### Public Member Functions

- [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 Member Functions

- bool [ReserveSpace](#) (uint32 size)
- bool [HaveRoomInCurrentBlock](#) (uint32 size)

## Protected Attributes

- [state\\_t state](#)
- uint8 \* [pBasePosition](#)
- uint8 \* [pPosition](#)
- uint32 [length](#)
- const [OsclMemoryFragment](#) \* [nextFragPtr](#)
- int [fragsLeft](#)
- const [OsclMemoryFragment](#) \* [firstFragPtr](#)
- int [numFrags](#)
- [OsclMemoryFragment](#) [specialFragBuffer](#)

### 7.103.1 Member Enumeration Documentation

#### 7.103.1.1 enum OsclBinStream::state\_t [protected]

Enumerator:

*GOOD\_STATE*

*EOF\_STATE*

*FAIL\_STATE*

### 7.103.2 Constructor & Destructor Documentation

#### 7.103.2.1 OsclBinStream::OsclBinStream () [inline]

### 7.103.3 Member Function Documentation

#### 7.103.3.1 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

#### 7.103.3.2 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

**7.103.3.3 bool OsclBinStream::eof ()**

This method determines if end of stream has been reached.

**Returns**

true if end of stream has been reached.

**7.103.3.4 bool OsclBinStream::fail ()**

This method determines if an error has occurred in the stream.

**Returns**

true if an error occurred in the stream.

**7.103.3.5 bool OsclBinStream::good ()**

This method determines if the stream is ok.

**Returns**

true if stream is ok.

**7.103.3.6 bool OsclBinStream::HaveRoomInCurrentBlock (uint32 *size*) [protected]****7.103.3.7 uint32 OsclBinStream::PositionInBlock ()**

This method returns the current stream position.

**Returns**

stream position.

**7.103.3.8 bool OsclBinStream::ReserveSpace (uint32 *size*) [protected]****7.103.3.9 void OsclBinStream::Seek (uint32 *absPosition*)**

This method seeks to the specified stream position.

**Returns**

Stream position.

**7.103.3.10 void OsclBinStream::seekFromCurrentPosition (int32 *offset*)**

This method seeks to the specified offset from the current location.

**Parameters**

*offset* from current stream location

**7.103.3.11 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.

**7.103.4 Field Documentation****7.103.4.1 const OsclMemoryFragment\* OsclBinStream::firstFragPtr [protected]****7.103.4.2 int OsclBinStream::fragsLeft [protected]****7.103.4.3 uint32 OsclBinStream::length [protected]****7.103.4.4 const OsclMemoryFragment\* OsclBinStream::nextFragPtr [protected]****7.103.4.5 int OsclBinStream::numFrags [protected]****7.103.4.6 uint8\* OsclBinStream::pBasePosition [protected]****7.103.4.7 uint8\* OsclBinStream::pPosition [protected]****7.103.4.8 OsclMemoryFragment OsclBinStream::specialFragBuffer [protected]****7.103.4.9 state\_t OsclBinStream::state [protected]**

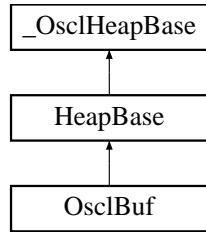
The documentation for this class was generated from the following file:

- [oscl\\_bin\\_stream.h](#)

## 7.104 OsclBuf Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclBuf:



### Public Member Functions

- [OsclBuf](#) (int32 size)
- int32 [Length](#) ()
- [OsclPtr Des](#) ()
- [OsclPtrC DesC](#) ()

### Static Public Member Functions

- static [OsclBuf \\* NewL](#) (int32 size)
- static void [Delete](#) ([OsclBuf](#) \*a)

### Data Fields

- uint8 \* [iBuffer](#)
- int32 [iMaxLength](#)
- int32 [iLength](#)

#### 7.104.1 Constructor & Destructor Documentation

**7.104.1.1 [OsclBuf::OsclBuf](#) (int32 size) [inline]**

#### 7.104.2 Member Function Documentation

**7.104.2.1 static void [OsclBuf::Delete](#) ([OsclBuf](#) \*a) [inline, static]**

References [iBuffer](#), [OSCL\\_DELETE](#), and [OSCL\\_FREE](#).

**7.104.2.2 [OsclPtr OsclBuf::Des](#) () [inline]**

References [iBuffer](#), [iLength](#), and [iMaxLength](#).

**7.104.2.3 OsclPtrC OsclBuf::DesC () [inline]**

References iBuffer, iLength, and iMaxLength.

**7.104.2.4 int32 OsclBuf::Length () [inline]**

References iLength.

**7.104.2.5 static OsclBuf\* OsclBuf::NewL (int32 size) [inline, static]**

References OsclError::Leave(), OSCL\_DELETE, OSCL\_MALLOC, OSCL\_NEW, and OsclErrNoMemory.

### 7.104.3 Field Documentation

**7.104.3.1 uint8\* OsclBuf::iBuffer**

Referenced by Delete(), Des(), and DesC().

**7.104.3.2 int32 OsclBuf::iLength**

Referenced by Des(), DesC(), and Length().

**7.104.3.3 int32 OsclBuf::iMaxLength**

Referenced by Des(), and DesC().

The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 7.105 Oscl\_File::OsclCacheObserver Class Reference

```
#include <oscl_file_io.h>
```

### Public Member Functions

- virtual [~OsclCacheObserver \(\)](#)
- virtual [OsclFileCacheBuffer \\* ChooseCurCache \(OsclFileCache &aContext, TOsclFileOffset aPos\)=0](#)

#### 7.105.1 Detailed Description

For defining a cache observer. Cache observer can implement customized cache schemes by replacing the SetCachePosition routine.

#### 7.105.2 Constructor & Destructor Documentation

**7.105.2.1 virtual Oscl\_File::OsclCacheObserver::~OsclCacheObserver () [inline, virtual]**

#### 7.105.3 Member Function Documentation

**7.105.3.1 virtual OsclFileCacheBuffer\* Oscl\_File::OsclCacheObserver::ChooseCurCache (OsclFileCache & aContext, TOsclFileOffset aPos) [pure virtual]**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_io.h](#)

## 7.106 OsclCompareLess< T > Class Template Reference

```
#include <oscl_priqueue.h>
```

### Public Member Functions

- int [compare](#) (T &a, T &b) const

```
template<class T> class OsclCompareLess< T >
```

#### 7.106.1 Member Function Documentation

**7.106.1.1 template<class T > int OsclCompareLess< T >::compare (T & a, T & b) const [inline]**

The documentation for this class was generated from the following file:

- [oscl\\_priqueue.h](#)

## 7.107 OsclComponentRegistry Class Reference

```
#include <oscl_registry_serv_impl.h>
```

### Public Member Functions

- [OsclComponentRegistry \(\)](#)
- [~OsclComponentRegistry \(\)](#)
- int32 [Register \(uint32 &alid, OSCL\\_String &, OsclComponentFactory\)](#)
- int32 [Unregister \(OSCL\\_String &\)](#)
- int32 [Unregister \(uint32\)](#)
- [OsclComponentFactory FindExact \(OSCL\\_String &\)](#)
- void [FindHierarchical \(OSCL\\_String &, Oscl\\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &\)](#)
- void [OpenSession \(\)](#)
- void [CloseSession \(\)](#)

### Data Fields

- [OsclComponentRegistryData iData](#)
- [OsclMutex iMutex](#)
- uint32 [iComponentIdCounter](#)
- uint32 [iNumSessions](#)

### 7.107.1 Detailed Description

Thread-safe singleton registry object.

## 7.107.2 Constructor & Destructor Documentation

7.107.2.1 `OsclComponentRegistry::OsclComponentRegistry ()`

7.107.2.2 `OsclComponentRegistry::~OsclComponentRegistry ()`

## 7.107.3 Member Function Documentation

7.107.3.1 `void OsclComponentRegistry::CloseSession ()`

7.107.3.2 `OsclComponentFactory OsclComponentRegistry::FindExact (OSCL_String &)`

7.107.3.3 `void OsclComponentRegistry::FindHierarchical (OSCL_String &, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > &)`

7.107.3.4 `void OsclComponentRegistry::OpenSession ()`

7.107.3.5 `int32 OsclComponentRegistry::Register (uint32 & aId, OSCL_String &, OsclComponentFactory)`

7.107.3.6 `int32 OsclComponentRegistry::Unregister (uint32)`

7.107.3.7 `int32 OsclComponentRegistry::Unregister (OSCL_String &)`

## 7.107.4 Field Documentation

7.107.4.1 `uint32 OsclComponentRegistry::iComponentIdCounter`

7.107.4.2 `OsclComponentRegistryData OsclComponentRegistry::iData`

7.107.4.3 `OsclMutex OsclComponentRegistry::iMutex`

7.107.4.4 `uint32 OsclComponentRegistry::iNumSessions`

The documentation for this class was generated from the following file:

- [oscl\\_registry\\_serv\\_impl.h](#)

## 7.108 OsclComponentRegistryData Class Reference

```
#include <oscl_registry_serv_impl.h>
```

### Public Member Functions

- `OsclComponentRegistryElement * Find (OSCL_String &, bool aExact)`

### Data Fields

- `Oscl_Vector< OsclComponentRegistryElement, OsclMemAllocator > iVec`

#### 7.108.1 Detailed Description

Registry

#### 7.108.2 Member Function Documentation

7.108.2.1 `OsclComponentRegistryElement* OsclComponentRegistryData::Find (OSCL_String &, bool aExact)`

#### 7.108.3 Field Documentation

7.108.3.1 `Oscl_Vector<OsclComponentRegistryElement, OsclMemAllocator> OsclComponentRegistryData::iVec`

The documentation for this class was generated from the following file:

- `oscl_registry_serv_impl.h`

## 7.109 OsclComponentRegistryElement Class Reference

```
#include <oscl_registry_serv_impl.h>
```

### Public Member Functions

- `OsclComponentRegistryElement (OSCL_String &, OsclComponentFactory)`
- `OsclComponentRegistryElement (const OsclComponentRegistryElement &)`
- `OsclComponentRegistryElement & operator= (const OsclComponentRegistryElement &src)`
- `~OsclComponentRegistryElement ()`
- `bool Match (OSCL_String &aStr, bool aExact)`

### Data Fields

- `OSCL_String * iId`
- `OsclComponentFactory iFactory`
- `uint32 iComponentId`

#### 7.109.1 Detailed Description

OS-independent declarations.

Data for each registered component.

#### 7.109.2 Constructor & Destructor Documentation

**7.109.2.1 OsclComponentRegistryElement::OsclComponentRegistryElement (OSCL\_String &, OsclComponentFactory)**

**7.109.2.2 OsclComponentRegistryElement::OsclComponentRegistryElement (const OsclComponentRegistryElement &)**

**7.109.2.3 OsclComponentRegistryElement::~OsclComponentRegistryElement ()**

#### 7.109.3 Member Function Documentation

**7.109.3.1 bool OsclComponentRegistryElement::Match (OSCL\_String & aStr, bool aExact)**

**7.109.3.2 OsclComponentRegistryElement& OsclComponentRegistryElement::operator= (const OsclComponentRegistryElement & src)**

#### 7.109.4 Field Documentation

**7.109.4.1 uint32 OsclComponentRegistryElement::iComponentId**

**7.109.4.2 OsclComponentFactory OsclComponentRegistryElement::iFactory**

**7.109.4.3 OSCL\_String\* OsclComponentRegistryElement::iId**

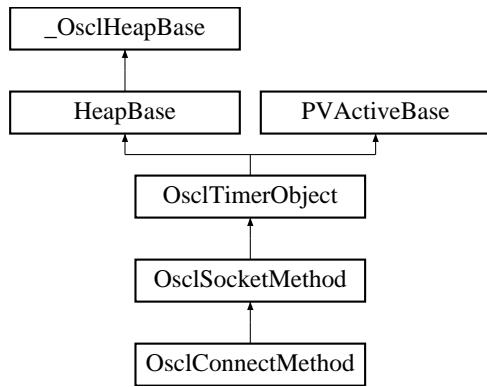
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_serv\\_impl.h](#)

## 7.110 OsclConnectMethod Class Reference

```
#include <oscl_socket_connect.h>
```

Inheritance diagram for OsclConnectMethod:



### Public Member Functions

- `~OsclConnectMethod ()`
- `TPVSocketEvent Connect (OsclNetworkAddress &aAddress, int32 aTimeout)`
- `OsclConnectRequest * ConnectRequest ()`

### Static Public Member Functions

- static `OsclConnectMethod * NewL (OsclIPSocketI &c)`

#### 7.110.1 Constructor & Destructor Documentation

##### 7.110.1.1 OsclConnectMethod::~OsclConnectMethod ()

#### 7.110.2 Member Function Documentation

##### 7.110.2.1 TPVSocketEvent OsclConnectMethod::Connect (OsclNetworkAddress & aAddress, int32 aTimeout)

Referenced by `OsclTCPSocketI::Connect()`.

##### 7.110.2.2 OsclConnectRequest\* OsclConnectMethod::ConnectRequest () [inline]

References `OsclSocketMethod::iSocketRequestAO`.

##### 7.110.2.3 static OsclConnectMethod\* OsclConnectMethod::NewL (OsclIPSocketI & c) [static]

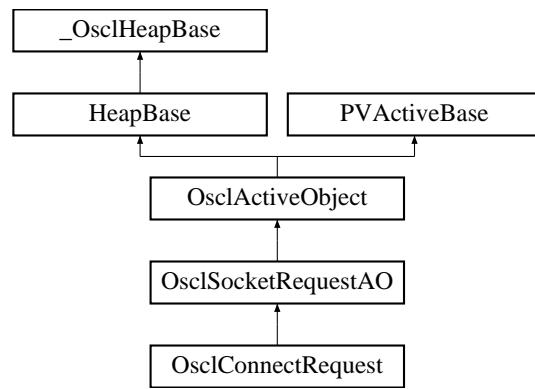
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_connect.h](#)

## 7.111 OsclConnectRequest Class Reference

```
#include <oscl_socket_connect.h>
```

Inheritance diagram for OsclConnectRequest:



### Public Member Functions

- [OsclConnectRequest \(OsclSocketMethod &c\)](#)
- void [Connect \(OsclNetworkAddress &aAddress\)](#)

#### 7.111.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.111.2 Constructor & Destructor Documentation

**7.111.2.1 OsclConnectRequest::OsclConnectRequest (OsclSocketMethod & c) [inline]**

#### 7.111.3 Member Function Documentation

**7.111.3.1 void OsclConnectRequest::Connect (OsclNetworkAddress & aAddress)**

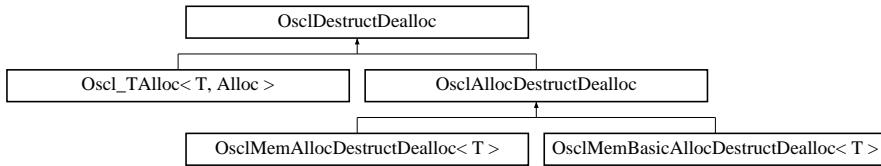
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_connect.h](#)

## 7.112 OsclDestructDealloc Class Reference

```
#include <oscl_defalloc.h>
```

Inheritance diagram for OsclDestructDealloc:



### Public Member Functions

- virtual [~OsclDestructDealloc \(\)](#)
- virtual void [destruct\\_and\\_dealloc \(OsclAny \\*ptr\)=0](#)

#### 7.112.1 Constructor & Destructor Documentation

**7.112.1.1 virtual OsclDestructDealloc::~OsclDestructDealloc () [inline, virtual]**

#### 7.112.2 Member Function Documentation

**7.112.2.1 virtual void OsclDestructDealloc::destruct\_and\_dealloc (OsclAny \*ptr) [pure virtual]**

Implemented in [Oscl\\_TAlloc< T, Alloc >](#), [OsclMemAllocDestructDealloc< T >](#), [OsclMemBasicAllocDestructDealloc< T >](#), [Oscl\\_TAlloc< entry\\_type, Alloc >](#), [Oscl\\_TAlloc< node\\_type, alloc\\_type >](#), [Oscl\\_TAlloc< char, alloc\\_type >](#), [Oscl\\_TAlloc< tag\\_base\\_unit, Alloc >](#), [Oscl\\_TAlloc< PVLogger, alloc\\_type >](#), and [Oscl\\_TAlloc< node\\_type, Alloc >](#).

Referenced by [OsclRefCounterMTDA< LockType >::removeRef\(\)](#), and [OsclRefCounterDA::removeRef\(\)](#).

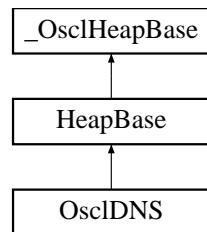
The documentation for this class was generated from the following file:

- [oscl\\_defalloc.h](#)

## 7.113 OsclDNS Class Reference

```
#include <oscl_dns.h>
```

Inheritance diagram for OsclDNS:



### Public Member Functions

- OSCL\_IMPORT\_REF ~OsclDNS ()
- OSCL\_IMPORT\_REF TPVDNSEvent GetHostByName (char \*name, OsclNetworkAddress &addr, int32 aTimeoutMsec=-1, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \*aAddressList=NULL)
- OSCL\_IMPORT\_REF void CancelGetHostByName ()

### Static Public Member Functions

- static OSCL\_IMPORT\_REF OsclDNS \* NewL (Oscl\_DefAlloc &alloc, OsclSocketServ &aServ, OsclDNSObserver &aObserver, uint32 aId)

### Friends

- class OsclDNSRequestAO

#### 7.113.1 Detailed Description

The DNS class

#### 7.113.2 Member Function Documentation

**7.113.2.1 static 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.

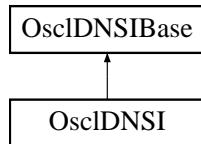
The documentation for this class was generated from the following file:

- [oscl\\_dns.h](#)

## 7.114 OsclDNSI Class Reference

```
#include <oscl_dns_imp_pv.h>
```

Inheritance diagram for OsclDNSI:



### Public Member Functions

- [`~OsclDNSI \(\)`](#)
- [`int32 Open \(OsclSocketServI &aServer\)`](#)
- [`int32 Close \(\)`](#)
- [`void GetHostByName \(GetHostByNameParam &, OsclDNSRequestAO &\)`](#)
- [`void GetHostByNameSuccess \(GetHostByNameParam &\)`](#)
- [`void GetNextHost \(OsclDNSRequestAO &\)`](#)
- [`void GetNextHostSuccess \(GetHostByNameParam &\)`](#)
- [`bool GetHostByNameResponseContainsAliasInfo \(\)`](#)

### Static Public Member Functions

- [`static OsclDNSI \* NewL \(Oscl\_DefAlloc &a\)`](#)

### Friends

- class [OsclDNSRequest](#)
- class [OsclGetHostByNameRequest](#)
- class [DNSRequestParam](#)

### 7.114.1 Detailed Description

[OsclDNSI](#), non-Symbian implementation

### 7.114.2 Constructor & Destructor Documentation

#### 7.114.2.1 [OsclDNSI::~OsclDNSI \(\)](#)

### 7.114.3 Member Function Documentation

#### 7.114.3.1 [int32 OsclDNSI::Close \(\) \[virtual\]](#)

Implements [OsclDNSIBase](#).

**7.114.3.2 void OsclDNSI::GetHostByName (GetHostNameParam &, OsclDNSRequestAO &) [virtual]**

Implements [OsclDNSIBase](#).

**7.114.3.3 bool OsclDNSI::GetHostByNameResponseContainsAliasInfo () [virtual]**

Implements [OsclDNSIBase](#).

**7.114.3.4 void OsclDNSI::GetHostByNameSuccess (GetHostNameParam &) [virtual]**

Implements [OsclDNSIBase](#).

**7.114.3.5 void OsclDNSI::GetNextHost (OsclDNSRequestAO &) [virtual]**

Implements [OsclDNSIBase](#).

**7.114.3.6 void OsclDNSI::GetNextHostSuccess (GetHostNameParam &) [virtual]**

Implements [OsclDNSIBase](#).

**7.114.3.7 static OsclDNSI\* OsclDNSI::NewL (Oscl\_DefAlloc & a) [static]**

**7.114.3.8 int32 OsclDNSI::Open (OsclSocketServI & aServer) [virtual]**

Implements [OsclDNSIBase](#).

## 7.114.4 Friends And Related Function Documentation

**7.114.4.1 friend class DNSRequestParam [friend]**

**7.114.4.2 friend class OsclDNSRequest [friend]**

Reimplemented from [OsclDNSIBase](#).

**7.114.4.3 friend class OsclGetHostByNameRequest [friend]**

Reimplemented from [OsclDNSIBase](#).

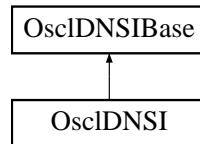
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_imp\\_pv.h](#)

## 7.115 OsclDNSIBase Class Reference

```
#include <oscl_dns_imp_base.h>
```

Inheritance diagram for OsclDNSIBase:



### Public Member Functions

- virtual ~OsclDNSIBase ()
- virtual int32 Open (OsclSocketServI &aServer)=0
- virtual int32 Close ()=0
- virtual void GetHostByName (GetHostByNameParam &, OsclDNSRequestAO &)=0
- virtual void GetHostByNameSuccess (GetHostByNameParam &)=0
- virtual bool GetHostByNameResponseContainsAliasInfo ()=0
- virtual void GetNextHost (OsclDNSRequestAO &)=0
- virtual void GetNextHostSuccess (GetHostByNameParam &)=0
- void CancelFxn (TPVDNSFxn)

### Protected Member Functions

- OsclDNSIBase (Oscl\_DefAlloc &a)
- virtual bool IsReady (OsclDNSRequestAO &aObject)=0
- virtual void CancelGetHostByName ()=0

### Protected Attributes

- Oscl\_DefAlloc & iAlloc
- OsclSocketServI \* iSocketServ

### Friends

- class OsclDNSRequest
- class OsclGetHostByNameRequest

#### 7.115.1 Detailed Description

[OsclDNSIBase](#) is a common base class for all implementations.

### 7.115.2 Constructor & Destructor Documentation

7.115.2.1 `virtual OsclDNSIBase::~OsclDNSIBase () [virtual]`

7.115.2.2 `OsclDNSIBase::OsclDNSIBase (Oscl_DefAlloc & a) [protected]`

### 7.115.3 Member Function Documentation

7.115.3.1 `void OsclDNSIBase::CancelFxn (TPVDNSFxn)`

7.115.3.2 `virtual void OsclDNSIBase::CancelGetHostName () [protected, pure virtual]`

7.115.3.3 `virtual int32 OsclDNSIBase::Close () [pure virtual]`

Implemented in [OsclDNSI](#).

7.115.3.4 `virtual void OsclDNSIBase::GetHostName (GetHostNameParam &, OsclDNSRequestAO &) [pure virtual]`

Implemented in [OsclDNSI](#).

7.115.3.5 `virtual bool OsclDNSIBase::GetHostNameResponseContainsAliasInfo () [pure virtual]`

Implemented in [OsclDNSI](#).

7.115.3.6 `virtual void OsclDNSIBase::GetHostNameSuccess (GetHostNameParam &) [pure virtual]`

Implemented in [OsclDNSI](#).

7.115.3.7 `virtual void OsclDNSIBase::GetNextHost (OsclDNSRequestAO &) [pure virtual]`

Implemented in [OsclDNSI](#).

7.115.3.8 `virtual void OsclDNSIBase::GetNextHostSuccess (GetHostNameParam &) [pure virtual]`

Implemented in [OsclDNSI](#).

7.115.3.9 `virtual bool OsclDNSIBase::IsReady (OsclDNSRequestAO & aObject) [protected, pure virtual]`

7.115.3.10 `virtual int32 OsclDNSIBase::Open (OsclSocketServI & aServer) [pure virtual]`

Implemented in [OsclDNSI](#).

## 7.115.4 Friends And Related Function Documentation

### 7.115.4.1 friend class OsclDNSRequest [friend]

Reimplemented in [OsclDNSI](#).

### 7.115.4.2 friend class OsclGetHostByNameRequest [friend]

Reimplemented in [OsclDNSI](#).

## 7.115.5 Field Documentation

### 7.115.5.1 Oscl\_DefAlloc& OsclDNSIBase::iAlloc [protected]

### 7.115.5.2 OsclSocketServI\* OsclDNSIBase::iSocketServ [protected]

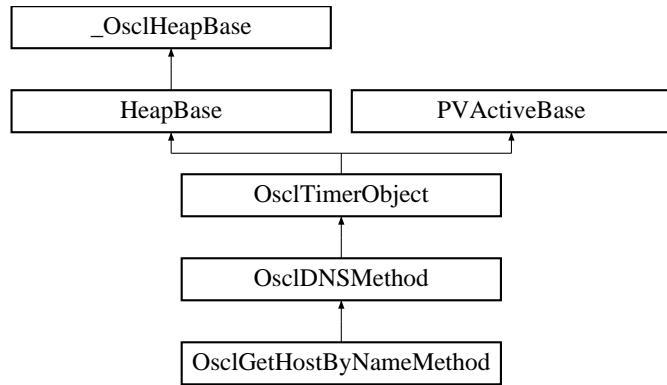
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_imp\\_base.h](#)

## 7.116 OsclDNSMethod Class Reference

```
#include <oscl_dns_method.h>
```

Inheritance diagram for OsclDNSMethod:



### Public Member Functions

- [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 Member Functions

- void [ConstructL \(OsclDNSObserver \\*aObserver, OsclDNSRequestAO \\*aAO, uint32 aId\)](#)
- bool [StartMethod \(int32 aTimeoutMsec\)](#)
- void [MethodDone \(\)](#)

### Protected Attributes

- [OsclDNSRequestAO \\* iDNSRequestAO](#)

#### 7.116.1 Detailed Description

This is the base class for all socket methods. It provides the timeout on socket requests.

## 7.116.2 Constructor & Destructor Documentation

**7.116.2.1 OsclDNSMethod::OsclDNSMethod (Oscl\_DefAlloc & *a*, const char \* *name*, TPVDNSFxn *fxn*) [inline]**

References PVLogger::GetLoggerObject(), and iLogger.

## 7.116.3 Member Function Documentation

**7.116.3.1 void OsclDNSMethod::Abort ()**

**7.116.3.2 void OsclDNSMethod::AbortAll ()**

**7.116.3.3 void OsclDNSMethod::CancelMethod ()**

**7.116.3.4 void OsclDNSMethod::ConstructL (OsclDNSObserver \* *aObserver*, OsclDNSRequestAO \* *aAO*, uint32 *aId*) [protected]**

**7.116.3.5 void OsclDNSMethod::MethodDone () [protected]**

**7.116.3.6 void OsclDNSMethod::Run () [virtual]**

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's [Run\(\)](#) function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

**7.116.3.7 bool OsclDNSMethod::StartMethod (int32 *aTimeoutMsec*) [protected]**

## 7.116.4 Field Documentation

**7.116.4.1 Oscl\_DefAlloc& OsclDNSMethod::iAlloc**

**7.116.4.2 TPVDNSFx OsclDNSMethod::iDNSFx**

**7.116.4.3 OsclDNSObserver\* OsclDNSMethod::iDNSObserver**

**7.116.4.4 OsclDNSRequestAO\* OsclDNSMethod::iDNSRequestAO [protected]**

**7.116.4.5 uint32 OsclDNSMethod::iId**

**7.116.4.6 PVLogger\* OsclDNSMethod::iLogger**

Referenced by OsclDNSRequestAO::ConstructL(), and OsclDNSMethod().

The documentation for this class was generated from the following file:

- [oscl\\_dns\\_method.h](#)

## 7.117 OsclDNSObserver Class Reference

```
#include <oscl_dns.h>
```

### Public Member Functions

- virtual OSCL\_IMPORT\_REF void [HandleDNSEvent](#) (int32 aId, [TPVDNSFxn](#) aFxn, [TPVDNSEvent](#) aEvent, int32 aError)=0
- virtual [~OsclDNSObserver](#) ()

#### 7.117.1 Detailed Description

DNS event observer. The client implements this to get asynchronous command completion.

#### 7.117.2 Member Function Documentation

##### 7.117.2.1 virtual OSCL\_IMPORT\_REF void OsclDNSObserver::HandleDNSEvent (int32 *aId*, *TPVDNSFxn* *aFxn*, *TPVDNSEvent* *aEvent*, int32 *aError*) [pure virtual]

DNS Event callback.

###### Parameters

- aId*:** The ID that was supplied when the DNS object was created.
- aEvent*:** Function completion event. Will be EPVDNSSuccess, EPVDNSTimeout, or EPVDNSFailure.
- aError*:** When the event is EPVDNSFailure, this may contain a platform-specific error code, or zero if none is available.

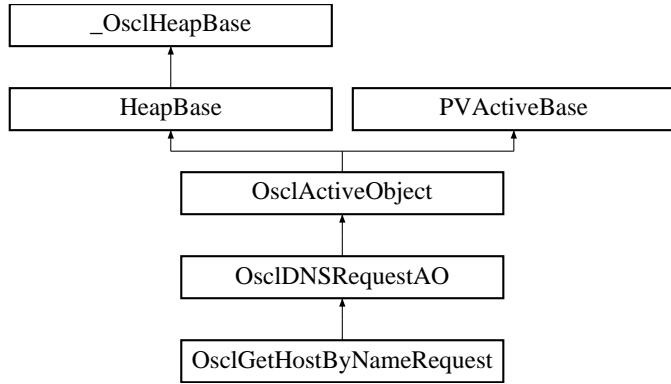
The documentation for this class was generated from the following file:

- [oscl\\_dns.h](#)

## 7.118 OsclDNSRequestAO Class Reference

```
#include <oscl_dns_method.h>
```

Inheritance diagram for OsclDNSRequestAO:



### Protected Member Functions

- `OsclDNSRequestAO` (const char \*name)
- void `ConstructL` (`OsclDNSI` \*aDNS, `OsclDNSMethod` \*aMethod)
- void `Abort` ()
- void `NewRequest` ()
- void `RequestDone` ()
- int `GetSocketError` ()
- `OsclSocketServI` \* `Serv` ()
- void `DoCancel` ()
- void `Run` ()
- virtual void `Success` ()
- virtual void `Failure` ()
- virtual void `Cancelled` ()

### Protected Attributes

- `OsclDNSI` \* `iDNSI`
- `OsclDNSMethod` \* `iDNSMethod`
- int32 `iSocketError`
- `PVLogger` \* `iLogger`

### Friends

- class `OsclDNSI`
- class `OsclDNSMethod`
- class `OsclDNSRequest`
- class `GetHostNameParam`

### 7.118.1 Detailed Description

This is the base class for all requests to the socket server.

### 7.118.2 Constructor & Destructor Documentation

**7.118.2.1 OsclDNSRequestAO::OsclDNSRequestAO (const char \* *name*) [inline, protected]**

### 7.118.3 Member Function Documentation

**7.118.3.1 void OsclDNSRequestAO::Abort () [inline, protected]**

References OsclActiveObject::Cancel(), and OsclActiveObject::RemoveFromScheduler().

**7.118.3.2 virtual void OsclDNSRequestAO::Cancelled () [inline, protected, virtual]**

**7.118.3.3 void OsclDNSRequestAO::ConstructL (OsclDNSI \* *aDNS*, OsclDNSMethod \* *aMethod*) [inline, protected]**

References iDNSI, iDNSMethod, OsclDNSMethod::iLogger, iLogger, OsclError::Leave(), and OsclErrorGeneral.

**7.118.3.4 void OsclDNSRequestAO::DoCancel () [protected, virtual]**

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will complete the request. If any additional action is needed, the derived class may override this. If the derived class does override DoCancel, it must complete the request.

Reimplemented from [OsclActiveObject](#).

**7.118.3.5 virtual void OsclDNSRequestAO::Failure () [inline, protected, virtual]**

**7.118.3.6 int OsclDNSRequestAO::GetSocketError () [protected]**

**7.118.3.7 void OsclDNSRequestAO::NewRequest () [protected]**

**7.118.3.8 void OsclDNSRequestAO::RequestDone () [protected]**

**7.118.3.9 void OsclDNSRequestAO::Run () [protected, virtual]**

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's [Run\(\)](#) function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request

2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls [ExecError\(\)](#) to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

**7.118.3.10 OsclSocketServI\* OsclDNSRequestAO::Serv () [protected]**

**7.118.3.11 virtual void OsclDNSRequestAO::Success () [inline, protected, virtual]**

## 7.118.4 Friends And Related Function Documentation

**7.118.4.1 friend class GetHostNameParam [friend]**

**7.118.4.2 friend class OsclDNSI [friend]**

**7.118.4.3 friend class OsclDNSMethod [friend]**

**7.118.4.4 friend class OsclDNSRequest [friend]**

## 7.118.5 Field Documentation

**7.118.5.1 OsclDNSI\* OsclDNSRequestAO::iDNSI [protected]**

Referenced by [ConstructL\(\)](#).

**7.118.5.2 OsclDNSMethod\* OsclDNSRequestAO::iDNSMethod [protected]**

Referenced by [ConstructL\(\)](#).

**7.118.5.3 PVLogger\* OsclDNSRequestAO::iLogger [protected]**

Referenced by [ConstructL\(\)](#).

**7.118.5.4 int32 OsclDNSRequestAO::iSocketError [protected]**

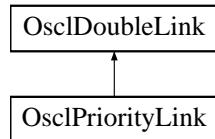
The documentation for this class was generated from the following file:

- [oscl\\_dns\\_method.h](#)

## 7.119 OsclDoubleLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleLink:



### Public Member Functions

- [OsclDoubleLink \(\)](#)
- [void Remove \(\)](#)
- [void InsertAfter \(OsclDoubleLink \\*aLink\)](#)
- [void InsertBefore \(OsclDoubleLink \\*aLink\)](#)

### Data Fields

- [OsclDoubleLink \\* iNext](#)
- [OsclDoubleLink \\* iPrev](#)

#### 7.119.1 Constructor & Destructor Documentation

##### 7.119.1.1 [OsclDoubleLink::OsclDoubleLink \(\) \[inline\]](#)

#### 7.119.2 Member Function Documentation

##### 7.119.2.1 [void OsclDoubleLink::InsertAfter \(OsclDoubleLink \\* aLink\)](#)

##### 7.119.2.2 [void OsclDoubleLink::InsertBefore \(OsclDoubleLink \\* aLink\)](#)

##### 7.119.2.3 [void OsclDoubleLink::Remove \(\)](#)

#### 7.119.3 Field Documentation

##### 7.119.3.1 [OsclDoubleLink\\* OsclDoubleLink::iNext](#)

Referenced by [OsclDoubleRunner< T >::operator++\(\)](#), and [OsclDoubleRunner< T >::SetToHead\(\)](#).

##### 7.119.3.2 [OsclDoubleLink\\* OsclDoubleLink::iPrev](#)

Referenced by [OsclDoubleRunner< T >::SetToTail\(\)](#).

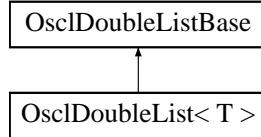
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.120 OsclDoubleList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleList< T >:



### Public Member Functions

- OSCL\_INLINE [OsclDoubleList\(\)](#)
- OSCL\_INLINE [OsclDoubleList\(int32 anOffset\)](#)
- OSCL\_INLINE void [InsertHead\(T &aRef\)](#)
- OSCL\_INLINE void [InsertTail\(T &aRef\)](#)
- OSCL\_INLINE bool [IsHead\(const T \\*aPtr\) const](#)
- OSCL\_INLINE bool [IsTail\(const T \\*aPtr\) const](#)
- OSCL\_INLINE T \* [Head\(\) const](#)
- OSCL\_INLINE T \* [Tail\(\) const](#)

```
template<class T> class OsclDoubleList< T >
```

#### 7.120.1 Constructor & Destructor Documentation

**7.120.1.1 template<class T > OSCL\_INLINE OsclDoubleList< T >::OsclDoubleList ()**

**7.120.1.2 template<class T > OSCL\_INLINE OsclDoubleList< T >::OsclDoubleList (int32  
anOffset)**

#### 7.120.2 Member Function Documentation

**7.120.2.1 template<class T > OSCL\_INLINE T\* OsclDoubleList< T >::Head () const**

**7.120.2.2 template<class T > OSCL\_INLINE void OsclDoubleList< T >::InsertHead (T & aRef)**

**7.120.2.3 template<class T > OSCL\_INLINE void OsclDoubleList< T >::InsertTail (T & aRef)**

**7.120.2.4 template<class T > OSCL\_INLINE bool OsclDoubleList< T >::IsHead (const T \* aPtr)  
const**

**7.120.2.5 template<class T > OSCL\_INLINE bool OsclDoubleList< T >::IsTail (const T \* aPtr)  
const**

**7.120.2.6 template<class T > OSCL\_INLINE T\* OsclDoubleList< T >::Tail () const**

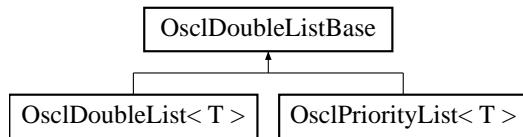
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.121 OsclDoubleListBase Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclDoubleListBase:



### Public Member Functions

- bool [IsEmpty \(\) const](#)
- void [SetOffset \(int32 anOffset\)](#)
- void [Reset \(\)](#)
- [OsclDoubleLink \\* getHead \(\)](#)
- int32 [getOffset \(\)](#)

### Protected Member Functions

- [OsclDoubleListBase \(\)](#)
- [OsclDoubleListBase \(int32 anOffset\)](#)
- void [InsertHead \(OsclAny \\*aPtr\)](#)
- void [InsertTail \(OsclAny \\*aPtr\)](#)
- void [Insert \(OsclAny \\*aPtr\)](#)

### Protected Attributes

- [OsclDoubleLink iHead](#)
- int32 [iOffset](#)

#### 7.121.1 Constructor & Destructor Documentation

**7.121.1.1 OsclDoubleListBase::OsclDoubleListBase () [protected]**

**7.121.1.2 OsclDoubleListBase::OsclDoubleListBase (int32 *anOffset*) [protected]**

#### 7.121.2 Member Function Documentation

**7.121.2.1 OsclDoubleLink\* OsclDoubleListBase::getHead () [inline]**

References iHead.

Referenced by OsclDoubleRunner< T >::OsclDoubleRunner().

**7.121.2.2 int32 OsclDoubleListBase::getOffset () [inline]**

References iOffset.

Referenced by OsclDoubleRunner< T >::OsclDoubleRunner().

**7.121.2.3 void OsclDoubleListBase::Insert (OsclAny \* aPtr) [protected]****7.121.2.4 void OsclDoubleListBase::InsertHead (OsclAny \* aPtr) [protected]****7.121.2.5 void OsclDoubleListBase::InsertTail (OsclAny \* aPtr) [protected]****7.121.2.6 bool OsclDoubleListBase::IsEmpty () const****7.121.2.7 void OsclDoubleListBase::Reset ()****7.121.2.8 void OsclDoubleListBase::SetOffset (int32 anOffset)**

### 7.121.3 Field Documentation

**7.121.3.1 OsclDoubleLink OsclDoubleListBase::iHead [protected]**

Referenced by getHead().

**7.121.3.2 int32 OsclDoubleListBase::iOffset [protected]**

Referenced by getOffset().

The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.122 OsclDoubleRunner< T > Class Template Reference

```
#include <oscl_double_list.h>
```

### Public Member Functions

- [OsclDoubleRunner \(OsclDoubleListBase &aQue\)](#)
- void [Set \(T &aLink\)](#)
- [operator T \\* \(\)](#)
- T \* [operator++ \(int\)](#)
- T \* [operator-- \(int\)](#)
- void [SetToHead \(\)](#)
- void [SetToTail \(\)](#)

### Protected Attributes

- int32 [iOffset](#)
- [OsclDoubleLink \\* iHead](#)
- [OsclDoubleLink \\* iNext](#)

**template<class T> class OsclDoubleRunner< T >**

#### 7.122.1 Constructor & Destructor Documentation

##### 7.122.1.1 template<class T> OsclDoubleRunner< T >::OsclDoubleRunner (OsclDoubleListBase & aQue) [inline]

References OsclDoubleListBase::getHead(), OsclDoubleListBase::getOffset(), OsclDoubleRunner< T >::iHead, OsclDoubleRunner< T >::iNext, OsclDoubleRunner< T >::iOffset, and NULL.

#### 7.122.2 Member Function Documentation

##### 7.122.2.1 template<class T> OsclDoubleRunner< T >::operator T \* () [inline]

References OsclDoubleRunner< T >::iNext, OsclDoubleRunner< T >::iOffset, NULL, and OsclPtrSub().

##### 7.122.2.2 template<class T> T\* OsclDoubleRunner< T >::operator++ (int) [inline]

References OsclDoubleLink::iNext, OsclDoubleRunner< T >::iNext, OsclDoubleRunner< T >::iOffset, NULL, and OsclPtrSub().

##### 7.122.2.3 template<class T> T\* OsclDoubleRunner< T >::operator-- (int)

##### 7.122.2.4 template<class T> void OsclDoubleRunner< T >::Set (T & aLink) [inline]

References OsclDoubleRunner< T >::iNext, OsclDoubleRunner< T >::iOffset, and OsclPtrAdd().

**7.122.2.5 template<class T> void OsclDoubleRunner< T >::SetToHead () [inline]**

References OsclDoubleRunner< T >::iHead, OsclDoubleLink::iNext, and OsclDoubleRunner< T >::iNext.

**7.122.2.6 template<class T> void OsclDoubleRunner< T >::SetToTail () [inline]**

References OsclDoubleRunner< T >::iHead, OsclDoubleRunner< T >::iNext, and OsclDoubleLink::iPrev.

### 7.122.3 Field Documentation

**7.122.3.1 template<class T> OsclDoubleLink\* OsclDoubleRunner< T >::iHead [protected]**

Referenced by OsclDoubleRunner< T >::OsclDoubleRunner(), OsclDoubleRunner< T >::SetToHead(), and OsclDoubleRunner< T >::SetToTail().

**7.122.3.2 template<class T> OsclDoubleLink\* OsclDoubleRunner< T >::iNext [protected]**

Referenced by OsclDoubleRunner< T >::operator T \*(), OsclDoubleRunner< T >::operator++(), OsclDoubleRunner< T >::OsclDoubleRunner(), OsclDoubleRunner< T >::Set(), OsclDoubleRunner< T >::SetToHead(), and OsclDoubleRunner< T >::SetToTail().

**7.122.3.3 template<class T> int32 OsclDoubleRunner< T >::iOffset [protected]**

Referenced by OsclDoubleRunner< T >::operator T \*(), OsclDoubleRunner< T >::operator++(), OsclDoubleRunner< T >::OsclDoubleRunner(), and OsclDoubleRunner< T >::Set().

The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.123 OsclError Class Reference

```
#include <oscl_error.h>
```

### Static Public Member Functions

- static OSCL\_IMPORT\_REF void **PushL** (*\_OsclHeapBase* \*aPtr)
- static OSCL\_IMPORT\_REF void **PushL** (*OsclAny* \*aPtr)
- static OSCL\_IMPORT\_REF void **PushL** (*OsclTrapItem* anItem)
- static OSCL\_IMPORT\_REF void **Pop** ()
- static OSCL\_IMPORT\_REF void **Pop** (int32 aCount)
- static OSCL\_IMPORT\_REF void **PopDealloc** ()
- static OSCL\_IMPORT\_REF void **PopDealloc** (int32 aCount)
- static OSCL\_IMPORT\_REF void **Leave** (int32 aReason)
- static OSCL\_IMPORT\_REF void **LeaveIfNull** (*OsclAny* \*a)
- static OSCL\_IMPORT\_REF void **LeaveIfError** (int32 aReason)

### 7.123.1 Detailed Description

User Error class

### 7.123.2 Member Function Documentation

#### 7.123.2.1 static 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.

Referenced by OsclSocketMethod::ConstructL(), OsclDNSRequestAO::ConstructL(), OsclTLSRegistryEx::getInstance(), OsclSingletonRegistryEx::getInstance(), OsclSingletonRegistryEx::lockAndGetInstance(), OsclBuf::NewL(), OsclTLSRegistryEx::registerInstance(), OsclSingletonRegistryEx::registerInstance(), and OsclSingletonRegistryEx::registerInstanceAndUnlock().

#### 7.123.2.2 static OSCL\_IMPORT\_REF void OsclError::LeaveIfError (int32 *aReason*) [static]

Evaluate the input parameter, and if it is an error code (non-zero), then do a Leave with the provided reason code.

#### 7.123.2.3 static 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.

Referenced by OsclMemBasicAllocator::allocate(), and OsclMemAllocator::allocate().

#### 7.123.2.4 static OSCL\_IMPORT\_REF void OsclError::Pop (int32 *aCount*) [static]

Pop the cleanup stack N times

**7.123.2.5 static OSCL\_IMPORT\_REF void OsclError::Pop () [static]**

Pop the cleanup stack

**7.123.2.6 static OSCL\_IMPORT\_REF void OsclError::PopDealloc (int32 aCount) [static]**

PopDealloc N times

**7.123.2.7 static OSCL\_IMPORT\_REF void OsclError::PopDealloc () [static]**

Destroy the item on the top of the cleanup stack and pop it

**7.123.2.8 static OSCL\_IMPORT\_REF void OsclError::PushL (OsclTrapItem anItem) [static]**

Push an [OsclTrapItem](#) onto the cleanup stack

**7.123.2.9 static OSCL\_IMPORT\_REF void OsclError::PushL (OsclAny \* aPtr) [static]**

Push an OsclAny item onto the cleanup stack.

**7.123.2.10 static OSCL\_IMPORT\_REF void OsclError::PushL (\_OsclHeapBase \* aPtr) [static]**

Cleanup stack operations. Push an [\\_OsclHeapBase](#) item onto the cleanup stack.

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 7.124 OsclErrorAllocator Class Reference

This class provides static methods to invoke the user defined memory allocation routines.

```
#include <oscl_error_allocator.h>
```

### Public Member Functions

- **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 Member Functions

- static **OsclAny \* allocate (uint32 aSize)**  
*static method to allocate a block of memory on heap*
- static **OsclAny deallocate (OsclAny \*aPointer)**  
*static method to deallocate a block of memory on heap*

#### 7.124.1 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

#### 7.124.2 Constructor & Destructor Documentation

##### 7.124.2.1 OsclErrorAllocator::OsclErrorAllocator (Oscl\_DefAlloc \* allocator) [inline]

constructor method

###### Parameters

*allocator* - a pointer to the concrete object that provides the allocator/deallocator

#### 7.124.3 Member Function Documentation

##### 7.124.3.1 static OsclAny\* OsclErrorAllocator::allocate (uint32 aSize) [inline, static]

static method to allocate a block of memory on heap

**Parameters**

*aSize* - number of bytes to allocate

References NULL, and OSCL\_ASSERT.

**7.124.3.2 static 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

References Oscl\_DefAlloc::deallocate(), NULL, and OSCL\_ASSERT.

**7.124.3.3 void OsclErrorAllocator::operator delete (OsclAny \* *aPtr*, OsclAny \* *aPtr2*) [inline]**

delete operator that doesn't do anything, user has to deallocate manually

References OSCL\_UNUSED\_ARG.

**7.124.3.4 void\* OsclErrorAllocator::operator new (uint32 *size*, OsclAny \* *aPtr*) [inline]**

placement new operator that allocates memory using the user defined methods

References OSCL\_UNUSED\_ARG.

The documentation for this class was generated from the following file:

- [oscl\\_error\\_allocator.h](#)

## 7.125 OsclErrorTrap Class Reference

```
#include <oscl_error.h>
```

### Static Public Member Functions

- static OSCL\_IMPORT\_REF int32 [Init \(Oscl\\_DefAlloc \\*aAlloc=NULL\)](#)
- static OSCL\_IMPORT\_REF int32 [Cleanup \(\)](#)
- static OSCL\_IMPORT\_REF [OsclErrorTrapImp \\* GetErrorTrapImp \(\)](#)

#### 7.125.1 Member Function Documentation

##### 7.125.1.1 static OSCL\_IMPORT\_REF int32 OsclErrorTrap::Cleanup () [static]

Cleanup and destroy error trap for the calling thread.

###### Returns

0 for success, or an error

##### 7.125.1.2 static OSCL\_IMPORT\_REF OsclErrorTrapImp\* OsclErrorTrap::GetErrorTrapImp () [static]

Get the ErrorTrapImp for the current thread. Leaves on error.

##### 7.125.1.3 static OSCL\_IMPORT\_REF int32 OsclErrorTrap::Init (Oscl\_DefAlloc \* *aAlloc* = NULL) [static]

Allocate and initialize error trap for the calling thread.

###### Parameters

*aAlloc*,: optional, allocator to use for the internal implementation.

###### Returns

0 for success, or an error

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 7.126 OsclErrorTrapImp Class Reference

```
#include <oscl_error_trapcleanup.h>
```

### Public Member Functions

- OSCL\_IMPORT\_REF void [UnTrap \(\)](#)

### Static Public Member Functions

- static OSCL\_IMPORT\_REF [OsclErrorTrapImp \\* Trap \(\)](#)
- static OSCL\_IMPORT\_REF [OsclErrorTrapImp \\* TrapNoTls \(OsclErrorTrapImp \\*\)](#)

### Data Fields

- [OsclJump \\* iJumpData](#)
- int32 [iLeave](#)
- [OsclTrapStack \\* iTrapStack](#)

### Friends

- class [OsclErrorTrap](#)
- class [OsclError](#)
- class [OsclExecScheduler](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclJump](#)
- class [OsclJumpMark](#)
- class [OsclTrapStack](#)
- class [CPVInterfaceProxy](#)
- class [OsclScheduler](#)

### 7.126.1 Detailed Description

A per-thread cleanup stack with nested trap support.

### 7.126.2 Member Function Documentation

#### 7.126.2.1 static OSCL\_IMPORT\_REF OsclErrorTrapImp\* OsclErrorTrapImp::Trap () [static]

PV trap cleanup. Public for use in macros only.

#### 7.126.2.2 static OSCL\_IMPORT\_REF OsclErrorTrapImp\* OsclErrorTrapImp::TrapNoTls (OsclErrorTrapImp \*) [static]

#### 7.126.2.3 OSCL\_IMPORT\_REF void OsclErrorTrapImp::UnTrap ()

these are used in public macros, but aren't intended as public methods or members.

### 7.126.3 Friends And Related Function Documentation

- 7.126.3.1 **friend class CPVInterfaceProxy [friend]**
- 7.126.3.2 **friend class OsclError [friend]**
- 7.126.3.3 **friend class OsclErrorTrap [friend]**
- 7.126.3.4 **friend class OsclExecScheduler [friend]**
- 7.126.3.5 **friend class OsclExecSchedulerCommonBase [friend]**
- 7.126.3.6 **friend class OsclJump [friend]**
- 7.126.3.7 **friend class OsclJumpMark [friend]**
- 7.126.3.8 **friend class OsclScheduler [friend]**
- 7.126.3.9 **friend class OsclTrapStack [friend]**

### 7.126.4 Field Documentation

- 7.126.4.1 **OsclJump\* OsclErrorTrapImp::iJumpData**
- 7.126.4.2 **int32 OsclErrorTrapImp::iLeave**
- 7.126.4.3 **OsclTrapStack\* OsclErrorTrapImp::iTrapStack**

The documentation for this class was generated from the following file:

- [oscl\\_error\\_trapcleanup.h](#)

## 7.127 OsclException< LeaveCode > Class Template Reference

[oscl\\_exception.h](#) contains all the exception handling macros and classes

```
#include <oscl_exception.h>
```

### Public Member Functions

- [OsclException \(\)](#)

### Static Public Member Functions

- static int [getLeaveCode \(\)](#)

#### 7.127.1 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

#### 7.127.2 Constructor & Destructor Documentation

7.127.2.1 **template<int LeaveCode> OsclException< LeaveCode >::OsclException () [inline]**

#### 7.127.3 Member Function Documentation

7.127.3.1 **template<int LeaveCode> static int OsclException< LeaveCode >::getLeaveCode () [inline, static]**

The documentation for this class was generated from the following file:

- [oscl\\_exception.h](#)

## 7.128 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 or indirectly) by new. When the [OsclExclusiveArrayPtr](#) expires, its destructor uses delete to free the memory.

```
#include <oscl_exclusive_ptr.h>
```

### Public Member Functions

- [OsclExclusiveArrayPtr \(T \\*inPtr=0\)](#)  
*Default constructor Initializes the pointer and takes ownership.*
- [OsclExclusiveArrayPtr \(OsclExclusiveArrayPtr< T > &\\_Y\)](#)  
*Copy constructor.*
- [OsclExclusiveArrayPtr< T > & operator= \(OsclExclusiveArrayPtr< T > &\\_Y\)](#)  
*Assignment operator from an another [OsclExclusiveArrayPtr](#).*
- virtual [~OsclExclusiveArrayPtr \(\)](#)  
*Destructor.*
- [T & operator\\* \(\) const](#)  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- [T \\* operator-> \(\) const](#)  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- [T \\* get \(\) const](#)  
*get() method returns the pointer, currently owned by the class.*
- [T \\* release \(\)](#)  
*release() method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- [bool set \(T \\*ptr\)](#)  
*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- [T \\* \\_Ptr](#)

#### 7.128.1 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 or indirectly) by new. When the [OsclExclusiveArrayPtr](#) expires,

its destructor uses delete to free the memory. The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an `OsclExclusivePtr` object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The `OsclExclusivePtr` is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### 7.128.2 Constructor & Destructor Documentation

**7.128.2.1 template<class T> OsclExclusiveArrayPtr< T >::OsclExclusiveArrayPtr (T \* *inPtr* = 0) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**7.128.2.2 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.

**7.128.2.3 template<class T> virtual OsclExclusiveArrayPtr< T >::~OsclExclusiveArrayPtr () [inline, virtual]**

Destructor.

The pointer is deleted in case this class still has ownership

References `OsclExclusiveArrayPtr< T >::_Ptr`.

### 7.128.3 Member Function Documentation

**7.128.3.1 template<class T> T\* OsclExclusiveArrayPtr< T >::get () const [inline]**

`get()` method returns the pointer, currently owned by the class.

References `OsclExclusiveArrayPtr< T >::_Ptr`.

Referenced by `OsclExclusiveArrayPtr< T >::operator=()`.

**7.128.3.2 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.

**7.128.3.3 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.

#### **7.128.3.4 template<class T> OsclExclusiveArrayPtr<T>& OsclExclusiveArrayPtr< T >::operator= (OsclExclusiveArrayPtr< T > & \_Y) [inline]**

Assignment operator from 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.

References [OsclExclusiveArrayPtr< T >::\\_Ptr](#), [OsclExclusiveArrayPtr< T >::get\(\)](#), and [OsclExclusiveArrayPtr< T >::release\(\)](#).

#### **7.128.3.5 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.

References [OsclExclusiveArrayPtr< T >::\\_Ptr](#), and [NULL](#).

Referenced by [OsclExclusiveArrayPtr< T >::operator=\(\)](#).

#### **7.128.3.6 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.

References [OsclExclusiveArrayPtr< T >::\\_Ptr](#), and [NULL](#).

### **7.128.4 Field Documentation**

#### **7.128.4.1 template<class T> T\* OsclExclusiveArrayPtr< T >::\_Ptr [protected]**

Referenced by [OsclExclusiveArrayPtr< T >::get\(\)](#), [OsclExclusiveArrayPtr< T >::operator=\(\)](#), [OsclExclusiveArrayPtr< T >::release\(\)](#), [OsclExclusiveArrayPtr< T >::set\(\)](#), and [OsclExclusiveArrayPtr< T >::~OsclExclusiveArrayPtr\(\)](#).

The documentation for this class was generated from the following file:

- [oscl\\_exclusive\\_ptr.h](#)

## 7.129 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 Member Functions

- [OsclExclusivePtr \(T \\*inPtr=0\)](#)  
*Default constructor Initializes the pointer and takes ownership.*
- [OsclExclusivePtr \(OsclExclusivePtr< T > &\\_Y\)](#)  
*Copy constructor.*
- [OsclExclusivePtr< T > & operator= \(OsclExclusivePtr< T > &\\_Y\)](#)  
*Assignment operator from an another [OsclExclusivePtr](#).*
- virtual [~OsclExclusivePtr \(\)](#)  
*Destructor.*
- [T & operator\\* \(\) const](#)  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- [T \\* operator-> \(\) const](#)  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- [T \\* get \(\) const](#)  
*[get\(\)](#) method returns the pointer, currently owned by the class.*
- [T \\* release \(\)](#)  
*[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- [bool set \(T \\*ptr\)](#)  
*[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- [T \\* \\_Ptr](#)

#### 7.129.1 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

## 7.129.2 Constructor & Destructor Documentation

**7.129.2.1 template<class T> OsclExclusivePtr< T >::OsclExclusivePtr (T \* *inPtr* = 0) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**7.129.2.2 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.

**7.129.2.3 template<class T> virtual OsclExclusivePtr< T >::~OsclExclusivePtr () [inline, virtual]**

Destructor.

The pointer is deleted in case this class still has ownership

References [OsclExclusivePtr< T >::\\_Ptr](#).

## 7.129.3 Member Function Documentation

**7.129.3.1 template<class T> T\* OsclExclusivePtr< T >::get () const [inline]**

[get\(\)](#) method returns the pointer, currently owned by the class.

References [OsclExclusivePtr< T >::\\_Ptr](#).

Referenced by [OsclExclusivePtr< T >::operator=\(\)](#).

**7.129.3.2 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.

**7.129.3.3 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.

#### **7.129.3.4 template<class T> OsclExclusivePtr<T>& OsclExclusivePtr< T >::operator= (OsclExclusivePtr< T > & \_Y) [inline]**

Assignment operator from 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.

References [OsclExclusivePtr< T >::\\_Ptr](#), [OsclExclusivePtr< T >::get\(\)](#), and [OsclExclusivePtr< T >::release\(\)](#).

#### **7.129.3.5 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.

References [OsclExclusivePtr< T >::\\_Ptr](#), and [NULL](#).

Referenced by [OsclExclusivePtr< T >::operator=\(\)](#).

#### **7.129.3.6 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.

References [OsclExclusivePtr< T >::\\_Ptr](#), and [NULL](#).

### **7.129.4 Field Documentation**

#### **7.129.4.1 template<class T> T\* OsclExclusivePtr< T >::\_Ptr [protected]**

Referenced by [OsclExclusivePtr< T >::get\(\)](#), [OsclExclusivePtr< T >::operator=\(\)](#), [OsclExclusivePtr< T >::release\(\)](#), [OsclExclusivePtr< T >::set\(\)](#), and [OsclExclusivePtr< T >::~OsclExclusivePtr\(\)](#).

The documentation for this class was generated from the following file:

- [oscl\\_exclusive\\_ptr.h](#)

## 7.130 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 Member Functions

- [OsclExclusivePtrA](#) (T \*inPtr=0)  
*Default constructor Initializes the pointer and takes ownership.*
- [OsclExclusivePtrA](#) ([OsclExclusivePtrA](#)< T, Alloc > &\_Y)  
*Copy constructor.*
- [OsclExclusivePtrA](#)< T, Alloc > & [operator=](#) ([OsclExclusivePtrA](#)< T, Alloc > &\_Y)  
*Assignment operator from an another [OsclExclusiveArrayPtr](#).*
- virtual [~OsclExclusivePtrA](#) ()  
*Destructor.*
- T & [operator\\*](#) () const  
*The indirection operator (\*) accesses a value indirectly, through a pointer.*
- T \* [operator->](#) () const  
*The indirection operator (->) accesses a value indirectly, through a pointer.*
- T \* [get](#) () const  
*[get\(\)](#) method returns the pointer, currently owned by the class.*
- T \* [release](#) ()  
*[release\(\)](#) method releases ownership of the pointer, currently owned by the class. It returns the pointer as well.*
- bool [set](#) (T \*ptr)  
*[set\(\)](#) method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- T \* [\\_Ptr](#)

#### 7.130.1 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

### 7.130.2 Constructor & Destructor Documentation

**7.130.2.1 template<class T, class Alloc> OsclExclusivePtrA< T, Alloc >::OsclExclusivePtrA (T \*  
inPtr = 0) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

**7.130.2.2 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.

**7.130.2.3 template<class T, class Alloc> virtual OsclExclusivePtrA< T, Alloc  
>::~OsclExclusivePtrA () [inline, virtual]**

Destructor.

The pointer is deleted in case this class still has ownership

References [OsclExclusivePtrA< T, Alloc >::\\_Ptr](#).

### 7.130.3 Member Function Documentation

**7.130.3.1 template<class T, class Alloc> T\* OsclExclusivePtrA< T, Alloc >::get () const  
[inline]**

[get\(\)](#) method returns the pointer, currently owned by the class.

References [OsclExclusivePtrA< T, Alloc >::\\_Ptr](#).

Referenced by [OsclExclusivePtrA< T, Alloc >::operator=\(\)](#).

**7.130.3.2 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.

### 7.130.3.3 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.

### 7.130.3.4 template<class T, class Alloc> OsclExclusivePtrA<T, Alloc>& OsclExclusivePtrA< T, Alloc >::operator= (OsclExclusivePtrA< T, Alloc > & \_Y) [inline]

Assignment operator from 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.

References [OsclExclusivePtrA< T, Alloc >::\\_Ptr](#), [OsclExclusivePtrA< T, Alloc >::get\(\)](#), and [OsclExclusivePtrA< T, Alloc >::release\(\)](#).

### 7.130.3.5 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.

References [OsclExclusivePtrA< T, Alloc >::\\_Ptr](#), and [NULL](#).

Referenced by [OsclExclusivePtrA< T, Alloc >::operator=\(\)](#).

### 7.130.3.6 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.

References [OsclExclusivePtrA< T, Alloc >::\\_Ptr](#), and [NULL](#).

## 7.130.4 Field Documentation

### 7.130.4.1 template<class T, class Alloc> T\* OsclExclusivePtrA< T, Alloc >::\_Ptr [protected]

Referenced by [OsclExclusivePtrA< T, Alloc >::get\(\)](#), [OsclExclusivePtrA< T, Alloc >::operator=\(\)](#), [OsclExclusivePtrA< T, Alloc >::release\(\)](#), [OsclExclusivePtrA< T, Alloc >::set\(\)](#), and

`OsclExclusivePtrA< T, Alloc >::~OsclExclusivePtrA()`.

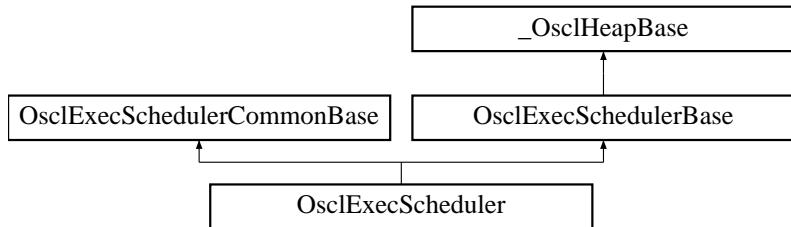
The documentation for this class was generated from the following file:

- [oscl\\_exclusive\\_ptr.h](#)

## 7.131 OsclExecScheduler Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for OsclExecScheduler:



### Public Member Functions

- OSCL\_IMPORT\_REF void [RunSchedulerNonBlocking](#) (int32 aTargetCount, int32 &aReady, uint32 &aDelayMsec)
- OSCL\_IMPORT\_REF void [RegisterForCallback](#) (OsclSchedulerObserver \*aCallback, [OsclAny](#) \*aCallbackContext)

### Static Public Member Functions

- static OSCL\_IMPORT\_REF [OsclExecScheduler](#) \* [Current](#) ()

### Friends

- class [OsclScheduler](#)

#### 7.131.1 Member Function Documentation

**7.131.1.1 static OSCL\_IMPORT\_REF OsclExecScheduler\* OsclExecScheduler::Current () [static]**

Get currently installed scheduler for calling thread, or NULL if no scheduler is installed.

**7.131.1.2 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.

**7.131.1.3 OSCL\_IMPORT\_REF void OsclExecScheduler::RunSchedulerNonBlocking (int32 aTargetCount, int32 & aReady, uint32 & aDelayMsec)**

Non-Blocking scheduler APIs. Run PV scheduler in non-blocking mode. This call returns when the desired number of Run calls have been made, or when there are no more active objects that are ready to run.

### Parameters

*aTargetCount*,: (input param) the maximum number of Run calls to make.

*aReady*,: (output param) tells the number of active objects that are currently ready to run.

*aDelayMsec*,: (output param) If no active objects are ready to run, but one or more active objects are waiting on timers, this parameter will tell the time interval from the current time until the first of the pending timer objects will be ready to run, in milliseconds.

Note: On Symbian, non-blocking mode is not supported and this call will leave.

## 7.131.2 Friends And Related Function Documentation

### 7.131.2.1 friend class OsclScheduler [friend]

Reimplemented from [OsclExecSchedulerCommonBase](#).

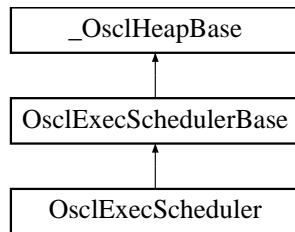
The documentation for this class was generated from the following file:

- [oscl\\_scheduler.h](#)

## 7.132 OsclExecSchedulerBase Class Reference

```
#include <oscl_scheduler_types.h>
```

Inheritance diagram for OsclExecSchedulerBase:



### Friends

- class [OsclExecScheduler](#)
- class [OsclCoeActiveScheduler](#)
- class [PVActiveBase](#)

### 7.132.1 Detailed Description

This file defines the OsclExecBase, OsclTimerBase, and [OsclExecSchedulerBase](#) classes. These are the base classes for PV AOs and PV Scheduler. We want the PV exec objects to be usable with either a PV scheduler or a non-PV native Symbian scheduler. We also want the PV scheduler to be usable with non-PV exec objects. Therefore, the PV scheduler and AO classes derived from Symbian classes on Symbian platforms. On non-Symbian platforms, the PV classes derive from classes with a similar API to the Symbian classes. OsclActiveSchedulerBase is the base for [OsclExecScheduler](#). The non-Symbian OsclActiveSchedulerBase class is functionally similar to a subset of Symbian CActiveScheduler class.

### 7.132.2 Friends And Related Function Documentation

**7.132.2.1 friend class OsclCoeActiveScheduler [friend]**

**7.132.2.2 friend class OsclExecScheduler [friend]**

**7.132.2.3 friend class PVActiveBase [friend]**

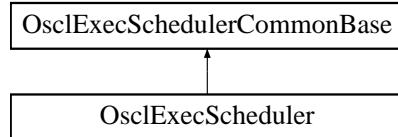
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_types.h](#)

## 7.133 OsclExecSchedulerCommonBase Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for OsclExecSchedulerCommonBase:



### Public Member Functions

- 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 Member Functions

- static OSCL\_IMPORT\_REF [OsclNameString< PVSCHEDNAMELEN > \\*](#) [GetName](#) ()
- static OSCL\_IMPORT\_REF uint32 [GetId](#) ()

### Protected Member Functions

- 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](#) \*)
- [PVActiveBase](#) \* [FindPVB](#) ([PVActiveBase](#) \*active, [OsclDoubleList< PVActiveBase >](#) &)
- void [CleanupExecQ](#) ()
- void [InitExecQ](#) (int)
- void [ResetLogPerf](#) ()
- void [IncLogPerf](#) (uint32)

## Static Protected Member Functions

- static [OsclExecSchedulerCommonBase \\* GetScheduler \(\)](#)
- static [OsclExecSchedulerCommonBase \\* SetScheduler \(OsclExecSchedulerCommonBase \\*\)](#)

## 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](#)
- [PVLogger \\* iLogger](#)
- [PVLogger \\* iDebugLogger](#)
- char \* [iLogPerfIndentStr](#)
- int32 [iLogPerfIndentStrLen](#)
- uint32 [iLogPerfTotal](#)
- [Oscl\\_DefAlloc \\* iAlloc](#)
- [OsclMemAllocator iDefAlloc](#)

## Static Protected Attributes

- static const uint32 [iTTimeCompareThreshold](#)

## Friends

- class [OsclScheduler](#)
- class [PVThreadContext](#)
- class [OsclCoeActiveScheduler](#)
- class [OsclTimerCompare](#)
- class [OsclReadyQ](#)
- class [OsclError](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)
- class [PVActiveBase](#)
- class [PVSchedulerStopper](#)
- class [OsclExecScheduler](#)

### 7.133.1 Constructor & Destructor Documentation

- 7.133.1.1 `virtual OsclExecSchedulerCommonBase::~OsclExecSchedulerCommonBase ()`  
`[protected, virtual]`
- 7.133.1.2 `OsclExecSchedulerCommonBase::OsclExecSchedulerCommonBase (Oscl_DefAlloc *)`  
`[protected]`

### 7.133.2 Member Function Documentation

- 7.133.2.1 `void OsclExecSchedulerCommonBase::AddToExecTimerQ (PVActiveBase * active, uint32) [protected]`
- 7.133.2.2 `void OsclExecSchedulerCommonBase::BeginScheduling (bool blocking, bool native) [protected]`
- 7.133.2.3 `void OsclExecSchedulerCommonBase::BlockingLoopL () [protected]`
- 7.133.2.4 `void OsclExecSchedulerCommonBase::CallRunExec (PVActiveBase *) [protected]`
- 7.133.2.5 `void OsclExecSchedulerCommonBase::CleanupExecQ () [protected]`
- 7.133.2.6 `virtual void OsclExecSchedulerCommonBase::ConstructL (const char * name, int) [protected, virtual]`
- 7.133.2.7 `void OsclExecSchedulerCommonBase::EndScheduling () [protected]`
- 7.133.2.8 `void OsclExecSchedulerCommonBase::Error (int32 anError) const [protected]`
- 7.133.2.9 `PVActiveBase* OsclExecSchedulerCommonBase::FindPVBase (PVActiveBase * active, OsclDoubleList< PVActiveBase > &) [protected]`
- 7.133.2.10 `static OSCL_IMPORT_REF uint32 OsclExecSchedulerCommonBase::GetId () [static]`

Get numeric ID of current thread.

- 7.133.2.11 `static OSCL_IMPORT_REF OsclNameString< PVSCHEDNAMELEN >* OsclExecSchedulerCommonBase::GetName () [static]`

Get name of scheduler for current thread.

- 7.133.2.12 **static OsclExecSchedulerCommonBase\* OsclExecSchedulerCommonBase::GetScheduler () [static, protected]**
- 7.133.2.13 **void OsclExecSchedulerCommonBase::IncLogPerf (uint32) [protected]**
- 7.133.2.14 **void OsclExecSchedulerCommonBase::InitExecQ (int) [protected]**
- 7.133.2.15 **void OsclExecSchedulerCommonBase::InstallScheduler () [protected]**
- 7.133.2.16 **bool OsclExecSchedulerCommonBase::IsInstalled () [inline, protected]**
- 7.133.2.17 **bool OsclExecSchedulerCommonBase::IsStarted () [protected]**
- 7.133.2.18 **void OsclExecSchedulerCommonBase::PendComplete (PVActiveBase \*, int32 *aReason*, TPVThreadContext *aContext*) [protected]**
- 7.133.2.19 **void OsclExecSchedulerCommonBase::RequestCanceled (PVActiveBase \*) [protected]**
- 7.133.2.20 **void OsclExecSchedulerCommonBase::ResetLogPerf () [protected]**
- 7.133.2.21 **OSCL\_IMPORT\_REF void OsclExecSchedulerCommonBase::ResumeScheduler ()**

Resume scheduling immediately. This API only applies to a blocking loop scheduler.

- 7.133.2.22 **static OsclExecSchedulerCommonBase\* OsclExecSchedulerCommonBase::SetScheduler (OsclExecSchedulerCommonBase \*) [static, protected]**
  - 7.133.2.23 **OSCL\_IMPORT\_REF void OsclExecSchedulerCommonBase::StartNativeScheduler ()**
- Start the OS native scheduling loop. This is an alternative to the PV scheduling loop. To stop the native scheduler, use the StopScheduler API.
- 7.133.2.24 **OSCL\_IMPORT\_REF void OsclExecSchedulerCommonBase::StartScheduler (OsclSemaphore \* *sem* = NULL)**

Start scheduling. This call blocks until scheduler is stopped or an error occurs.

#### Parameters

*sem*: optional startup semaphore. If provided, the scheduler will signal this semaphore when the startup has progressed to the point that it's safe to call StopScheduler or SuspendScheduler from another thread.

- 7.133.2.25 **OSCL\_IMPORT\_REF void OsclExecSchedulerCommonBase::StopScheduler ()**

Stop scheduling. This API may be called from the scheduling thread or some other thread.

**7.133.2.26 OSCL\_IMPORT\_REF void OsclExecSchedulerCommonBase::SuspendScheduler ()**

Suspend scheduling when the current Run is complete. This API only applies to a blocking loop scheduler.

**7.133.2.27 void OsclExecSchedulerCommonBase::UninstallScheduler () [protected]****7.133.2.28 PVActiveBase\* OsclExecSchedulerCommonBase::UpdateTimers (uint32 & aDelay) [protected]****7.133.2.29 PVActiveBase\* OsclExecSchedulerCommonBase::UpdateTimersMsec (uint32 & aDelay) [protected]****7.133.2.30 PVActiveBase\* OsclExecSchedulerCommonBase::WaitForReadyAO () [protected]****7.133.3 Friends And Related Function Documentation****7.133.3.1 friend class OsclActiveObject [friend]****7.133.3.2 friend class OsclCoeActiveScheduler [friend]****7.133.3.3 friend class OsclError [friend]****7.133.3.4 friend class OsclExecScheduler [friend]****7.133.3.5 friend class OsclReadyQ [friend]****7.133.3.6 friend class OsclScheduler [friend]**

Reimplemented in [OsclExecScheduler](#).



- 7.133.3.7 friend class OsclTimerCompare [friend]
- 7.133.3.8 friend class OsclTimerObject [friend]
- 7.133.3.9 friend class PVActiveBase [friend]
- 7.133.3.10 friend class PVSchedulerStopper [friend]
- 7.133.3.11 friend class PVThreadContext [friend]

#### 7.133.4 Field Documentation

- 7.133.4.1 Oscl\_DefAlloc\* OsclExecSchedulerCommonBase::iAlloc [protected]
- 7.133.4.2 bool OsclExecSchedulerCommonBase::iBlockingMode [protected]
- 7.133.4.3 PVLogger\* OsclExecSchedulerCommonBase::iDebugLogger [protected]
- 7.133.4.4 OsclMemAllocator OsclExecSchedulerCommonBase::iDefAlloc [protected]
- 7.133.4.5 bool OsclExecSchedulerCommonBase::iDoStop [protected]
- 7.133.4.6 bool OsclExecSchedulerCommonBase::iDoSuspend [protected]
- 7.133.4.7 OsclErrorTrapImp\* OsclExecSchedulerCommonBase::iErrorTrapImp [protected]
- 7.133.4.8 OsclTimerQ OsclExecSchedulerCommonBase::iExecTimerQ [protected]
- 7.133.4.9 PVLogger\* OsclExecSchedulerCommonBase::iLogger [protected]
- 7.133.4.10 char\* OsclExecSchedulerCommonBase::iLogPerfIndentStr [protected]
- 7.133.4.11 int32 OsclExecSchedulerCommonBase::iLogPerfIndentStrLen [protected]
- 7.133.4.12 uint32 OsclExecSchedulerCommonBase::iLogPerfTotal [protected]
- 7.133.4.13 OsclNameString<PVSCEDNAMELEN> OsclExecSchedulerCommonBase::iName [protected]
- 7.133.4.14 bool OsclExecSchedulerCommonBase::iNativeMode [protected]
- 7.133.4.15 uint32 OsclExecSchedulerCommonBase::iNumAOAdded [protected]
- 7.133.4.16 OsclReadyQ OsclExecSchedulerCommonBase::iReadyQ [protected]
- 7.133.4.17 OsclSemaphore OsclExecSchedulerCommonBase::iResumeSem [protected]
- 7.133.4.18 PVSchedulerStopper\* OsclExecSchedulerCommonBase::iStopper [protected]
- 7.133.4.19 OsclNoYieldMutex OsclExecSchedulerCommonBase::iStopperCrit [protected]
- 7.133.4.20 bool OsclExecSchedulerCommonBase::iSuspended [protected]
- 7.133.4.21 PVThreadContext OsclExecSchedulerCommonBase::iThreadContext [protected]
- 7.133.4.22 const uint32 OsclExecSchedulerCommonBase::iTimeCompareThreshold [static, protected]

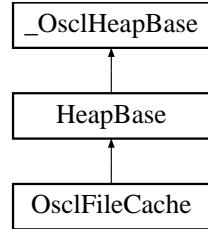
The documentation for this class was generated from the following file:

- [oscl\\_scheduler.h](#)

## 7.134 OsclFileCache Class Reference

```
#include <oscl_file_cache.h>
```

Inheritance diagram for OsclFileCache:



### Public Member Functions

- [OsclFileCache \(Oscl\\_File &aContainer\)](#)
- [~OsclFileCache \(\)](#)
- int32 [Open \(uint32 mode, uint32 cache\\_size\)](#)
- void [Close \(\)](#)
- uint32 [Read \(void \\*outputBuffer, uint32 size, uint32 numelements\)](#)
- uint32 [Write \(const void \\*inputBuffer, uint32 size, uint32 numelements\)](#)
- [TOsclFileOffset FileSize \(\)](#)
- int32 [Seek \(TOsclFileOffset offset, Oscl\\_File::seek\\_type origin\)](#)
- [TOsclFileOffset Tell \(\)](#)
- int32 [Flush \(\)](#)
- int32 [EndOfFile \(\)](#)
- OSCL\_IMPORT\_REF [OsclFileCacheBuffer \\* AddFixedCache \(const Oscl\\_File::OsclFixedCacheParam &\)](#)

### Data Fields

- [OsclFileCacheBuffer \\_movableCache](#)
- [Oscl\\_Vector< OsclFileCacheBuffer, OsclMemAllocator > \\_fixedCaches](#)

### Friends

- class [OsclFileCacheBuffer](#)

### 7.134.1 Constructor & Destructor Documentation

**7.134.1.1 OsclFileCache::OsclFileCache (Oscl\_File & *aContainer*)**

**7.134.1.2 OsclFileCache::~OsclFileCache ()**

### 7.134.2 Member Function Documentation

**7.134.2.1 OSCL\_IMPORT\_REF OsclFileCacheBuffer\* OsclFileCache::AddFixedCache (const Oscl\_File::OsclFixedCacheParam &)**

**7.134.2.2 void OsclFileCache::Close ()**

**7.134.2.3 int32 OsclFileCache::EndOfFile () [inline]**

References FileSize(), and Tell().

**7.134.2.4 TOsclFileOffset OsclFileCache::FileSize () [inline]**

Referenced by EndOfFile().

**7.134.2.5 int32 OsclFileCache::Flush ()**

**7.134.2.6 int32 OsclFileCache::Open (uint32 *mode*, uint32 *cache\_size*)**

**7.134.2.7 uint32 OsclFileCache::Read (void \* *outputBuffer*, uint32 *size*, uint32 *numelements*)**

**7.134.2.8 int32 OsclFileCache::Seek (TOsclFileOffset *offset*, Oscl\_File::seek\_type *origin*)**

**7.134.2.9 TOsclFileOffset OsclFileCache::Tell () [inline]**

References OsclFileCacheBuffer::currentPos, and OsclFileCacheBuffer::filePosition.

Referenced by EndOfFile().

**7.134.2.10 uint32 OsclFileCache::Write (const void \* *inputBuffer*, uint32 *size*, uint32 *numelements*)**

### 7.134.3 Friends And Related Function Documentation

**7.134.3.1 friend class OsclFileCacheBuffer [friend]**

### 7.134.4 Field Documentation

**7.134.4.1 Oscl\_Vector<OsclFileCacheBuffer, OsclMemAllocator> OsclFileCache::\_fixedCaches**

**7.134.4.2 OsclFileCacheBuffer OsclFileCache::\_movableCache**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_cache.h](#)

## 7.135 OsclFileCacheBuffer Class Reference

```
#include <oscl_file_cache.h>
```

### Public Member Functions

- [OsclFileCacheBuffer \(\)](#)
- [int32 SetPosition \(TOsclFileOffset pos\)](#)
- [int32 PrepRead \(\)](#)
- [int32 PrepWrite \(\)](#)
- [int32 WriteUpdatesToFile \(\)](#)
- [int32 FillFromFile \(uint32, uint32\)](#)
- [bool IsUpdated \(\)](#)
- [bool Contains \(TOsclFileOffset pos\)](#)
- [bool Preceeds \(TOsclFileOffset pos\)](#)

### Data Fields

- [OsclFileCache \\* iContainer](#)
- [bool isFixed](#)
- [uint32 capacity](#)
- [uint32 usableSize](#)
- [uint8 \\* pBuffer](#)
- [TOsclFileOffset filePosition](#)
- [uint32 currentPos](#)
- [uint32 endPos](#)
- [uint32 updateStart](#)
- [uint32 updateEnd](#)

#### 7.135.1 Constructor & Destructor Documentation

7.135.1.1 [OsclFileCacheBuffer::OsclFileCacheBuffer \(\) \[inline\]](#)

#### 7.135.2 Member Function Documentation

7.135.2.1 [bool OsclFileCacheBuffer::Contains \(TOsclFileOffset pos\) \[inline\]](#)

References filePosition, and usableSize.

7.135.2.2 [int32 OsclFileCacheBuffer::FillFromFile \(uint32, uint32\)](#)

7.135.2.3 [bool OsclFileCacheBuffer::IsUpdated \(\) \[inline\]](#)

References updateEnd, and updateStart.

7.135.2.4 [bool OsclFileCacheBuffer::Preceeds \(TOsclFileOffset pos\) \[inline\]](#)

References filePosition, and usableSize.

**7.135.2.5 int32 OsclFileCacheBuffer::PrepRead ()**

**7.135.2.6 int32 OsclFileCacheBuffer::PrepWrite ()**

**7.135.2.7 int32 OsclFileCacheBuffer::SetPosition (TOsclFileOffset *pos*)**

**7.135.2.8 int32 OsclFileCacheBuffer::WriteUpdatesToFile ()**

### **7.135.3 Field Documentation**

**7.135.3.1 uint32 OsclFileCacheBuffer::capacity**

**7.135.3.2 uint32 OsclFileCacheBuffer::currentPos**

Referenced by OsclFileCache::Tell().

**7.135.3.3 uint32 OsclFileCacheBuffer::endPos**

**7.135.3.4 TOsclFileOffset OsclFileCacheBuffer::filePosition**

Referenced by Contains(), Preceeds(), and OsclFileCache::Tell().

**7.135.3.5 OsclFileCache\* OsclFileCacheBuffer::iContainer**

**7.135.3.6 bool OsclFileCacheBuffer::isFixed**

**7.135.3.7 uint8\* OsclFileCacheBuffer::pBuffer**

**7.135.3.8 uint32 OsclFileCacheBuffer::updateEnd**

Referenced by IsUpdated().

**7.135.3.9 uint32 OsclFileCacheBuffer::updateStart**

Referenced by IsUpdated().

**7.135.3.10 uint32 OsclFileCacheBuffer::usableSize**

Referenced by Contains(), and Preceeds().

The documentation for this class was generated from the following file:

- [oscl\\_file\\_cache.h](#)

## 7.136 OsclFileHandle Class Reference

```
#include <oscl_file_handle.h>
```

### Public Member Functions

- [OsclFileHandle \(TOsclFileHandle aHandle\)](#)
- [OsclFileHandle \(const OsclFileHandle &aHandle\)](#)
- [TOsclFileHandle Handle \(\) const](#)

### Friends

- class [Oscl\\_File](#)

### 7.136.1 Detailed Description

[OsclFileHandle](#) is a container for a handle to a previously-opened file.

### 7.136.2 Constructor & Destructor Documentation

[7.136.2.1 OsclFileHandle::OsclFileHandle \(TOsclFileHandle \*aHandle\*\) \[inline\]](#)

[7.136.2.2 OsclFileHandle::OsclFileHandle \(const OsclFileHandle & \*aHandle\*\) \[inline\]](#)

### 7.136.3 Member Function Documentation

[7.136.3.1 TOsclFileHandle OsclFileHandle::Handle \(\) const \[inline\]](#)

### 7.136.4 Friends And Related Function Documentation

[7.136.4.1 friend class Oscl\\_File \[friend\]](#)

The documentation for this class was generated from the following file:

- [oscl\\_file\\_handle.h](#)

## 7.137 OsclFileManager Class Reference

```
#include <oscl_file_manager.h>
```

### Public Types

- enum `OSCL_FILE_ATTRIBUTE_TYPE` {
   
`OSCL_FILE_ATTRIBUTE_READONLY` = 0x00000001, `OSCL_FILE_ATTRIBUTE_HIDDEN` = 0x00000002, `OSCL_FILE_ATTRIBUTE_SYSTEM` = 0x00000004, `OSCL_FILE_ATTRIBUTE_DIRECTORY` = 0x00000010,
   
`OSCL_FILE_ATTRIBUTE_ARCHIVE` = 0x00000020, `OSCL_FILE_ATTRIBUTE_NORMAL` = 0x00000080 }

### Static Public Member Functions

- static `OSCL_IMPORT_REF` bool `OsclGetFileSize` (const `oscl_wchar` \*aFileName, `uint64` &aFileSize)
- static `OSCL_IMPORT_REF` bool `OsclGetFileSize` (const char \*aFileName, `uint64` &aFileSize)
- static `OSCL_IMPORT_REF` bool `OsclGetFileCreationTime` (const `oscl_wchar` \*aFileName, `uint64` &aFileCreationTime)
- static `OSCL_IMPORT_REF` bool `OsclGetFileCreationTime` (const char \*aFileName, `uint64` &aFileCreationTime)
- static `OSCL_IMPORT_REF` bool `OsclGetFileLastAccessTime` (const `oscl_wchar` \*aFileName, `uint64` &aFileLastAccessTime)
- static `OSCL_IMPORT_REF` bool `OsclGetFileLastAccessTime` (const char \*aFileName, `uint64` &aFileLastAccessTime)
- static `OSCL_IMPORT_REF` bool `OsclGetFileLastWriteTime` (const `oscl_wchar` \*aFileName, `uint64` &aFileLastWriteTime)
- static `OSCL_IMPORT_REF` bool `OsclGetFileLastWriteTime` (const char \*aFileName, `uint64` &aFileLastWriteTime)
- static `OSCL_IMPORT_REF` bool `OsclGetFileAttributes` (const `oscl_wchar` \*aFileName, `uint32` &aFileAttributes)
- static `OSCL_IMPORT_REF` bool `OsclGetFileAttributes` (const char \*aFileName, `uint32` &aFileAttributes)
- static `OSCL_IMPORT_REF` void `OsclExtractFilenameFromFullPath` (const char \*aPath, char \*&aFileName)
- static `OSCL_IMPORT_REF` void `OsclExtractFilenameFromFullPath` (const `oscl_wchar` \*aPath, `oscl_wchar` \*&aFileName)

### 7.137.1 Member Enumeration Documentation

#### 7.137.1.1 enum OsclFileManager::OSCL\_FILE\_ATTRIBUTE\_TYPE

Enumerator:

`OSCL_FILE_ATTRIBUTE_READONLY`  
`OSCL_FILE_ATTRIBUTE_HIDDEN`  
`OSCL_FILE_ATTRIBUTE_SYSTEM`  
`OSCL_FILE_ATTRIBUTE_DIRECTORY`

*OSCL\_FILE\_ATTRIBUTE\_ARCHIVE*  
*OSCL\_FILE\_ATTRIBUTE\_NORMAL*

The documentation for this class was generated from the following file:

- [oscl\\_file\\_manager.h](#)

## 7.138 OsclFileStats Class Reference

```
#include <oscl_file_stats.h>
```

### Public Member Functions

- [OsclFileStats \(Oscl\\_File \\*c\)](#)
- void [Start \(uint32 &aTicks\)](#)
- void [End \(TOsclFileOp aOp, uint32 aStart, uint32 aParam=0, TOsclFileOffset aParam2=0\)](#)
- void [Log \(TOsclFileOp, PVLogger \\*, uint32\)](#)
- void [LogAll \(PVLogger \\*, uint32\)](#)

#### 7.138.1 Constructor & Destructor Documentation

7.138.1.1 [OsclFileStats::OsclFileStats \(Oscl\\_File \\* c\)](#)

#### 7.138.2 Member Function Documentation

7.138.2.1 [void OsclFileStats::End \(TOsclFileOp aOp, uint32 aStart, uint32 aParam = 0, TOsclFileOffset aParam2 = 0\)](#)

7.138.2.2 [void OsclFileStats::Log \(TOsclFileOp, PVLogger \\*, uint32\)](#)

7.138.2.3 [void OsclFileStats::LogAll \(PVLogger \\*, uint32\)](#)

7.138.2.4 [void OsclFileStats::Start \(uint32 & aTicks\)](#)

The documentation for this class was generated from the following file:

- [oscl\\_file\\_stats.h](#)

## 7.139 OsclFileStatsItem Class Reference

```
#include <oscl_file_stats.h>
```

### Data Fields

- uint32 [iOpCount](#)
- uint64 [iParam](#)
- TOsclFileOffset [iParam2](#)
- uint32 [iStartTick](#)
- uint32 [iTTotalTicks](#)

### 7.139.1 Field Documentation

**7.139.1.1 uint32 OsclFileStatsItem::iOpCount**

**7.139.1.2 uint64 OsclFileStatsItem::iParam**

**7.139.1.3 TOsclFileOffset OsclFileStatsItem::iParam2**

**7.139.1.4 uint32 OsclFileStatsItem::iStartTick**

**7.139.1.5 uint32 OsclFileStatsItem::iTTotalTicks**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_stats.h](#)

## 7.140 Oscl\_File::OsclFixedCacheParam Class Reference

```
#include <oscl_file_io.h>
```

### Public Member Functions

- bool [Contains \(TOsclFileOffset pos\) const](#)

### Data Fields

- TOsclFileOffset [iFilePosition](#)
- uint32 [iSize](#)

#### 7.140.1 Detailed Description

Parameters for defining a fixed cache

#### 7.140.2 Member Function Documentation

##### 7.140.2.1 bool Oscl\_File::OsclFixedCacheParam::Contains (TOsclFileOffset *pos*) const [inline]

References iFilePosition, and iSize.

#### 7.140.3 Field Documentation

##### 7.140.3.1 TOsclFileOffset Oscl\_File::OsclFixedCacheParam::iFilePosition

Referenced by Contains().

##### 7.140.3.2 uint32 Oscl\_File::OsclFixedCacheParam::iSize

Referenced by Contains().

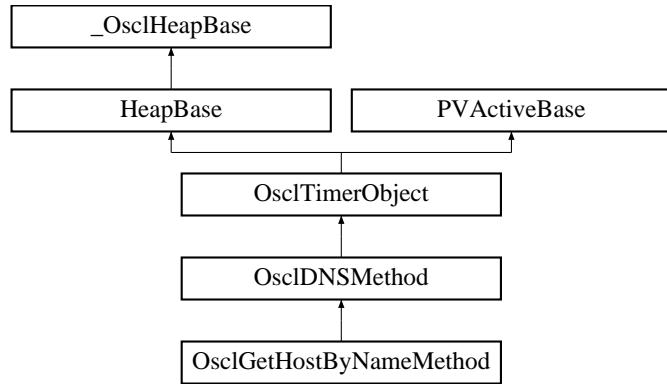
The documentation for this class was generated from the following file:

- [oscl\\_file\\_io.h](#)

## 7.141 OsclGetHostByNameMethod Class Reference

```
#include <oscl_dns_gethostbyname.h>
```

Inheritance diagram for OsclGetHostByNameMethod:



### Public Member Functions

- `~OsclGetHostByNameMethod ()`
- `TPVDNSEvent GetHostByName (char *name, OsclNetworkAddress *addr, int32 aTimeout, Oscl_-Vector< OsclNetworkAddress, OsclMemAllocator > *aAddressList)`

### Static Public Member Functions

- static `OsclGetHostByNameMethod * NewL (Oscl_DefAlloc &a, OsclDNSI *aDNS, OsclDNSObserver *aObserver, uint32 aId)`

#### 7.141.1 Constructor & Destructor Documentation

##### 7.141.1.1 OsclGetHostByNameMethod::~OsclGetHostByNameMethod ()

#### 7.141.2 Member Function Documentation

##### 7.141.2.1 TPVDNSEvent OsclGetHostByNameMethod::GetHostByName (char \* name, OsclNetworkAddress \* addr, int32 aTimeout, Oscl\_-Vector< OsclNetworkAddress, OsclMemAllocator > \* aAddressList)

##### 7.141.2.2 static OsclGetHostByNameMethod\* OsclGetHostByNameMethod::NewL (Oscl\_DefAlloc &a, OsclDNSI \*aDNS, OsclDNSObserver \*aObserver, uint32 aId) [static]

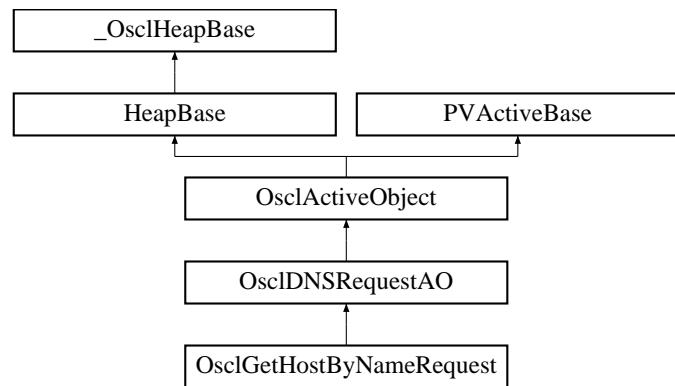
The documentation for this class was generated from the following file:

- `oscl_dns_gethostbyname.h`

## 7.142 OsclGetHostByNameRequest Class Reference

```
#include <oscl_dns_gethostbyname.h>
```

Inheritance diagram for OsclGetHostByNameRequest:



### Friends

- class [OsclGetHostByNameMethod](#)

#### 7.142.1 Friends And Related Function Documentation

##### 7.142.1.1 friend class OsclGetHostByNameMethod [friend]

The documentation for this class was generated from the following file:

- [oscl\\_dns\\_gethostbyname.h](#)

## 7.143 OsclInit Class Reference

```
#include <oscl_init.h>
```

### Static Public Member Functions

- static OSCL\_IMPORT\_REF void [Init](#) (int32 &aError, const [OsclSelect](#) \*aSelect=NULL)
- static OSCL\_IMPORT\_REF void [Cleanup](#) (int32 &aError, const [OsclSelect](#) \*aSelect=NULL)

#### 7.143.1 Detailed Description

Per-thread oscl initialization and cleanup.

#### 7.143.2 Member Function Documentation

##### 7.143.2.1 static OSCL\_IMPORT\_REF void OsclInit::Cleanup (int32 & aError, const OsclSelect \* aSelect = NULL) [static]

This routine cleans up the Oscl modules in the calling thread.

#### Parameters

*err*,: (output) error code of any leave that occurs in initialization.

*config*,: (input param) optional set of initialization parameters. If null, then full initialization with default parameters will be performed. For proper cleanup, the parameters should match the ones used during the Init call.

##### 7.143.2.2 static OSCL\_IMPORT\_REF void OsclInit::Init (int32 & aError, const OsclSelect \* aSelect = NULL) [static]

This routine initializes the Oscl modules in the calling thread.

#### Parameters

*err*,: (output) error code of any leave that occurs in initialization.

*config*,: (input param) optional set of initialization parameters. If null, then full initialization with default parameters will be performed.

The documentation for this class was generated from the following file:

- [oscl\\_init.h](#)

## 7.144 OsclInteger64Transport Struct Reference

```
#include <oscl_int64_utils.h>
```

### Data Fields

- uint32 **iHigh**
- uint32 **iLow**

#### 7.144.1 Detailed Description

##### [OsclInteger64Transport](#) Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

#### 7.144.2 Field Documentation

##### 7.144.2.1 uint32 [OsclInteger64Transport::iHigh](#)

##### 7.144.2.2 uint32 [OsclInteger64Transport::iLow](#)

The documentation for this struct was generated from the following file:

- [oscl\\_int64\\_utils.h](#)

## 7.145 OsclIpMReq Class Reference

```
#include <oscl_socket_types.h>
```

### Public Member Functions

- [OsclIpMReq \(const char \\*intrfcAddr, const char \\*multcstAddr\)](#)

### Data Fields

- [OsclNameString< PVNETWORKADDRESS\\_LEN > interfaceAddr](#)
- [OsclNameString< PVNETWORKADDRESS\\_LEN > multicastAddr](#)

#### 7.145.1 Constructor & Destructor Documentation

**7.145.1.1 OsclIpMReq::OsclIpMReq (const char \* *intrfcAddr*, const char \* *multcstAddr*)**  
[[inline](#)]

#### 7.145.2 Field Documentation

**7.145.2.1 OsclNameString<PVNETWORKADDRESS\_LEN> OsclIpMReq::interfaceAddr**

**7.145.2.2 OsclNameString<PVNETWORKADDRESS\_LEN> OsclIpMReq::multicastAddr**

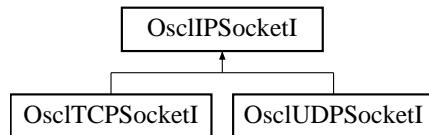
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_types.h](#)

## 7.146 OsclIPSocketI Class Reference

```
#include <oscl_ip_socket.h>
```

Inheritance diagram for OsclIPSocketI:



### Public Member Functions

- int32 [Bind \(OsclNetworkAddress &aAddress\)](#)
- int32 [Join \(OsclNetworkAddress &aAddress\)](#)
- int32 [SetRecvBufferSize \(uint32 size\)](#)
- int32 [SetOptionToReuseAddress \(\)](#)
- int32 [SetTOS \(const OsclSocketTOS &aTOS\)](#)
- int32 [GetPeerName \(OsclNetworkAddress &peerName\)](#)
- virtual int32 [Close \(\)=0](#)
- virtual uint8 \* [GetRecvData \(int32 \\*aLength\)=0](#)
- virtual uint8 \* [GetSendData \(int32 \\*aLength\)=0](#)
- virtual [~OsclIPSocketI \(\)](#)
- void [ThreadLogoff \(\)](#)
- void [ThreadLogon \(OsclSocketObserver \\*aObs, OsclSocketServI \\*aServ\)](#)
- [OsclSocketServI \\* SocketServ \(\)](#)
- [Oscl\\_DefAlloc & Alloc \(\)](#)

### Protected Member Functions

- [OsclIPSocketI \(Oscl\\_DefAlloc &a\)](#)
- void [ConstructL \(OsclSocketObserver \\*aObs, OsclSocketI \\*aSock, OsclSocketServI \\*aServ, uint32 aId\)](#)

### Protected Attributes

- [Oscl\\_DefAlloc & iAlloc](#)
- [OsclNetworkAddress iAddress](#)
- uint32 [iId](#)
- [OsclSocketObserver \\* iObserver](#)
- [OsclSocketI \\* iSocket](#)
- [OsclSocketServI \\* iSocketServ](#)
- [PVLogger \\* iLogger](#)

### Friends

- class [OsclSocketRequestAO](#)
- class [OsclSocketMethod](#)

### 7.146.1 Constructor & Destructor Documentation

7.146.1.1 `virtual OsclIPSocketI::~OsclIPSocketI () [inline, virtual]`

7.146.1.2 `OsclIPSocketI::OsclIPSocketI (Oscl_DefAlloc & a) [inline, protected]`

### 7.146.2 Member Function Documentation

7.146.2.1 `Oscl_DefAlloc& OsclIPSocketI::Alloc () [inline]`

References iAlloc.

Referenced by OsclSocketRequestAO::Alloc(), and OsclSocketMethod::Alloc().

7.146.2.2 `int32 OsclIPSocketI::Bind (OsclNetworkAddress & aAddress)`

7.146.2.3 `virtual int32 OsclIPSocketI::Close () [pure virtual]`

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

7.146.2.4 `void OsclIPSocketI::ConstructL (OsclSocketObserver * aObs, OsclSocketI * aSock, OsclSocketServI * aServ, uint32 aId) [protected]`

7.146.2.5 `int32 OsclIPSocketI::GetPeerName (OsclNetworkAddress & peerName)`

7.146.2.6 `virtual uint8* OsclIPSocketI::GetRecvData (int32 * aLength) [pure virtual]`

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

7.146.2.7 `virtual uint8* OsclIPSocketI::GetSendData (int32 * aLength) [pure virtual]`

Implemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

7.146.2.8 `int32 OsclIPSocketI::Join (OsclNetworkAddress & aAddress)`

7.146.2.9 `int32 OsclIPSocketI::SetOptionToReuseAddress ()`

7.146.2.10 `int32 OsclIPSocketI::SetRecvBufferSize (uint32 size)`

7.146.2.11 `int32 OsclIPSocketI::SetTOS (const OsclSocketTOS & aTOS)`

7.146.2.12 `OsclSocketServI* OsclIPSocketI::SocketServ () [inline]`

References iSocketServ.

7.146.2.13 `void OsclIPSocketI::ThreadLogoff ()`

Reimplemented in [OsclTCPSocketI](#), and [OsclUDPSocketI](#).

**7.146.2.14 void OsclIPSocketI::ThreadLogon (OsclSocketObserver \* *aObs*, OsclSocketServI \* *aServ*)**

### 7.146.3 Friends And Related Function Documentation

**7.146.3.1 friend class OsclSocketMethod [friend]**

**7.146.3.2 friend class OsclSocketRequestAO [friend]**

### 7.146.4 Field Documentation

**7.146.4.1 OsclNetworkAddress OsclIPSocketI::iAddress [protected]**

Referenced by OsclUDPSocketI::BindAsync(), and OsclTCPSocketI::BindAsync().

**7.146.4.2 Oscl\_DefAlloc& OsclIPSocketI::iAlloc [protected]**

Referenced by Alloc().

**7.146.4.3 uint32 OsclIPSocketI::iId [protected]**

Referenced by OsclSocketRequestAO::Id().

**7.146.4.4 PVLogger\* OsclIPSocketI::iLogger [protected]**

**7.146.4.5 OsclSocketObserver\* OsclIPSocketI::iObserver [protected]**

Referenced by OsclTCPSocketI::Accept(), OsclUDPSocketI::BindAsync(), OsclTCPSocketI::BindAsync(), OsclTCPSocketI::Connect(), OsclTCPSocketI::ListenAsync(), OsclTCPSocketI::Recv(), OsclUDPSocketI::RecvFrom(), OsclTCPSocketI::Send(), OsclUDPSocketI::SendTo(), OsclTCPSocketI::Shutdown(), and OsclSocketRequestAO::SocketObserver().

**7.146.4.6 OsclSocketI\* OsclIPSocketI::iSocket [protected]**

Referenced by OsclTCPSocketI::Listen(), and OsclSocketRequestAO::SocketI().

**7.146.4.7 OsclSocketServI\* OsclIPSocketI::iSocketServ [protected]**

Referenced by SocketServ().

The documentation for this class was generated from the following file:

- [oscl\\_ip\\_socket.h](#)

## 7.147 OsclJump Class Reference

```
#include <oscl_error_imp_jumps.h>
```

### Public Member Functions

- void [Jump](#) (int a)
- jmp\_buf \* [Top](#) ()
- [~OsclJump](#) ()

### Static Public Member Functions

- static OSCL\_IMPORT\_REF void [StaticJump](#) (int a)

### Friends

- class [OsclErrorTrapImp](#)

#### 7.147.1 Constructor & Destructor Documentation

##### 7.147.1.1 [OsclJump::~OsclJump \(\)](#) [inline]

References OSCL\_ASSERT.

#### 7.147.2 Member Function Documentation

##### 7.147.2.1 [void OsclJump::Jump \(int a\)](#) [inline]

References \_OSCL\_Abort(), OSCL\_ASSERT, and Top().

##### 7.147.2.2 [static OSCL\\_IMPORT\\_REF void OsclJump::StaticJump \(int a\)](#) [static]

##### 7.147.2.3 [jmp\\_buf\\* OsclJump::Top \(\)](#) [inline]

References OSCL\_ASSERT.

Referenced by [Jump\(\)](#).

#### 7.147.3 Friends And Related Function Documentation

##### 7.147.3.1 [friend class OsclErrorTrapImp](#) [friend]

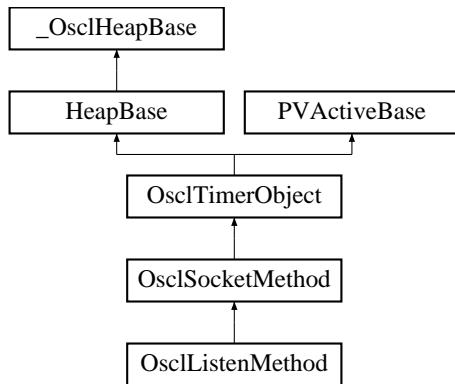
The documentation for this class was generated from the following file:

- [oscl\\_error\\_imp\\_jumps.h](#)

## 7.148 OsclListenMethod Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OsclListenMethod:



### Public Member Functions

- [~OsclListenMethod \(\)](#)
- [TPVSocketEvent Listen \(uint32 qsize, int32 aTimeout\)](#)
- [OsclListenRequest \\* ListenRequest \(\)](#)

### Static Public Member Functions

- static [OsclListenMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 7.148.1 Constructor & Destructor Documentation

##### 7.148.1.1 OsclListenMethod::~OsclListenMethod ()

#### 7.148.2 Member Function Documentation

##### 7.148.2.1 TPVSocketEvent OsclListenMethod::Listen (uint32 *qsize*, int32 *aTimeout*)

Referenced by OsclTCPSocketI::ListenAsync().

##### 7.148.2.2 OsclListenRequest\* OsclListenMethod::ListenRequest () [inline]

References OsclSocketMethod::iSocketRequestAO.

##### 7.148.2.3 static OsclListenMethod\* OsclListenMethod::NewL (OsclIPSocketI & *c*) [static]

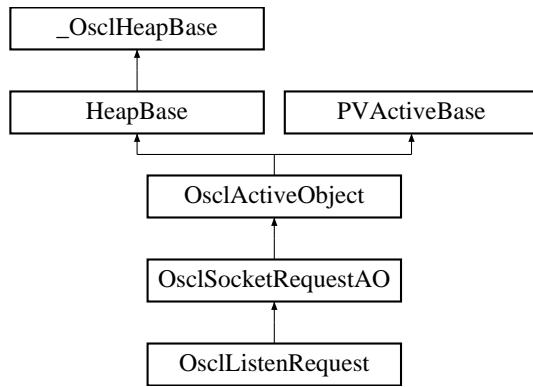
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_listen.h](#)

## 7.149 OsclListenRequest Class Reference

```
#include <oscl_socket_listen.h>
```

Inheritance diagram for OsclListenRequest:



### Public Member Functions

- [OsclListenRequest \(OsclSocketMethod &c\)](#)
- void [Listen \(uint32 qsize\)](#)

#### 7.149.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.149.2 Constructor & Destructor Documentation

**7.149.2.1 OsclListenRequest::OsclListenRequest (OsclSocketMethod & c) [inline]**

#### 7.149.3 Member Function Documentation

**7.149.3.1 void OsclListenRequest::Listen (uint32 qsize)**

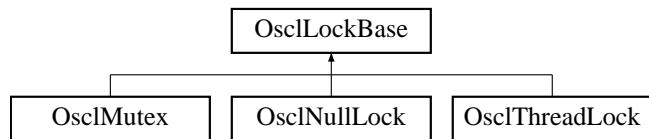
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_listen.h](#)

## 7.150 OsclLockBase Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclLockBase:



### Public Member Functions

- virtual void [Lock \(\)=0](#)
- virtual void [Unlock \(\)=0](#)
- virtual [~OsclLockBase \(\)](#)

#### 7.150.1 Constructor & Destructor Documentation

7.150.1.1 virtual OsclLockBase::[~OsclLockBase \(\)](#) [inline, [virtual](#)]

#### 7.150.2 Member Function Documentation

7.150.2.1 virtual void OsclLockBase::[Lock \(\)](#) [[pure virtual](#)]

Implemented in [OsclNullLock](#), [OsclMutex](#), and [OsclThreadLock](#).

7.150.2.2 virtual void OsclLockBase::[Unlock \(\)](#) [[pure virtual](#)]

Implemented in [OsclNullLock](#), [OsclMutex](#), and [OsclThreadLock](#).

The documentation for this class was generated from the following file:

- [oscl\\_lock\\_base.h](#)

## 7.151 OsclMem Class Reference

```
#include <oscl_mem.h>
```

### Static Public Member Functions

- static OSCL\_IMPORT\_REF void [Init \(\)](#)
- static OSCL\_IMPORT\_REF void [Cleanup \(\)](#)

#### 7.151.1 Member Function Documentation

##### 7.151.1.1 static OSCL\_IMPORT\_REF void OsclMem::Cleanup () [static]

Per-thread cleanup of Oscl Memory

###### Exceptions

Leaves on error;

##### 7.151.1.2 static 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.

###### Exceptions

Leaves on error

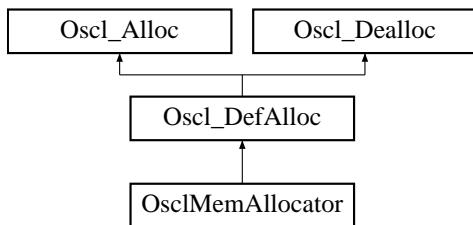
The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 7.152 OsclMemAllocator Class Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemAllocator:



### Public Member Functions

- [OsclAny \\* allocate \(const uint32 n\)](#)
- [void deallocate \(OsclAny \\*p\)](#)

#### 7.152.1 Detailed Description

A simple allocator class. Configurable as to whether this goes through the memory manager or not.

#### 7.152.2 Member Function Documentation

##### 7.152.2.1 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](#).

References `_oscl_malloc()`, `Oscl_DefAlloc::allocate_f()`, `OsclError::LeaveIfNull()`, `NULL`, and `oscl_memset()`.

Referenced by `OsclMemAllocDestructDealloc< T >::allocate()`.

##### 7.152.2.2 void OsclMemAllocator::deallocate (OsclAny \*p) [inline, virtual]

Implements [Oscl\\_DefAlloc](#).

References `OSCL_FREE`.

Referenced by `OsclMemAllocDestructDealloc< T >::deallocate()`.

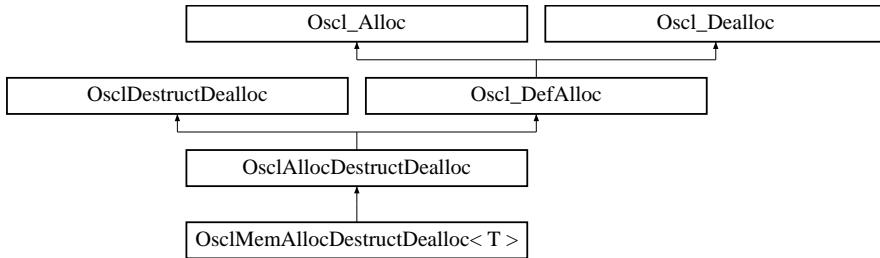
The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 7.153 OsclMemAllocDestructDealloc< T > Class Template Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemAllocDestructDealloc< T >:



### Public Member Functions

- [OsclAny \\* allocate \(const uint32 size\)](#)
- void [deallocate \(OsclAny \\*p\)](#)
- void [destruct\\_and\\_dealloc \(OsclAny \\*p\)](#)

#### 7.153.1 Detailed Description

**template<class T> class OsclMemAllocDestructDealloc< T >**

An [OsclAllocDestructDealloc](#) class that uses [OsclMemAllocator](#).

#### 7.153.2 Member Function Documentation

**7.153.2.1 template<class T > OsclAny\* OsclMemAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

References [OsclMemAllocator::allocate\(\)](#), [Oscl\\_DefAlloc::allocate\\_fl\(\)](#), and [NULL](#).

**7.153.2.2 template<class T > void OsclMemAllocDestructDealloc< T >::deallocate (OsclAny \* p) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

References [OsclMemAllocator::deallocate\(\)](#).

Referenced by [OsclMemAllocDestructDealloc< T >::destruct\\_and\\_dealloc\(\)](#).

**7.153.2.3 template<class T > void OsclMemAllocDestructDealloc< T >::destruct\_and\_dealloc (OsclAny \* p) [inline, virtual]**

Implements [OsclDestructDealloc](#).

References OsclMemAllocDestructDealloc< T >::deallocate(), and OSCL\_UNUSED\_ARG.

The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 7.154 OsclMemAudit Class Reference

```
#include <oscl_mem_audit.h>
```

The documentation for this class was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 7.155 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 Member Functions

- **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 Member Functions

- static void **deallocate** (T \*ptr)

## Data Fields

- bool [\\_Ownership](#)

### 7.155.1 Detailed Description

```
template<class T, class _Allocator = Oscl_TAlloc<T, OsclMemAllocator>> class
OSCLMemAutoPtr< T, _Allocator >
```

The oscl\_auto\_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or indirectly) by new. When the oscl\_auto\_ptr expires, its destructor uses delete to free the memory. The purpose of this class is to provide a way to prevent accidental memory leaks in a class or a method, due to "not remembering to delete" variables allocated on the heap. Thus if you assign an address returned by new to an oscl\_auto\_ptr object, you don't have to remember to free the memory later, it will be freed automatically when the object goes out of scope. The oscl\_auto\_ptr is an example of a smart pointer, an object that acts like a pointer, but with additional features. The class is defined so that it acts like a regular pointer in most respects

### 7.155.2 Constructor & Destructor Documentation

**7.155.2.1 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>>
OSCLMemAutoPtr< T, \_Allocator >::OSCLMemAutoPtr (T \* *inPtr* = 0) [inline,
explicit]**

Default constructor Initializes the pointer and takes ownership.

**7.155.2.2 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>>
OSCLMemAutoPtr< T, \_Allocator >::OSCLMemAutoPtr (const OSCLMemAutoPtr<
T > & *Y*) [inline]**

Copy constructor.

Initializes the pointer and takes ownership from another oscl\_auto\_ptr. Note that the other class does NOT own the pointer any longer, and hence it is NOT its responsibility to free it.

**7.155.2.3 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

References OSCLMemAutoPtr< T, \_Allocator >::\_Ownership.

### 7.155.3 Member Function Documentation

**7.155.3.1 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> void
OSCLMemAutoPtr< T, \_Allocator >::allocate (oscl\_memsize\_t *size*) [inline]**

References OSCLMemAutoPtr< T, \_Allocator >::\_Ownership.

**7.155.3.2 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> static void OSCLMemAutoPtr< T, \_Allocator >::deallocate (T \*ptr) [inline, static]**

**7.155.3.3 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.

Referenced by `OSCLMemAutoPtr< T, _Allocator >::operator=()`.

**7.155.3.4 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> T& OSCLMemAutoPtr< T, \_Allocator >::operator\* () const [inline]**

The indirection operator (\*) accesses a value indirectly, through a pointer.

This operator ensures that the `OSCLMemAutoPtr` can be used like the regular pointer that it was initialized with.

**7.155.3.5 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> T\* OSCLMemAutoPtr< T, \_Allocator >::operator-> () const [inline]**

The indirection operator (->) accesses a value indirectly, through a pointer.

This operator ensures that the `OSCLMemAutoPtr` can be used like the regular pointer that it was initialized with.

**7.155.3.6 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 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.

References `OSCLMemAutoPtr< T, _Allocator >::_Ownership`, `OSCLMemAutoPtr< T, _Allocator >::get()`, and `OSCLMemAutoPtr< T, _Allocator >::release()`.

**7.155.3.7 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.

Referenced by OSCLMemAutoPtr< T, \_Allocator >::operator=().

**7.155.3.8 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.

References OSCLMemAutoPtr< T, \_Allocator >::\_Ownership.

**7.155.3.9 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.

References OSCLMemAutoPtr< T, \_Allocator >::\_Ownership.

## 7.155.4 Field Documentation

**7.155.4.1 template<class T, class \_Allocator = Oscl\_TAlloc<T, OsclMemAllocator>> bool OSCLMemAutoPtr< T, \_Allocator >::\_Ownership**

Referenced by OSCLMemAutoPtr< T, \_Allocator >::allocate(), OSCLMemAutoPtr< T, \_Allocator >::operator=(), OSCLMemAutoPtr< T, \_Allocator >::setWithoutOwnership(), OSCLMemAutoPtr< T, \_Allocator >::takeOwnership(), and OSCLMemAutoPtr< T, \_Allocator >::~OSCLMemAutoPtr().

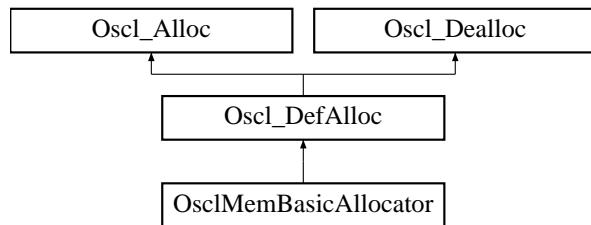
The documentation for this class was generated from the following file:

- [oscl\\_mem\\_auto\\_ptr.h](#)

## 7.156 OsclMemBasicAllocator Class Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemBasicAllocator:



### Public Member Functions

- [OsclAny \\* allocate \(const uint32 n\)](#)
- [void deallocate \(OsclAny \\*p\)](#)

#### 7.156.1 Detailed Description

A simple allocator class that does not use the memory management.

Note: this allocator is for internal use by Oscl only. Higher level code should use [OsclMemAllocator](#).

#### 7.156.2 Member Function Documentation

##### 7.156.2.1 [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](#).

References [\\_oscl\\_malloc\(\)](#), [OsclError::LeaveIfNull\(\)](#), and [oscl\\_memset\(\)](#).

Referenced by [OsclMemBasicAllocDestructDealloc< T >::allocate\(\)](#).

##### 7.156.2.2 [void OsclMemBasicAllocator::deallocate \(OsclAny \\*p\) \[inline, virtual\]](#)

Implements [Oscl\\_DefAlloc](#).

References [\\_oscl\\_free\(\)](#).

Referenced by [OsclMemBasicAllocDestructDealloc< T >::deallocate\(\)](#).

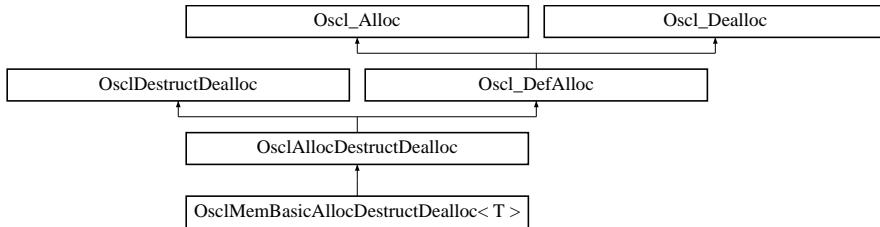
The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 7.157 OsclMemBasicAllocDestructDealloc< T > Class Template Reference

```
#include <oscl_mem.h>
```

Inheritance diagram for OsclMemBasicAllocDestructDealloc< T >:



### Public Member Functions

- [OsclAny \\* allocate \(const uint32 size\)](#)
- [void deallocate \(OsclAny \\*p\)](#)
- [void destruct\\_and\\_dealloc \(OsclAny \\*p\)](#)

#### 7.157.1 Detailed Description

**template<class T> class OsclMemBasicAllocDestructDealloc< T >**

An [OsclAllocDestructDealloc](#) class that uses [OsclMemBasicAllocator](#).

#### 7.157.2 Member Function Documentation

**7.157.2.1 template<class T> OsclAny\* OsclMemBasicAllocDestructDealloc< T >::allocate (const uint32 size) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

References [OsclMemBasicAllocator::allocate\(\)](#), [Oscl\\_DefAlloc::allocate\\_fl\(\)](#), and [NULL](#).

**7.157.2.2 template<class T> void OsclMemBasicAllocDestructDealloc< T >::deallocate (OsclAny \*p) [inline, virtual]**

Implements [Oscl\\_DefAlloc](#).

References [OsclMemBasicAllocator::deallocate\(\)](#).

Referenced by [OsclMemBasicAllocDestructDealloc< T >::destruct\\_and\\_dealloc\(\)](#).

**7.157.2.3 template<class T> void OsclMemBasicAllocDestructDealloc< T >::destruct\_and\_dealloc (OsclAny \*p) [inline, virtual]**

Implements [OsclDestructDealloc](#).

References OsclMemBasicAllocDestructDealloc< T >::deallocate(), and OSCL\_UNUSED\_ARG.

The documentation for this class was generated from the following file:

- [oscl\\_mem.h](#)

## 7.158 OsclMemGlobalAuditObject Class Reference

```
#include <oscl_mem.h>
```

### Public Types

- `typedef OsclMemAudit audit_type`

### Static Public Member Functions

- `static OSCL_IMPORT_REF audit_type * getGlobalMemAuditObject ()`

### Friends

- class `OsclMem`

#### 7.158.1 Member Typedef Documentation

##### 7.158.1.1 `typedef OsclMemAudit OsclMemGlobalAuditObject::audit_type`

#### 7.158.2 Member Function Documentation

##### 7.158.2.1 `static OSCL_IMPORT_REF audit_type* OsclMemGlobalAuditObject::getGlobalMemAuditObject () [static]`

returns the global audit object. For use in macros only-- not a public API.

#### 7.158.3 Friends And Related Function Documentation

##### 7.158.3.1 `friend class OsclMem [friend]`

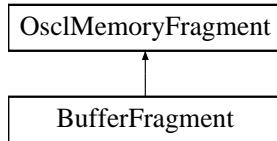
The documentation for this class was generated from the following file:

- `oscl_mem.h`

## 7.159 OsclMemoryFragment Struct Reference

```
#include <oscl_types.h>
```

Inheritance diagram for OsclMemoryFragment:



### Data Fields

- uint32 [len](#)
- void \* [ptr](#)

#### 7.159.1 Field Documentation

##### 7.159.1.1 uint32 OsclMemoryFragment::len

Referenced by BufFragGroup< ChainClass, max\_frags >::AddFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::GetLocalFragment(), and MediaData< ChainClass, max\_frags, local\_bufsize >::IsLocalData().

##### 7.159.1.2 void\* OsclMemoryFragment::ptr

Referenced by BufFragGroup< ChainClass, max\_frags >::AddFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::AddLocalFragment(), MediaData< ChainClass, max\_frags, local\_bufsize >::GetLocalFragment(), and MediaData< ChainClass, max\_frags, local\_bufsize >::IsLocalData().

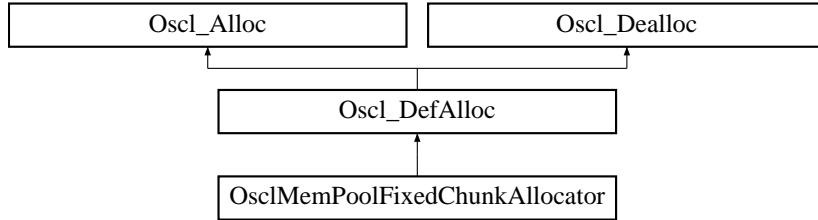
The documentation for this struct was generated from the following file:

- [oscl\\_types.h](#)

## 7.160 OsclMemPoolFixedChunkAllocator Class Reference

```
#include <oscl_mem_mempool.h>
```

Inheritance diagram for OsclMemPoolFixedChunkAllocator:



### Public Member Functions

- OSCL\_IMPORT\_REF OsclMemPoolFixedChunkAllocator (const uint32 numchunk=1, const uint32 chunksize=0, Oscl\_DefAlloc \*gen\_alloc=NULL, const uint32 chunkalignment=0)
- virtual OSCL\_IMPORT\_REF void enablenullpointerreturn ()
- virtual OSCL\_IMPORT\_REF ~OsclMemPoolFixedChunkAllocator ()
- virtual OSCL\_IMPORT\_REF OsclAny \* allocate (const uint32 n)
- virtual OSCL\_IMPORT\_REF void deallocate (OsclAny \*p)
- virtual OSCL\_IMPORT\_REF void notifyfreechunkavailable (OsclMemPoolFixedChunkAllocatorObserver &obs, OsclAny \*aContextData=NULL)
- virtual OSCL\_IMPORT\_REF void CancelFreeChunkAvailableCallback ()
- OSCL\_IMPORT\_REF void addRef ()
- OSCL\_IMPORT\_REF void removeRef ()

### Protected Member Functions

- virtual OSCL\_IMPORT\_REF void createmempool ()
- virtual OSCL\_IMPORT\_REF void destroymempool ()

### Protected Attributes

- uint32 iNumChunk
- uint32 iChunkSize
- uint32 iChunkSizeMemAligned
- uint32 iChunkAlignment
- Oscl\_DefAlloc \* iMemPoolAllocator
- OsclAny \* iMemPool
- OsclAny \* iMemPoolAligned
- Oscl\_Vector< OsclAny \*, OsclMemAllocator > iFreeMemChunkList
- bool iCheckNextAvailableFreeChunk
- OsclMemPoolFixedChunkAllocatorObserver \* iObserver
- OsclAny \* iNextAvailableContextData
- int32 iRefCount
- bool iEnableNullPtrReturn

## 7.160.1 Constructor & Destructor Documentation

**7.160.1.1 OSCL\_IMPORT\_REF OsclMemPoolFixedChunkAllocator::OsclMemPoolFixedChunkAllocator (const uint32 numchunk = 1, const uint32 chunkszie = 0, Oscl\_DefAlloc \*gen\_alloc = NULL, const uint32 chunkalignment = 0)**

This API throws an exception when the memory allocation for pool fails If numchunk and chunkszie parameters are not set, memory pool of 1 chunk will be created in the first call to allocate. The chunk size will be set to the n passed in for [allocate\(\)](#). If numchunk parameter is set to 0, the memory pool will use 1 for numchunk. If chunkalignment is set to 0, memory pool will use default allocator alignment (8-byte) If chunkalignment is > 0, memory pool will align all buffers in the mempool to the specified alignment. Alignment should be a power of 2

### Returns

void

**7.160.1.2 virtual OSCL\_IMPORT\_REF OsclMemPoolFixedChunkAllocator::~OsclMemPoolFixedChunkAllocator () [virtual]**

The destructor for the memory pool

## 7.160.2 Member Function Documentation

**7.160.2.1 OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::addRef ()**

Increments the reference count for this memory pool allocator

### Returns

void

**7.160.2.2 virtual OSCL\_IMPORT\_REF OsclAny\* OsclMemPoolFixedChunkAllocator::allocate (const uint32 n) [virtual]**

This API throws an exception when n is greater than the fixed chunk size or there are no free chunk available in the pool, if "enablenullpointerreturn" has not been called. If the memory pool hasn't been created yet, the pool will be created with chunk size equal to n so n must be greater than 0. Exception will be thrown if memory allocation for the memory pool fails.

### Returns

pointer to available chunk from memory pool

Implements [Oscl\\_DefAlloc](#).

**7.160.2.3 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::CancelFreeChunkAvailableCallback () [virtual]**

This API will cancel any past callback requests..

**Returns**

void

**7.160.2.4 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::createmempool()** [protected, virtual]

**7.160.2.5 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::deallocate(OsclAny \*p)** [virtual]

This API throws an exception when the pointer p passed in is not part of the memory pool. Exception will be thrown if the memory pool is not set up yet.

**Returns**

void

Implements [Oscl\\_DefAlloc](#).

**7.160.2.6 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::destroymempool()** [protected, virtual]

**7.160.2.7 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::enablenullpointerreturn()** [virtual]

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "OSCL\_LEAVE(OsclErrNoResources)" allocate API will return NULL.

**Returns**

void

**7.160.2.8 virtual OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::notifyfreechunkavailable(OsclMemPoolFixedChunkAllocatorObserver & obs, OsclAny \* aContextData = NULL)** [virtual]

This API will set the flag to send a callback via specified observer object when the next memory chunk is deallocated by [deallocate\(\)](#) call..

**Returns**

void

**7.160.2.9 OSCL\_IMPORT\_REF void OsclMemPoolFixedChunkAllocator::removeRef()**

Decrements the reference count for this memory pool allocator When the reference count goes to 0, this instance of the memory pool object is deleted

**Returns**

void

### 7.160.3 Field Documentation

- 7.160.3.1 `bool OsclMemPoolFixedChunkAllocator::iCheckNextAvailableFreeChunk [protected]`
- 7.160.3.2 `uint32 OsclMemPoolFixedChunkAllocator::iChunkAlignment [protected]`
- 7.160.3.3 `uint32 OsclMemPoolFixedChunkAllocator::iChunkSize [protected]`
- 7.160.3.4 `uint32 OsclMemPoolFixedChunkAllocator::iChunkSizeMemAligned [protected]`
- 7.160.3.5 `bool OsclMemPoolFixedChunkAllocator::iEnableNullPtrReturn [protected]`
- 7.160.3.6 `Oscl_Vector<OsclAny*, OsclMemAllocator> OsclMemPoolFixedChunkAllocator::iFreeMemChunkList [protected]`
- 7.160.3.7 `OsclAny* OsclMemPoolFixedChunkAllocator::iMemPool [protected]`
- 7.160.3.8 `OsclAny* OsclMemPoolFixedChunkAllocator::iMemPoolAligned [protected]`
- 7.160.3.9 `Oscl_DefAlloc* OsclMemPoolFixedChunkAllocator::iMemPoolAllocator [protected]`
- 7.160.3.10 `OsclAny* OsclMemPoolFixedChunkAllocator::iNextAvailableContextData [protected]`
- 7.160.3.11 `uint32 OsclMemPoolFixedChunkAllocator::iNumChunk [protected]`
- 7.160.3.12 `OsclMemPoolFixedChunkAllocatorObserver* OsclMemPoolFixedChunkAllocator::iObserver [protected]`
- 7.160.3.13 `int32 OsclMemPoolFixedChunkAllocator::iRefCount [protected]`

The documentation for this class was generated from the following file:

- [oscl\\_mem\\_mempool.h](#)

## 7.161 OsclMemPoolFixedChunkAllocatorObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

### Public Member Functions

- virtual void [freechunkavailable \(OsclAny \\*aContextData\)=0](#)
- virtual [~OsclMemPoolFixedChunkAllocatorObserver \(\)](#)

#### 7.161.1 Constructor & Destructor Documentation

##### 7.161.1.1 virtual

```
OsclMemPoolFixedChunkAllocatorObserver::~OsclMemPoolFixedChunkAllocatorObserver  
() [inline, virtual]
```

#### 7.161.2 Member Function Documentation

##### 7.161.2.1 virtual void OsclMemPoolFixedChunkAllocatorObserver::freechunkavailable (OsclAny \* *aContextData*) [pure virtual]

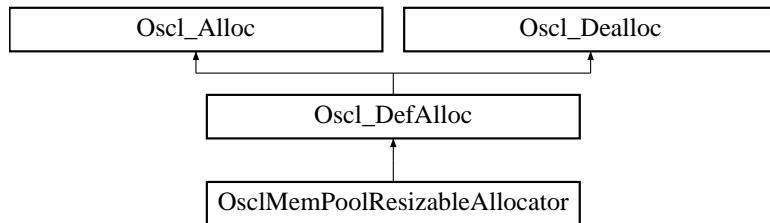
The documentation for this class was generated from the following file:

- [oscl\\_mem\\_mempool.h](#)

## 7.162 OsclMemPoolResizableAllocator Class Reference

```
#include <oscl_mem_mempool.h>
```

Inheritance diagram for OsclMemPoolResizableAllocator:



### Data Structures

- struct [MemPoolBlockInfo](#)
- struct [MemPoolBufferInfo](#)

### Public Member Functions

- 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 Member Functions

- virtual ~[OsclMemPoolResizableAllocator](#) ()
- [MemPoolBufferInfo](#) \* [addnewmempoolbuffer](#) (uint32 aBufferSize)
- void [destroyallmempoolbuffers](#) ()
- [MemPoolBlockInfo](#) \* [findfreeblock](#) (uint32 aBlockSize)

- `OsclAny * allocateblock (MemPoolBlockInfo &aBlockPtr, uint32 aNumBytes)`
- `void deallocateblock (MemPoolBlockInfo &aBlockPtr)`
- `bool validateblock (OsclAny *aBlockBufPtr)`
- `uint32 getMemPoolBufferSize (MemPoolBufferInfo *aBufferInfo) const`
- `uint32 getMemPoolBufferAllocatedSize (MemPoolBufferInfo *aBufferInfo) const`
- `uint32 memoryPoolBufferMgmtOverhead () const`

## Protected Attributes

- `uint32 iMemPoolBufferSize`
- `uint32 iMemPoolBufferNumLimit`
- `uint32 iExpectedNumBlocksPerBuffer`
- `uint32 iMaxNewMemPoolBufferSz`
- `Oscl_DefAlloc * iMemPoolBufferAllocator`
- `Oscl_Vector< MemPoolBufferInfo *, OsclMemAllocator > iMemPoolBufferList`
- `uint32 iMemPoolPrevAllocBufferIndex`
- `uint32 iBufferInfoAlignedSize`
- `uint32 iBlockInfoAlignedSize`
- `bool iCheckNextAvailable`
- `uint32 iRequestedNextAvailableSize`
- `OsclAny * iNextAvailableContextData`
- `OsclMemPoolResizableAllocatorObserver * iObserver`
- `bool iCheckFreeMemoryAvailable`
- `uint32 iRequestedAvailableFreeMemSize`
- `OsclAny * iFreeMemContextData`
- `OsclMemPoolResizableAllocatorMemoryObserver * iFreeMemPoolObserver`
- `int32 iRefCount`
- `bool iEnableNullPtrReturn`
- `OsclMemPoolResizableAllocatorLogger * iDebugLogger`

### 7.162.1 Constructor & Destructor Documentation

**7.162.1.1 OSCL\_IMPORT\_REF OsclMemPoolResizableAllocator::OsclMemPoolResizableAllocator (uint32 *aMemPoolBufferSize*,  
*aMemPoolBufferNumLimit* = 0, *aExpectedNumBlocksPerBuffer* = 0,  
*Oscl\_DefAlloc* \* *gen\_alloc* = NULL)**

Create the memory pool allocator with resizing functionality. The size of the memory pool buffer needs to be passed-in. The maximum number of memory pool buffers, expected number of blocks in a memory pool buffer, and outside allocator are optional. This API throws an exception when the memory allocation for the pool buffer fails. If memory pool buffer number limit parameter is not set, the assumption is that there is no limit and memory pool will grow as needed. If the expected number of blocks is not set or not known, the memory pool will use a default value to 10 to allocate extra memory for the block info header.

#### Returns

`void`

**7.162.1.2 virtual OsclMemPoolResizableAllocator::~OsclMemPoolResizableAllocator ()  
[protected, virtual]**

The destructor for the memory pool. Should not be called directly. Use [removeRef\(\)](#) instead.

**7.162.2 Member Function Documentation****7.162.2.1 MemPoolBufferInfo\* OsclMemPoolResizableAllocator::addnewmempoolbuffer (uint32  
aBufferSize) [protected]****7.162.2.2 OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::addRef ()**

Increments the reference count for this memory pool allocator

**Returns**

void

**7.162.2.3 virtual OSCL\_IMPORT\_REF OsclAny\* OsclMemPoolResizableAllocator::allocate  
(const uint32 aNumBytes) [virtual]**

Allocates a block from the memory pool that is at least in size requested This API throws an exception if there isn't enough memory (if "enablenullpointerreturn" has not been called) for the requested amount in the pool or if the extra pool buffer cannot be allocated.

**Returns**

Pointer to memory buffer from memory pool

Implements [Oscl\\_DefAlloc](#).

**7.162.2.4 OsclAny\* OsclMemPoolResizableAllocator::allocateblock (MemPoolBlockInfo &  
aBlockPtr, uint32 aNumBytes) [protected]****7.162.2.5 virtual OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::CancelFreeChunkAvailableCallback () [virtual]**

This API will cancel any past callback requests..

**Returns**

void

**7.162.2.6 OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::CancelFreeMemoryAvailableCallback ()****7.162.2.7 virtual OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::deallocate  
(OsclAny \* aPtr) [virtual]**

Deallocates and returns a block back to the memory pool This API throws an exception if the pointer passed in is not part of the memory pool, aligned, or has corrupted block header.

**Returns**

void

Implements [Oscl\\_DefAlloc](#).

**7.162.2.8 void OsclMemPoolResizableAllocator::deallocateblock (MemPoolBlockInfo & aBlockPtr) [protected]**

**7.162.2.9 void OsclMemPoolResizableAllocator::destroyallmempoolbuffers () [protected]**

**7.162.2.10 virtual OSCL\_IMPORT\_REF void OsclMemPoolResizableAllocator::enablenullpointerreturn () [virtual]**

This API will disable exceptions in case the memory pool runs out of memory Instead of doing "OSCL\_LEAVE(OsclErrNoResources)" allocate API will return NULL.

**Returns**

void

**7.162.2.11 MemPoolBlockInfo\* OsclMemPoolResizableAllocator::findfreeblock (uint32 aBlockSize) [protected]**

**7.162.2.12 virtual OSCL\_IMPORT\_REF uint32 OsclMemPoolResizableAllocator::getAllocatedSize () const [virtual]**

Returns the number of bytes allocated from the buffer<including the="" overhead="" bytes="" that="" may="" be="" allocated="" by="" the="" allocater="" to="" keep="" track="" of="" the="" chunks="" allocated>="">

**7.162.2.13 virtual OSCL\_IMPORT\_REF uint32 OsclMemPoolResizableAllocator::getAvailableSize () const [virtual]**

Returns the number of bytes available with the buffer

**7.162.2.14 OSCL\_IMPORT\_REF uint32 OsclMemPoolResizableAllocator::getBufferSize () const**

Returns the size of the buffer <including the="" overhead="" bytes="" that="" may="" be="" allocated="" by="" the="" allocater>="">

**7.162.2.15 virtual OSCL\_IMPORT\_REF uint32 OsclMemPoolResizableAllocator::getLargestContiguousFreeBlockSize () const [virtual]**

Returns the size of the largest available chunk in the memory.

- 7.162.2.16** `uint32 OsclMemPoolResizableAllocator::getMemPoolBufferAllocatedSize (MemPoolBufferInfo * aBufferInfo) const [protected]`
- 7.162.2.17** `uint32 OsclMemPoolResizableAllocator::getMemPoolBufferSize (MemPoolBufferInfo * aBufferInfo) const [protected]`
- 7.162.2.18** `uint32 OsclMemPoolResizableAllocator::memoryPoolBufferMgmtOverhead () const [protected]`
- 7.162.2.19** `virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::notifyfreeblockavailable (OsclMemPoolResizableAllocatorObserver & aObserver, uint32 aRequestedSize = 0, OsclAny * aContextData = NULL) [virtual]`

This API will set the flag to send a callback via specified observer object when the next memory block is deallocated by `deallocate()` call. If the optional requested size parameter is set, the callback is sent when a free memory space of requested size becomes available. The optional context data is returned with the callback and can be used by the user to differentiate between different instances of memory pool objects. This memory pool only allows one notify to be queued. Another call to this function will just overwrite the previous call.

#### Returns

`void`

- 7.162.2.20** `virtual OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::notifyfreememoryavailable (OsclMemPoolResizableAllocatorMemoryObserver & aObserver, uint32 aRequestedSize = 0, OsclAny * aContextData = NULL) [virtual]`

- 7.162.2.21** `OSCL_IMPORT_REF void OsclMemPoolResizableAllocator::removeRef ()`

Decrements the reference count for this memory pool allocator. When the reference count goes to 0, this instance of the memory pool object is deleted.

#### Returns

`void`

- 7.162.2.22** `virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::setMaxSzForNewMemPoolBuffer (uint32 aMaxNewMemPoolBufferSz) [virtual]`

- 7.162.2.23** `virtual OSCL_IMPORT_REF bool OsclMemPoolResizableAllocator::trim (OsclAny * aPtr, uint32 aBytesToFree) [virtual]`

Returns a tail segment of a previously allocated memory block back to the memory pool. The passed-in pointer to the memory buffer is still valid after the call completes but the buffer size is smaller by the specified amount that was freed. This function allows the user to allocate a larger size block initially when the amount needed is unknown and then return the unused portion of the block when the amount becomes known. This API throws an exception if the pointer passed in is not part of the memory pool or the size to return is bigger than the size of the passed-in block. Exception will be thrown if the memory pool is not set up yet.

**Returns**

bool True if trim operation successful. False if the block wasn't trimmed



---

**7.162.2.24** `bool OsclMemPoolResizableAllocator::validateblock (OsclAny * aBlockBufPtr)` [protected]

### 7.162.3 Field Documentation

**7.162.3.1** `uint32 OsclMemPoolResizableAllocator::iBlockInfoAlignedSize` [protected]

**7.162.3.2** `uint32 OsclMemPoolResizableAllocator::iBufferInfoAlignedSize` [protected]

**7.162.3.3** `bool OsclMemPoolResizableAllocator::iCheckFreeMemoryAvailable` [protected]

**7.162.3.4** `bool OsclMemPoolResizableAllocator::iCheckNextAvailable` [protected]

**7.162.3.5** `OsclMemPoolResizableAllocatorLogger* OsclMemPoolResizableAllocator::iDebugLogger` [protected]

**7.162.3.6** `bool OsclMemPoolResizableAllocator::iEnableNullPtrReturn` [protected]

**7.162.3.7** `uint32 OsclMemPoolResizableAllocator::iExpectedNumBlocksPerBuffer` [protected]

**7.162.3.8** `OsclAny* OsclMemPoolResizableAllocator::iFreeMemContextData` [protected]

**7.162.3.9** `OsclMemPoolResizableAllocatorMemoryObserver* OsclMemPoolResizableAllocator::iFreeMemPoolObserver` [protected]

**7.162.3.10** `uint32 OsclMemPoolResizableAllocator::iMaxNewMemPoolBufferSz` [protected]

**7.162.3.11** `Oscl_DefAlloc* OsclMemPoolResizableAllocator::iMemPoolBufferAllocator` [protected]

**7.162.3.12** `Oscl_Vector<MemPoolBufferInfo*, OsclMemAllocator> OsclMemPoolResizableAllocator::iMemPoolBufferList` [protected]

**7.162.3.13** `uint32 OsclMemPoolResizableAllocator::iMemPoolBufferNumLimit` [protected]

**7.162.3.14** `uint32 OsclMemPoolResizableAllocator::iMemPoolBufferSize` [protected]

**7.162.3.15** `uint32 OsclMemPoolResizableAllocator::iMemPoolPrevAllocBufferIndex` [protected]

**7.162.3.16** `OsclAny* OsclMemPoolResizableAllocator::iNextAvailableContextData` [protected]

**7.162.3.17** `OsclMemPoolResizableAllocatorObserver* OsclMemPoolResizableAllocator::iObserver` [protected]

**7.162.3.18** `int32 OsclMemPoolResizableAllocator::iRefCount` [protected]

**7.162.3.19** `uint32 OsclMemPoolResizableAllocator::iRequestedAvailableFreeMemSize` [protected]

**7.162.3.20** `uint32 OsclMemPoolResizableAllocator::iRequestedNextAvailableSize` [protected]

- [oscl\\_mem\\_mempool.h](#)

## 7.163 OsclMemPoolResizableAllocatorMemoryObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

### Public Member Functions

- virtual void `freememoryavailable (OsclAny *aContextData)=0`
- virtual `~OsclMemPoolResizableAllocatorMemoryObserver ()`

#### 7.163.1 Constructor & Destructor Documentation

7.163.1.1 virtual  
OsclMemPoolResizableAllocatorMemoryObserver::~OsclMemPoolResizableAllocatorMemoryObserver  
(**[inline, virtual]**)

#### 7.163.2 Member Function Documentation

7.163.2.1 virtual void OsclMemPoolResizableAllocatorMemoryObserver::freememoryavailable  
(OsclAny \* *aContextData*) [**pure virtual**]

The documentation for this class was generated from the following file:

- `oscl_mem_mempool.h`

## 7.164 OsclMemPoolResizableAllocatorObserver Class Reference

```
#include <oscl_mem_mempool.h>
```

### Public Member Functions

- virtual void [freeblockavailable \(OsclAny \\*aContextData\)=0](#)
- virtual [~OsclMemPoolResizableAllocatorObserver \(\)](#)

#### 7.164.1 Constructor & Destructor Documentation

##### 7.164.1.1 virtual

```
OsclMemPoolResizableAllocatorObserver::~OsclMemPoolResizableAllocatorObserver  
() [inline, virtual]
```

#### 7.164.2 Member Function Documentation

##### 7.164.2.1 virtual void OsclMemPoolResizableAllocatorObserver::freeblockavailable (OsclAny \* *aContextData*) [pure virtual]

The documentation for this class was generated from the following file:

- [oscl\\_mem\\_mempool.h](#)

## 7.165 OsclMemStatsNode Class Reference

```
#include <oscl_mem_audit.h>
```

### Public Member Functions

- [OsclMemStatsNode \(\)](#)
- [void reset \(\)](#)
- [~OsclMemStatsNode \(\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size\)](#)
- [void \\* operator new \(oscl\\_memsize\\_t size, OsclMemStatsNode \\*ptr\)](#)
- [void operator delete \(void \\*ptr\) throw \(\)](#)

### Data Fields

- [MM\\_Stats\\_t \\* pMMStats](#)
- [MM\\_FailInsertParam \\* pMMFIParam](#)
- [char \\* tag](#)

#### 7.165.1 Constructor & Destructor Documentation

##### 7.165.1.1 OsclMemStatsNode::OsclMemStatsNode () [inline]

References NULL, pMMFIParam, pMMStats, and tag.

##### 7.165.1.2 OsclMemStatsNode::~OsclMemStatsNode () [inline]

References Oscl\_TAlloc< T, Alloc >::deallocate(), OSCL\_DELETE, pMMFIParam, pMMStats, and tag.

#### 7.165.2 Member Function Documentation

##### 7.165.2.1 void OsclMemStatsNode::operator delete (void \*ptr) throw () [inline]

References Oscl\_TAlloc< T, Alloc >::deallocate().

##### 7.165.2.2 void\* OsclMemStatsNode::operator new (oscl\_memsize\_t size, OsclMemStatsNode \*ptr) [inline]

References OSCL\_UNUSED\_ARG.

##### 7.165.2.3 void\* OsclMemStatsNode::operator new (oscl\_memsize\_t size) [inline]

References Oscl\_TAlloc< T, Alloc >::allocate(), and OSCL\_UNUSED\_ARG.

##### 7.165.2.4 void OsclMemStatsNode::reset () [inline]

References pMMFIParam, pMMStats, MM\_FailInsertParam::reset(), and MM\_Stats\_t::reset().

### 7.165.3 Field Documentation

#### 7.165.3.1 MM\_FailInsertParam\* OsclMemStatsNode::pMMFIParam

Referenced by OsclMemStatsNode(), reset(), and ~OsclMemStatsNode().

#### 7.165.3.2 MM\_Stats\_t\* OsclMemStatsNode::pMMStats

Referenced by OsclMemStatsNode(), reset(), and ~OsclMemStatsNode().

#### 7.165.3.3 char\* OsclMemStatsNode::tag

Referenced by OsclMemStatsNode(), and ~OsclMemStatsNode().

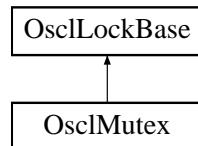
The documentation for this class was generated from the following file:

- [oscl\\_mem\\_audit.h](#)

## 7.166 OsclMutex Class Reference

```
#include <oscl_mutex.h>
```

Inheritance diagram for OsclMutex:



### Public Member Functions

- OSCL\_IMPORT\_REF OsclMutex ()
- virtual OSCL\_IMPORT\_REF ~OsclMutex ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Create (void)
- OSCL\_IMPORT\_REF void Lock ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError TryLock ()
- OSCL\_IMPORT\_REF void Unlock ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Close (void)

#### 7.166.1 Detailed Description

Class [OsclMutex](#)

#### 7.166.2 Constructor & Destructor Documentation

##### 7.166.2.1 OSCL\_IMPORT\_REF OsclMutex::OsclMutex ()

Class constructor

##### 7.166.2.2 virtual OSCL\_IMPORT\_REF OsclMutex::~OsclMutex () [virtual]

Class destructor

#### 7.166.3 Member Function Documentation

##### 7.166.3.1 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclMutex::Close (void)

Closes the Mutex

#### Parameters

*It* wont take any parameters

#### Returns

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

**7.166.3.2 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclMutex::Create (void)**

Creates the Mutex

**Parameters**

*No* input arguments

**Returns**

Returns the Error whether it is success or failure. Incase of failure it will return what is the specific error

**7.166.3.3 OSCL\_IMPORT\_REF void OsclMutex::Lock () [virtual]**

Locks the Mutex

**Parameters**

*It* wont take any parameters

**Returns**

Returns nothing

Implements [OsclLockBase](#).

**7.166.3.4 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclMutex::TryLock ()**

Try to lock the mutex,if the Mutex is already locked calling thread immediately returns with out blocking

**Parameters**

*It* wont take any parameters

**Returns**

Returns SUCCESS\_ERROR if the mutex was acquired, MUTEX\_LOCKED\_ERROR if the mutex cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.166.3.5 OSCL\_IMPORT\_REF void OsclMutex::Unlock () [virtual]**

Releases the Mutex

**Parameters**

*It* wont take any parameters

**Returns**

Returns nothing

Implements [OsclLockBase](#).

The documentation for this class was generated from the following file:

- [oscl\\_mutex.h](#)

## 7.167 OsclNameString< \_\_len > Class Template Reference

```
#include <oscl_namestring.h>
```

### Public Member Functions

- [OsclNameString \(\)](#)
- [OsclNameString \(const char a\[ \]\)](#)
- [OsclNameString \(uint8 \\*a\)](#)
- void [Set \(uint8 \\*a\)](#)
- void [Set \(const char a\[ \]\)](#)
- uint8 \* [Str \(\) const](#)
- int32 [MaxLen \(\) const](#)

### 7.167.1 Detailed Description

**template<int \_\_len> class OsclNameString< \_\_len >**

Name string class appropriate for passing short constant ASCII strings around. All strings are automatically truncated and null-terminated.

### 7.167.2 Constructor & Destructor Documentation

**7.167.2.1 template<int \_\_len> OsclNameString< \_\_len >::OsclNameString () [inline]**

**7.167.2.2 template<int \_\_len> OsclNameString< \_\_len >::OsclNameString (const char a[ ]) [inline]**

**7.167.2.3 template<int \_\_len> OsclNameString< \_\_len >::OsclNameString (uint8 \* a) [inline]**

### 7.167.3 Member Function Documentation

**7.167.3.1 template<int \_\_len> int32 OsclNameString< \_\_len >::MaxLen () const [inline]**

**7.167.3.2 template<int \_\_len> void OsclNameString< \_\_len >::Set (const char a[ ]) [inline]**

**7.167.3.3 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.

Referenced by OsclUDPSocketI::BindAsync(), OsclTCPSocketI::BindAsync(), OsclNameString< PVNETWORKADDRESS\_LEN >::OsclNameString(), GetHostByNameParam::PersistHostAddress(), and OsclNameString< PVNETWORKADDRESS\_LEN >::Set().

**7.167.3.4 template<int \_\_len> uint8\* OsclNameString< \_\_len >::Str () const [inline]**

Referenced by OsclUDPSocketI::BindAsync(), OsclTCPSocketI::BindAsync(), GetHostByNameParam::canPersistMoreHostAddresses(), OsclNetworkAddress::operator==(), and GetHostByNameParam::PersistHostAddress().

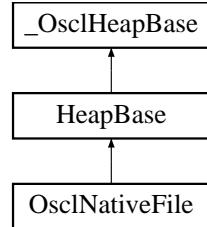
The documentation for this class was generated from the following file:

- [oscl\\_namestring.h](#)

## 7.168 OsclNativeFile Class Reference

```
#include <oscl_file_native.h>
```

Inheritance diagram for OsclNativeFile:



### Public Member Functions

- [OsclNativeFile \(\)](#)
- [~OsclNativeFile \(\)](#)
- [int32 Open \(const OsclFileHandle &, uint32 mode, const OsclNativeFileParams &params, Oscl\\_FileServer &fileserv\)](#)
- [int32 Open \(const oscl\\_wchar \\*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\\_FileServer &fileserv\)](#)
- [int32 Open \(const char \\*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\\_FileServer &fileserv\)](#)
- [uint32 Read \(OsclAny \\*buffer, uint32 size, uint32 numelements\)](#)
- [uint32 Write \(const OsclAny \\*buffer, uint32 size, uint32 numelements\)](#)
- [int32 Seek \(TOsclFileOffset offset, Oscl\\_File::seek\\_type origin\)](#)
- [TOsclFileOffset Tell \(\)](#)
- [int32 Flush \(\)](#)
- [int32 EndOfFile \(\)](#)
- [TOsclFileOffset Size \(\)](#)
- [int32 Close \(\)](#)
- [int32 SetSize \(uint32 size\)](#)
- [uint32 Mode \(\)](#)
- [int32 GetError \(\)](#)
- [int32 ReadAsync \(OsclAny \\*buffer, uint32 size, uint32 numelements, OsclAOStatus &status\)](#)
- [uint32 GetReadAsyncNumElements \(\)](#)
- [bool HasAsyncRead \(\)](#)
- [void ReadAsyncCancel \(\)](#)

### 7.168.1 Constructor & Destructor Documentation

**7.168.1.1 OsclNativeFile::OsclNativeFile ()**

**7.168.1.2 OsclNativeFile::~OsclNativeFile ()**

### 7.168.2 Member Function Documentation

**7.168.2.1 int32 OsclNativeFile::Close ()**

**7.168.2.2 int32 OsclNativeFile::EndOfFile ()**

**7.168.2.3 int32 OsclNativeFile::Flush ()**

**7.168.2.4 int32 OsclNativeFile::GetError ()**

**7.168.2.5 uint32 OsclNativeFile::GetReadAsyncNumElements ()**

Get the number of elements read in the last call to ReadAsync.

#### Returns

: number of elements read.

**7.168.2.6 bool OsclNativeFile::HasAsyncRead ()**

#### Returns

: true if async read is supported natively.

**7.168.2.7 uint32 OsclNativeFile::Mode () [inline]**

**7.168.2.8 int32 OsclNativeFile::Open (const char \*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\_FileServer &fileserv)**

**7.168.2.9 int32 OsclNativeFile::Open (const oscl\_wchar \*filename, uint32 mode, const OsclNativeFileParams &params, Oscl\_FileServer &fileserv)**

**7.168.2.10 int32 OsclNativeFile::Open (const OsclFileHandle &, uint32 mode, const OsclNativeFileParams &params, Oscl\_FileServer &fileserv)**

**7.168.2.11 uint32 OsclNativeFile::Read (OsclAny \*buffer, uint32 size, uint32 numelements)**

**7.168.2.12 int32 OsclNativeFile::ReadAsync (OsclAny \*buffer, uint32 size, uint32 numelements, OsclAOStatus &status)**

Asynchronous read.

#### Parameters

*buffer*,: data buffer, must be at least size\*numelements bytes

*size*,: size of elements

*numelements*,: number of elements to read  
*status*,: Request status for asynchronous completion

#### Returns

: 0 for success.

#### 7.168.2.13 void OsclNativeFile::ReadAsyncCancel ()

Cancel any pending async read.

#### 7.168.2.14 int32 OsclNativeFile::Seek (TOsclFileOffset *offset*, Oscl\_File::seek\_type *origin*)

#### 7.168.2.15 int32 OsclNativeFile::SetSize (uint32 *size*)

#### 7.168.2.16 TOsclFileOffset OsclNativeFile::Size ()

#### 7.168.2.17 TOsclFileOffset OsclNativeFile::Tell ()

#### 7.168.2.18 uint32 OsclNativeFile::Write (const OsclAny \* *buffer*, uint32 *size*, uint32 *numelements*)

The documentation for this class was generated from the following file:

- [oscl\\_file\\_native.h](#)

## 7.169 OsclNativeFileParams Class Reference

```
#include <oscl_file_types.h>
```

### Public Member Functions

- [OsclNativeFileParams](#) (uint32 mode=0, uint32 bufsize=0, uint32 asyncsize=0)

### Data Fields

- uint32 [iNativeAccessMode](#)
- uint32 [iNativeBufferSize](#)
- uint32 [iAsyncReadBufferSize](#)

#### 7.169.1 Constructor & Destructor Documentation

7.169.1.1 [OsclNativeFileParams::OsclNativeFileParams \(uint32 mode = 0, uint32 bufsize = 0, uint32 asyncsize = 0\) \[inline\]](#)

#### 7.169.2 Field Documentation

7.169.2.1 [uint32 OsclNativeFileParams::iAsyncReadBufferSize](#)

7.169.2.2 [uint32 OsclNativeFileParams::iNativeAccessMode](#)

7.169.2.3 [uint32 OsclNativeFileParams::iNativeBufferSize](#)

The documentation for this class was generated from the following file:

- [oscl\\_file\\_types.h](#)

## 7.170 OsclNetworkAddress Class Reference

```
#include <oscl_socket_types.h>
```

### Public Member Functions

- [OsclNetworkAddress \(\)](#)
- [OsclNetworkAddress \(const char \\*addr, int p\)](#)
- [bool operator== \(const OsclNetworkAddress &rhs\) const](#)

### Data Fields

- [OsclNameString< PVNETWORKADDRESS\\_LEN > ipAddr](#)
- [int port](#)

#### 7.170.1 Constructor & Destructor Documentation

**7.170.1.1 OsclNetworkAddress::OsclNetworkAddress () [inline]**

**7.170.1.2 OsclNetworkAddress::OsclNetworkAddress (const char \*addr, int p) [inline]**

#### 7.170.2 Member Function Documentation

**7.170.2.1 bool OsclNetworkAddress::operator== (const OsclNetworkAddress &rhs) const [inline]**

References ipAddr, oscl\_strcmp(), port, and OsclNameString< \_\_len >::Str().

#### 7.170.3 Field Documentation

**7.170.3.1 OsclNameString<PVNETWORKADDRESS\_LEN> OsclNetworkAddress::ipAddr**

Referenced by OsclUDPSocketI::BindAsync(), OsclTCPSocketI::BindAsync(), GetHostByNameParam::canPersistMoreHostAddresses(), operator==(), and GetHostByNameParam::PersistHostAddress().

**7.170.3.2 int OsclNetworkAddress::port**

Referenced by OsclUDPSocketI::BindAsync(), OsclTCPSocketI::BindAsync(), and operator==().

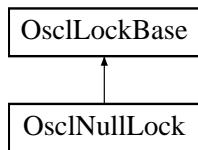
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_types.h](#)

## 7.171 OsclNullLock Class Reference

```
#include <oscl_lock_base.h>
```

Inheritance diagram for OsclNullLock:



### Public Member Functions

- virtual void [Lock \(\)](#)
- virtual void [Unlock \(\)](#)
- virtual [~OsclNullLock \(\)](#)

#### 7.171.1 Constructor & Destructor Documentation

7.171.1.1 virtual OsclNullLock::~OsclNullLock () [inline, virtual]

#### 7.171.2 Member Function Documentation

7.171.2.1 virtual void OsclNullLock::Lock () [inline, virtual]

Implements [OsclLockBase](#).

7.171.2.2 virtual void OsclNullLock::Unlock () [inline, virtual]

Implements [OsclLockBase](#).

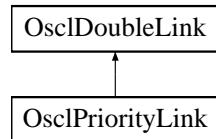
The documentation for this class was generated from the following file:

- [oscl\\_lock\\_base.h](#)

## 7.172 OsclPriorityLink Class Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclPriorityLink:



### Data Fields

- int32 [iPriority](#)

#### 7.172.1 Field Documentation

##### 7.172.1.1 int32 OsclPriorityLink::iPriority

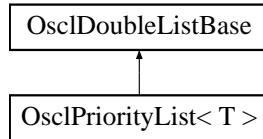
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.173 OsclPriorityList< T > Class Template Reference

```
#include <oscl_double_list.h>
```

Inheritance diagram for OsclPriorityList< T >:



### Public Member Functions

- OSCL\_INLINE OsclPriorityList()
- OSCL\_INLINE OsclPriorityList(int32 anOffset)
- OSCL\_INLINE void Insert(T &aRef)
- OSCL\_INLINE bool IsHead(const T \*aPtr) const
- OSCL\_INLINE bool IsTail(const T \*aPtr) const
- OSCL\_INLINE T \* Head() const
- OSCL\_INLINE T \* Tail() const

**template<class T> class OsclPriorityList< T >**

#### 7.173.1 Constructor & Destructor Documentation

**7.173.1.1 template<class T > OSCL\_INLINE OsclPriorityList< T >::OsclPriorityList()**

**7.173.1.2 template<class T > OSCL\_INLINE OsclPriorityList< T >::OsclPriorityList(int32  
anOffset)**

#### 7.173.2 Member Function Documentation

**7.173.2.1 template<class T > OSCL\_INLINE T\* OsclPriorityList< T >::Head() const**

**7.173.2.2 template<class T > OSCL\_INLINE void OsclPriorityList< T >::Insert(T &aRef)**

**7.173.2.3 template<class T > OSCL\_INLINE bool OsclPriorityList< T >::IsHead(const T \*  
aPtr) const**

**7.173.2.4 template<class T > OSCL\_INLINE bool OsclPriorityList< T >::IsTail(const T \*aPtr)  
const**

**7.173.2.5 template<class T > OSCL\_INLINE T\* OsclPriorityList< T >::Tail() const**

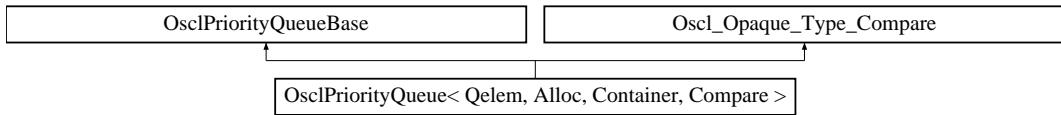
The documentation for this class was generated from the following file:

- [oscl\\_double\\_list.h](#)

## 7.174 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 Member Functions

- `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 Member Functions

- `void push_heap (iterator first, iterator last)`
- `void pop_heap (iterator first, iterator last)`
- `iterator find_heap (const value_type &input, iterator first, iterator last)`
- `int validate ()`
- `void swap (OsclAny *dest, const OsclAny *src)`
- `int compare_LT (OsclAny *a, OsclAny *b) const`
- `int compare_EQ (const OsclAny *a, const OsclAny *b) const`

### Protected Attributes

- `Container c`
- `Compare comp`

## Friends

- class [oscl\\_priqueue\\_test](#)

```
template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> class OsclPriorityQueue< Qelem, Alloc, Container, Compare >
```

### 7.174.1 Member Typedef Documentation

- 7.174.1.1 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> typedef Container::const_reference OsclPriorityQueue< Qelem, Alloc, Container, Compare >::const_reference`
- 7.174.1.2 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> typedef Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::container_type`
- 7.174.1.3 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> typedef Container::iterator OsclPriorityQueue< Qelem, Alloc, Container, Compare >::iterator`
- 7.174.1.4 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> typedef Container::value_type OsclPriorityQueue< Qelem, Alloc, Container, Compare >::value_type`

### 7.174.2 Constructor & Destructor Documentation

- 7.174.2.1 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> OsclPriorityQueue< Qelem, Alloc, Container, Compare >::OsclPriorityQueue () [inline]`
- 7.174.2.2 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> virtual OsclPriorityQueue< Qelem, Alloc, Container, Compare >::~OsclPriorityQueue () [inline, virtual]`

### 7.174.3 Member Function Documentation

- 7.174.3.1 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::compare_EQ (const OsclAny * a, const OsclAny * b) const [inline, protected, virtual]`

Return a==b.

Implements [Oscl\\_Opaque\\_Type\\_Compare](#).

- 7.174.3.2 `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::compare_LT (OsclAny * a, OsclAny * b) const [inline, protected, virtual]`

Return a<b.

Implements [Oscl\\_Opaque\\_Type\\_Compare](#).

**7.174.3.3** `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> bool OsclPriorityQueue< Qelem, Alloc, Container, Compare >::empty () const [inline]`

**7.174.3.4** `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]`

Referenced by `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::find_heap()`.

**7.174.3.5** `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::pop () [inline]`

**7.174.3.6** `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]`

Referenced by `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::pop()`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::pop_heap()`.

**7.174.3.7** `template<class Qelem, class Alloc, class Container = Oscl_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::push (const value_type & input) [inline]`

**7.174.3.8** `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]`

Referenced by `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::push()`, and `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::push_heap()`.

**7.174.3.9** `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]`

Referenced by `OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::remove()`.

**7.174.3.10** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::reserve (uint32 *n*) [inline]

**7.174.3.11** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> uint32 OsclPriorityQueue< Qelem, Alloc, Container, Compare >::size () const [inline]

**7.174.3.12** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> void OsclPriorityQueue< Qelem, Alloc, Container, Compare >::swap (OsclAny \* *a*, const OsclAny \* *b*) [inline, protected, virtual]

Swap element at "a" with element at "b". Both pointers must be non-NUL.

Implements [Oscl\\_Opaque\\_Type\\_Compare](#).

**7.174.3.13** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> const\_reference OsclPriorityQueue< Qelem, Alloc, Container, Compare >::top () const [inline]

**7.174.3.14** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> int OsclPriorityQueue< Qelem, Alloc, Container, Compare >::validate () [inline, protected]

**7.174.3.15** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> const Container& OsclPriorityQueue< Qelem, Alloc, Container, Compare >::vec () [inline]

## 7.174.4 Friends And Related Function Documentation

**7.174.4.1** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> friend class oscl\_priqueue\_test [friend]

## 7.174.5 Field Documentation

**7.174.5.1** template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Container OsclPriorityQueue< Qelem, Alloc, Container, Compare >::c [protected]

Referenced by OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::empty(), OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::OsclPriorityQueue(), OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::pop(), OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::push(), OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::reserve(), OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::size(), OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::top(), OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::validate(), and OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::vec().

**7.174.5.2 template<class Qelem, class Alloc, class Container = Oscl\_Vector<Qelem, Alloc>, class Compare = OsclCompareLess<Qelem>> Compare OsclPriorityQueue< Qelem, Alloc, Container, Compare >::comp [protected]**

Referenced by OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::compare\_LT(), and OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::validate().

The documentation for this class was generated from the following file:

- [oscl\\_priqueue.h](#)

## 7.175 OsclPriorityQueueBase Class Reference

```
#include <oscl_priqueue.h>
```

Inheritance diagram for OsclPriorityQueueBase:



### Protected Member Functions

- virtual ~OsclPriorityQueueBase ()
- OSCL\_IMPORT\_REF void push\_heap (OsclAny \*first, OsclAny \*last)
- OSCL\_IMPORT\_REF void pop\_heap (OsclAny \*first, OsclAny \*last)
- OSCL\_IMPORT\_REF OsclAny \* find\_heap (const OsclAny \*input, OsclAny \*first, OsclAny \*last)
- OSCL\_IMPORT\_REF int remove (const OsclAny \*input)
- void construct (Oscl\_Opaque\_Type\_Compare \*ot, Oscl\_Vector\_Base \*vec)

#### 7.175.1 Detailed Description

[OsclPriorityQueueBase](#) is a non-templatized base class for [OsclPriorityQueue](#). The purpose of this base class is to avoid large inline routines in the [OsclPriorityQueue](#) implementation. This class is not intended for direct instantiation except by [OsclPriorityQueue](#).

#### 7.175.2 Constructor & Destructor Documentation

**7.175.2.1 virtual OsclPriorityQueueBase::~OsclPriorityQueueBase () [inline, protected, virtual]**

#### 7.175.3 Member Function Documentation

**7.175.3.1 void OsclPriorityQueueBase::construct (Oscl\_Opaque\_Type\_Compare \* ot, Oscl\_Vector\_Base \* vec) [inline, protected]**

Referenced by [OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\\_Vector< TOsclReady, OsclReadyAlloc >, OsclTimerCompare >::OsclPriorityQueue\(\)](#).

**7.175.3.2 OSCL\_IMPORT\_REF OsclAny\* OsclPriorityQueueBase::find\_heap (const OsclAny \* input, OsclAny \* first, OsclAny \* last) [protected]**

**7.175.3.3 OSCL\_IMPORT\_REF void OsclPriorityQueueBase::pop\_heap (OsclAny \* first, OsclAny \* last) [protected]**

**7.175.3.4 OSCL\_IMPORT\_REF void OsclPriorityQueueBase::push\_heap (OsclAny \* first, OsclAny \* last) [protected]**

**7.175.3.5 OSCL\_IMPORT\_REF int OsclPriorityQueueBase::remove (const OsclAny \* input) [protected]**

The documentation for this class was generated from the following file:

- [oscl\\_priqueue.h](#)

## 7.176 OsclProcStatus Class Reference

```
#include <oscl_procstatus.h>
```

### Public Types

- enum eOsclProcError {
 SUCCESS\_ERROR = 0, OTHER\_ERROR, TOO\_MANY\_THREADS\_ERROR, BAD\_THREADID\_ADDR\_ERROR,
 MAX\_THRDS\_REACHED\_ERROR, INVALID\_THREAD\_ID\_ERROR, NOT\_ENOUGH\_MEMORY\_ERROR, OUTOFMEMORY\_ERROR,
 NOT\_ENOUGH\_RESOURCES\_ERROR, THREAD\_1\_INACTIVE\_ERROR, ALREADY\_SUSPENDED\_ERROR, NOT\_SUSPENDED\_ERROR,
 INVALID\_THREAD\_ERROR, INVALID\_PARAM\_ERROR, NO\_PERMISSION\_ERROR, INVALID\_PRIORITY\_ERROR,
 PSHARED\_NOT\_ZERO\_ERROR, EXCEED\_MAX\_COUNT\_VARIABLE\_ERROR, THREAD\_BLOCK\_ERROR, EXCEED\_MAX\_SEM\_COUNT\_ERROR,
 INVALID\_HANDLE\_ERROR, INVALID\_OPERATION\_ERROR, INVALID\_FUNCTION\_ERROR, INVALID\_ACCESS\_ERROR,
 INVALID\_ARGUMENT\_ERROR, SYSTEM\_RESOURCES\_UNAVAILABLE\_ERROR, INVALID\_POINTER\_ERROR, RELOCK\_MUTEX\_ERROR,
 THREAD\_NOT\_OWN\_MUTEX\_ERROR, MUTEX\_LOCKED\_ERROR, WAIT\_ABANDONED\_ERROR, WAIT\_TIMEOUT\_ERROR,
 SEM\_NOT\_SIGNALED\_ERROR, PSHARED\_ATTRIBUTE\_SETTING\_ERROR, NOT\_IMPLEMENTED }
 }

### 7.176.1 Detailed Description

Class [OsclProcStatus](#)

### 7.176.2 Member Enumeration Documentation

#### 7.176.2.1 enum OsclProcStatus::eOsclProcError

List of enums which contain error codes

**Enumerator:**

*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\_SIGNALLED\_ERROR  
PSHARED\_ATTRIBUTE\_SETTING\_ERROR  
NOT\_IMPLEMENTED*

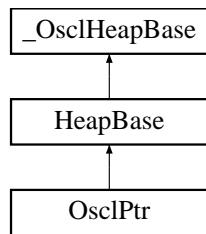
The documentation for this class was generated from the following file:

- [oscl\\_procstatus.h](#)

## 7.177 OsclPtr Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclPtr:



### Public Member Functions

- [OsclPtr \(uint8 \\*ptr, int32 &len, int32 max\)](#)
- [OsclPtr \(const OsclPtr &d\)](#)
- [uint8 \\* Ptr \(\)](#)
- [void SetLength \(int32 l\)](#)
- [int32 Length \(\)](#)
- [void Zero \(\)](#)
- [void Set \(OsclPtr &v\)](#)
- [void Set \(uint8 \\*ptr, int32 len, int32 max\)](#)
- [void Append \(OsclPtrC &v\)](#)

#### 7.177.1 Constructor & Destructor Documentation

**7.177.1.1 OsclPtr::OsclPtr (uint8 \*ptr, int32 &len, int32 max) [inline]**

**7.177.1.2 OsclPtr::OsclPtr (const OsclPtr &d) [inline]**

#### 7.177.2 Member Function Documentation

**7.177.2.1 void OsclPtr::Append (OsclPtrC &v) [inline]**

References OsclPtrC::Length(), OSCL\_ASSERT, oscl\_memmove(), and OsclPtrC::Ptr().

**7.177.2.2 int32 OsclPtr::Length () [inline]**

**7.177.2.3 uint8\* OsclPtr::Ptr () [inline]**

**7.177.2.4 void OsclPtr::Set (uint8 \*ptr, int32 len, int32 max) [inline]**

**7.177.2.5 void OsclPtr::Set (OsclPtr &v) [inline]**

**7.177.2.6 void OsclPtr::SetLength (int32 l) [inline]**

References OSCL\_ASSERT.

**7.177.2.7 void OsclPtr::Zero () [inline]**

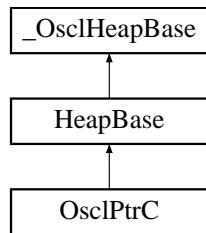
The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 7.178 OsclPtrC Class Reference

```
#include <oscl_file_async_read.h>
```

Inheritance diagram for OsclPtrC:



### Public Member Functions

- [OsclPtrC](#) (const uint8 \*ptr, int32 len, int32 max)
- [OsclPtrC](#) (const [OsclPtrC](#) &d)
- const uint8 \* [Ptr](#) ()
- void [SetLength](#) (int32 l)
- int32 [Length](#) ()
- void [Zero](#) ()
- void [Set](#) ([OsclPtrC](#) \*v)
- void [Set](#) (uint8 \*ptr, int32 len, int32 max)
- [OsclPtrC Right](#) (int32 size)
- [OsclPtrC Left](#) (int32 size)

#### 7.178.1 Constructor & Destructor Documentation

**7.178.1.1 [OsclPtrC::OsclPtrC](#) (const uint8 \*ptr, int32 len, int32 max) [inline]**

**7.178.1.2 [OsclPtrC::OsclPtrC](#) (const OsclPtrC & d) [inline]**

#### 7.178.2 Member Function Documentation

**7.178.2.1 [OsclPtrC OsclPtrC::Left](#) (int32 size) [inline]**

References OSCL\_ASSERT.

**7.178.2.2 [int32 OsclPtrC::Length](#) () [inline]**

Referenced by [OsclPtr::Append\(\)](#).

**7.178.2.3 [const uint8\\* OsclPtrC::Ptr](#) () [inline]**

Referenced by [OsclPtr::Append\(\)](#).

**7.178.2.4 OsclPtrC OsclPtrC::Right (int32 *size*) [inline]**

References OSCL\_ASSERT.

**7.178.2.5 void OsclPtrC::Set (uint8 \* *ptr*, int32 *len*, int32 *max*) [inline]****7.178.2.6 void OsclPtrC::Set (OsclPtrC \* *v*) [inline]****7.178.2.7 void OsclPtrC::SetLength (int32 *l*) [inline]**

References OSCL\_ASSERT.

**7.178.2.8 void OsclPtrC::Zero () [inline]**

The documentation for this class was generated from the following file:

- [oscl\\_file\\_async\\_read.h](#)

## 7.179 OsclRand Class Reference

```
#include <oscl_rand.h>
```

### Public Member Functions

- OSCL\_COND\_IMPORT\_REF void [Seed](#) (int32 seed)
- OSCL\_COND\_IMPORT\_REF int32 [Rand](#) ()

#### 7.179.1 Member Function Documentation

##### 7.179.1.1 OSCL\_COND\_IMPORT\_REF int32 OsclRand::Rand ()

##### 7.179.1.2 OSCL\_COND\_IMPORT\_REF void OsclRand::Seed (int32 *seed*)

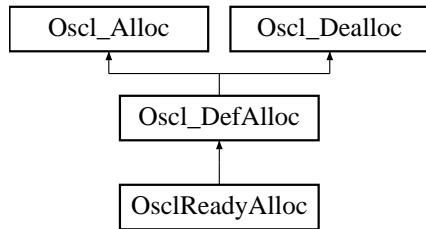
The documentation for this class was generated from the following file:

- [oscl\\_rand.h](#)

## 7.180 OsclReadyAlloc Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyAlloc:



### Public Member Functions

- [OsclAny \\* allocate \(const uint32 size\)](#)
- [OsclAny \\* allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\)](#)
- [void deallocate \(OsclAny \\*p\)](#)

#### 7.180.1 Member Function Documentation

##### 7.180.1.1 [OsclAny\\* OsclReadyAlloc::allocate \(const uint32 size\) \[virtual\]](#)

Implements [Oscl\\_DefAlloc](#).

##### 7.180.1.2 [OsclAny\\* OsclReadyAlloc::allocate\\_fl \(const uint32 size, const char \\*file\\_name, const int line\\_num\) \[virtual\]](#)

Reimplemented from [Oscl\\_DefAlloc](#).

##### 7.180.1.3 [void OsclReadyAlloc::deallocate \(OsclAny \\*p\) \[virtual\]](#)

Implements [Oscl\\_DefAlloc](#).

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 7.181 OsclReadyCompare Class Reference

```
#include <oscl_scheduler_readyq.h>
```

### Static Public Member Functions

- static int `compare (TOsclReady &a, TOsclReady &b)`

#### 7.181.1 Member Function Documentation

##### 7.181.1.1 static int OsclReadyCompare::compare (TOsclReady & *a*, TOsclReady & *b*) [static]

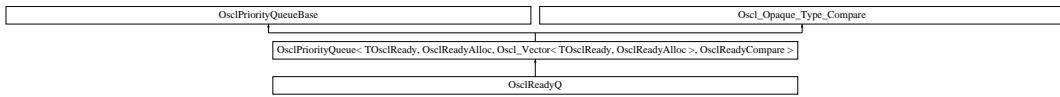
The documentation for this class was generated from the following file:

- `oscl_scheduler_readyq.h`

## 7.182 OsclReadyQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclReadyQ:



### Public Member Functions

- void [Construct](#) (int)
- void [ThreadLogon](#) ()
- void [ThreadLogoff](#) ()
- void [Remove](#) (TOsclReady)
- bool [IsIn](#) (TOsclReady)
- uint32 [Depth](#) ()
- TOsclReady [PopTop](#) ()
- TOsclReady [Top](#) ()
- TOsclReady [WaitAndPopTop](#) ()
- TOsclReady [WaitAndPopTop](#) (uint32)
- int32 [PendComplete](#) (PVActiveBase \*pvbase, int32 aReason)
- int32 [WaitForRequestComplete](#) (PVActiveBase \*)
- void [RegisterForCallback](#) (OsclSchedulerObserver \*aCallback, OsclAny \*aCallbackContext)
- void [TimerCallback](#) (uint32 aDelayMicrosec)
- OsclSchedulerObserver \* [Callback](#) ()

#### 7.182.1 Member Function Documentation

**7.182.1.1 OsclSchedulerObserver\* OsclReadyQ::Callback () [inline]**

**7.182.1.2 void OsclReadyQ::Construct (int)**

**7.182.1.3 uint32 OsclReadyQ::Depth () [inline]**

References OsclPriorityQueue< TOsclReady, OsclReadyAlloc, Oscl\_Vector< TOsclReady, OsclReadyAlloc >, OsclReadyCompare >::size().

- 7.182.1.4 `bool OsclReadyQ::IsIn (TOsclReady)`
- 7.182.1.5 `int32 OsclReadyQ::PendComplete (PVActiveBase * pibase, int32 aReason)`
- 7.182.1.6 `TOsclReady OsclReadyQ::PopTop ()`
- 7.182.1.7 `void OsclReadyQ::RegisterForCallback (OsclSchedulerObserver * aCallback, OsclAny * aCallbackContext)`
- 7.182.1.8 `void OsclReadyQ::Remove (TOsclReady)`
- 7.182.1.9 `void OsclReadyQ::ThreadLogoff ()`
- 7.182.1.10 `void OsclReadyQ::ThreadLogon ()`
- 7.182.1.11 `void OsclReadyQ::TimerCallback (uint32 aDelayMicrosec)`
- 7.182.1.12 `TOsclReady OsclReadyQ::Top ()`
- 7.182.1.13 `TOsclReady OsclReadyQ::WaitAndPopTop (uint32)`
- 7.182.1.14 `TOsclReady OsclReadyQ::WaitAndPopTop ()`
- 7.182.1.15 `int32 OsclReadyQ::WaitForRequestComplete (PVActiveBase *)`

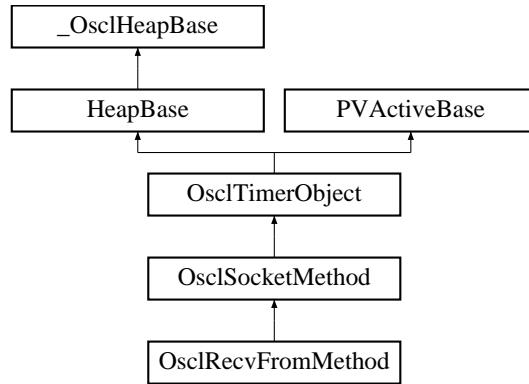
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 7.183 OsclRecvFromMethod Class Reference

```
#include <oscl_socket_recv_from.h>
```

Inheritance diagram for OsclRecvFromMethod:



### Public Member Functions

- [`~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 Member Functions

- static [`OsclRecvFromMethod \* NewL \(OsclIPSocketI &c\)`](#)

#### 7.183.1 Constructor & Destructor Documentation

##### 7.183.1.1 OsclRecvFromMethod::~OsclRecvFromMethod ()

#### 7.183.2 Member Function Documentation

##### 7.183.2.1 uint8\* OsclRecvFromMethod::GetRecvData (int32 \* aLength)

Referenced by OsclUDPSocketI::GetRecvData().

**7.183.2.2 static OsclRecvFromMethod\* OsclRecvFromMethod::NewL (OsclIPSocketI & c)  
[static]**

**7.183.2.3 TPVSocketEvent OsclRecvFromMethod::RecvFrom (uint8 \*& aPtr, uint32 aMaxLen,  
OsclNetworkAddress & aAddress, int32 aTimeout, uint32 aMultiMax, Oscl\_Vector<  
uint32, OsclMemAllocator > \* aPacketLen, Oscl\_Vector< OsclNetworkAddress,  
OsclMemAllocator > \* aPacketSource)**

Referenced by OsclUDPSocketI::RecvFrom().

**7.183.2.4 OsclRecvFromRequest\* OsclRecvFromMethod::RecvFromRequest () [inline]**

References OsclSocketMethod::iSocketRequestAO.

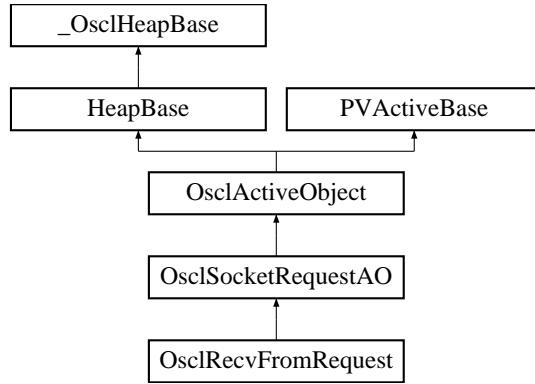
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_recv\\_from.h](#)

## 7.184 OsclRecvFromRequest Class Reference

```
#include <oscl_socket_recv_from.h>
```

Inheritance diagram for OsclRecvFromRequest:



### Public Member Functions

- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvFromRequest (OsclSocketMethod &c)`
- `void RecvFrom (uint8 *&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource)`
- `void Success ()`

#### 7.184.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.184.2 Constructor & Destructor Documentation

**7.184.2.1 OsclRecvFromRequest::OsclRecvFromRequest (OsclSocketMethod & c) [inline]**

#### 7.184.3 Member Function Documentation

**7.184.3.1 uint8\* OsclRecvFromRequest::GetRecvData (int32 \* aLength)**

**7.184.3.2 void OsclRecvFromRequest::RecvFrom (uint8 \*& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, uint32 aMultiMax, Oscl\_Vector< uint32, OsclMemAllocator > \* aPacketLen, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \* aPacketSource)**

**7.184.3.3 void OsclRecvFromRequest::Success () [virtual]**

Reimplemented from [OsclSocketRequestAO](#).

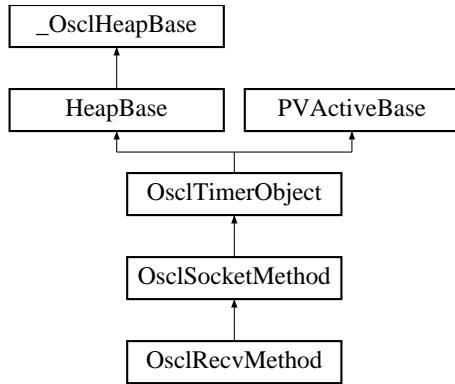
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_recv\\_from.h](#)

## 7.185 OsclRecvMethod Class Reference

```
#include <oscl_socket_recv.h>
```

Inheritance diagram for OsclRecvMethod:



### Public Member Functions

- `~OsclRecvMethod ()`
- `TPVSocketEvent Recv (uint8 *&aPtr, uint32 aMaxLen, int32 aTimeout)`
- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvRequest * RecvRequest ()`

### Static Public Member Functions

- static `OsclRecvMethod * NewL (OsclIPSocketI &c)`

#### 7.185.1 Constructor & Destructor Documentation

##### 7.185.1.1 OsclRecvMethod::~OsclRecvMethod ()

#### 7.185.2 Member Function Documentation

##### 7.185.2.1 uint8\* OsclRecvMethod::GetRecvData (int32 \* aLength)

Referenced by `OsclTCPSocketI::GetRecvData()`.

##### 7.185.2.2 static OsclRecvMethod\* OsclRecvMethod::NewL (OsclIPSocketI & c) [static]

##### 7.185.2.3 TPVSocketEvent OsclRecvMethod::Recv (uint8 \*& aPtr, uint32 aMaxLen, int32 aTimeout)

Referenced by `OsclTCPSocketI::Recv()`.

**7.185.2.4 OsclRecvRequest\* OsclRecvMethod::RecvRequest () [inline]**

References OsclSocketMethod::iSocketRequestAO.

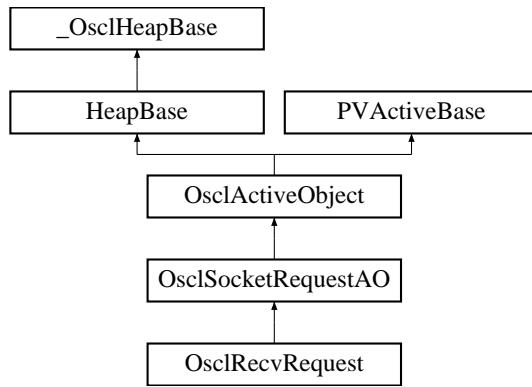
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_recv.h](#)

## 7.186 OsclRecvRequest Class Reference

```
#include <oscl_socket_recv.h>
```

Inheritance diagram for OsclRecvRequest:



### Public Member Functions

- `uint8 * GetRecvData (int32 *aLength)`
- `OsclRecvRequest (OsclSocketMethod &c)`
- `void Recv (uint8 *&aPtr, uint32 aMaxLen)`
- `void Success ()`

#### 7.186.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.186.2 Constructor & Destructor Documentation

**7.186.2.1 OsclRecvRequest::OsclRecvRequest (OsclSocketMethod & c) [inline]**

#### 7.186.3 Member Function Documentation

**7.186.3.1 `uint8* OsclRecvRequest::GetRecvData (int32 * aLength)`**

**7.186.3.2 `void OsclRecvRequest::Recv (uint8 *& aPtr, uint32 aMaxLen)`**

**7.186.3.3 `void OsclRecvRequest::Success () [virtual]`**

Reimplemented from [OsclSocketRequestAO](#).

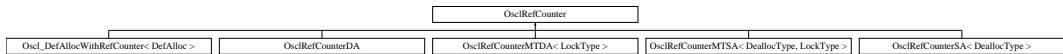
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_recv.h](#)

## 7.187 OsclRefCounter Class Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounter:



### Public Member Functions

- virtual void `addRef ()=0`
- virtual void `removeRef ()=0`
- virtual uint32 `getCount ()=0`
- virtual `~OsclRefCounter ()`

#### 7.187.1 Detailed Description

Interface class for [OsclRefCounter](#) implementations

#### 7.187.2 Constructor & Destructor Documentation

**7.187.2.1 virtual OsclRefCounter::~OsclRefCounter () [inline, virtual]**

#### 7.187.3 Member Function Documentation

**7.187.3.1 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 >](#).

Referenced by [OsclSharedPtr< TheClass >::Bind\(\)](#), [OsclRefCounterMemFrag::OsclRefCounterMemFrag\(\)](#), and [OsclSharedPtr< TheClass >::OsclSharedPtr\(\)](#).

**7.187.3.2 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 >](#).

Referenced by [OsclSharedPtr< TheClass >::get\\_count\(\)](#).

**7.187.3.3 virtual void OsclRefCounter::removeRef () [pure virtual]**

Delete from reference count

Implemented in [OsclRefCounterDA](#), [OsclRefCounterSA< DeallocType >](#), [OsclRefCounterMTDA< LockType >](#), [OsclRefCounterMTSA< DeallocType, LockType >](#), and [Oscl\\_DefAllocWithRefCounter< DefAlloc >](#).

Referenced by `OsclSharedPtr< TheClass >::Bind()`, and `OsclSharedPtr< TheClass >::~OsclSharedPtr()`.

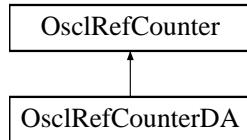
The documentation for this class was generated from the following file:

- [oscl\\_refcounter.h](#)

## 7.188 OsclRefCounterDA Class Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterDA:



### Public Member Functions

- [OsclRefCounterDA \(OsclAny \\*p, OsclDestructDealloc \\*dealloc\)](#)
- virtual [~OsclRefCounterDA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 7.188.1 Detailed Description

Implementation of an [OsclRefCounter](#) that uses a dynamically created deallocator.

#### 7.188.2 Constructor & Destructor Documentation

##### 7.188.2.1 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

References NULL, and OSCL\_ASSERT.

##### 7.188.2.2 virtual OsclRefCounterDA::~OsclRefCounterDA () [inline, virtual]

Destructor empty

### 7.188.3 Member Function Documentation

#### 7.188.3.1 void OsclRefCounterDA::addRef () [inline, virtual]

Add to the reference count

Implements [OsclRefCounter](#).

#### 7.188.3.2 uint32 OsclRefCounterDA::getCount () [inline, virtual]

Gets the current number of references

Implements [OsclRefCounter](#).

#### 7.188.3.3 void OsclRefCounterDA::removeRef () [inline, virtual]

Remove from the reference count

Implements [OsclRefCounter](#).

References OsclDestructDealloc::destruct\_and\_dealloc().

The documentation for this class was generated from the following file:

- [oscl\\_refcounter.h](#)

## 7.189 OsclRefCounterMemFrag Class Reference

```
#include <oscl_refcounter_memfrag.h>
```

### Public Member Functions

- `OsclRefCounterMemFrag (OsclMemoryFragment &m, OsclRefCounter *r, uint32 in_capacity)`
- `OsclRefCounterMemFrag (const OsclRefCounterMemFrag &x)`
- `OsclRefCounterMemFrag ()`
- `OsclRefCounterMemFrag & operator= (const OsclRefCounterMemFrag &x)`
- `~OsclRefCounterMemFrag ()`
- `OsclRefCounter * getRefCounter ()`
- `OsclMemoryFragment & getMemFrag ()`
- `OsclAny * getMemFragPtr ()`
- `uint32 getMemFragSize ()`
- `uint32 getCapacity ()`
- `uint32 getCount ()`

### 7.189.1 Detailed Description

Class to contain a memory fragment with it's associated reference counter.

### 7.189.2 Constructor & Destructor Documentation

#### 7.189.2.1 OsclRefCounterMemFrag::OsclRefCounterMemFrag (OsclMemoryFragment & m, OsclRefCounter \* r, uint32 in\_capacity) [inline]

Constructor. A valid memory fragment and reference counter are required as input. The memory fragment structure will be copied locally.

##### Parameters

- `m` reference to memory fragment
- `r` pointer to the reference counter associated with the memory fragment.

#### 7.189.2.2 OsclRefCounterMemFrag::OsclRefCounterMemFrag (const OsclRefCounterMemFrag & x) [inline]

Copy constructor.

#### 7.189.2.3 OsclRefCounterMemFrag::OsclRefCounterMemFrag () [inline]

Default constructor.

References `OsclRefCounter::addRef()`.

**7.189.2.4 OsclRefCounterMemFrag::~OsclRefCounterMemFrag () [inline]**

Destructor. Removes this object's reference from the reference counter. The reference counter will not be deleted. The reference counter is designed to self-delete when it's reference count reaches 0.

**7.189.3 Member Function Documentation****7.189.3.1 uint32 OsclRefCounterMemFrag::getCapacity () [inline]**

Returns the capacity of the memory fragment

**Returns****7.189.3.2 uint32 OsclRefCounterMemFrag::getCount () [inline]**

Returns the reference counter's current count.

**7.189.3.3 OsclMemoryFragment& OsclRefCounterMemFrag::getMemFrag () [inline]**

Returns a reference to the contained memory fragment structure.

**7.189.3.4 OsclAny\* OsclRefCounterMemFrag::getMemFragPtr () [inline]**

Returns a pointer to the memory fragment data.

**7.189.3.5 uint32 OsclRefCounterMemFrag::getMemFragSize () [inline]**

Returns the size of the memory fragment data which equals its filled size.

**Returns****7.189.3.6 OsclRefCounter\* OsclRefCounterMemFrag::getRefCounter () [inline]**

Returns a pointer to the contained reference counter object

**7.189.3.7 OsclRefCounterMemFrag& OsclRefCounterMemFrag::operator= (const OsclRefCounterMemFrag & x) [inline]**

Assignment Operator

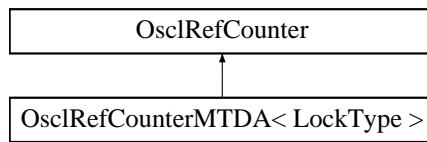
The documentation for this class was generated from the following file:

- [oscl\\_refcounter\\_memfrag.h](#)

## 7.190 OsclRefCounterMTDA< LockType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterMTDA< LockType >:



### Public Member Functions

- [OsclRefCounterMTDA \(OsclAny \\*p, OsclDestructDealloc \\*dealloc\)](#)
- virtual [~OsclRefCounterMTDA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 7.190.1 Detailed Description

**template<class LockType> class OsclRefCounterMTDA< LockType >**

Implementation of [OsclRefCounterDA](#) for multi-threaded use. A templated lock class must be specified.

#### 7.190.2 Constructor & Destructor Documentation

**7.190.2.1 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

References NULL, and OSCL\_ASSERT.

**7.190.2.2 template<class LockType> virtual OsclRefCounterMTDA< LockType >::~OsclRefCounterMTDA () [inline, virtual]**

Destructor empty

### 7.190.3 Member Function Documentation

**7.190.3.1 template<class LockType> void OsclRefCounterMTDA< LockType >::addRef () [inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**7.190.3.2 template<class LockType> uint32 OsclRefCounterMTDA< LockType >::getCount () [inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**7.190.3.3 template<class LockType> void OsclRefCounterMTDA< LockType >::removeRef () [inline, virtual]**

Remove from the reference count

Implements [OsclRefCounter](#).

References [OsclDestructDealloc::destruct\\_and\\_dealloc\(\)](#).

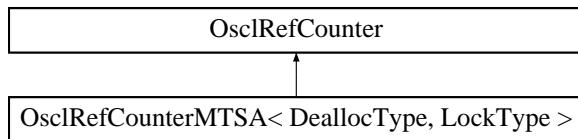
The documentation for this class was generated from the following file:

- [oscl\\_refcounter.h](#)

## 7.191 OsclRefCounterMTSA< DeallocType, LockType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterMTSA< DeallocType, LockType >:



### Public Member Functions

- [OsclRefCounterMTSA \(OsclAny \\*p\)](#)
- virtual [~OsclRefCounterMTSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 7.191.1 Detailed Description

`template<class DeallocType, class LockType> class OsclRefCounterMTSA< DeallocType, LockType >`

Implementation of [OsclRefCounterSA](#) for multi-threaded use. A templated lock class must be specified.

#### 7.191.2 Constructor & Destructor Documentation

##### 7.191.2.1 `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

References NULL, and OSCL\_ASSERT.

##### 7.191.2.2 `template<class DeallocType, class LockType> virtual OsclRefCounterMTSA< DeallocType, LockType >::~OsclRefCounterMTSA () [inline, virtual]`

Destructor empty

### 7.191.3 Member Function Documentation

**7.191.3.1 template<class DeallocType, class LockType> void OsclRefCounterMTSA<DeallocType, LockType >::addRef () [inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**7.191.3.2 template<class DeallocType, class LockType> uint32 OsclRefCounterMTSA<DeallocType, LockType >::getCount () [inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**7.191.3.3 template<class DeallocType, class LockType> void OsclRefCounterMTSA<DeallocType, LockType >::removeRef () [inline, virtual]**

Remove from the reference count

Implements [OsclRefCounter](#).

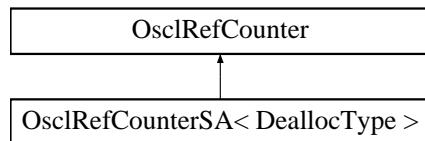
The documentation for this class was generated from the following file:

- [oscl\\_refcounter.h](#)

## 7.192 OsclRefCounterSA< DeallocType > Class Template Reference

```
#include <oscl_refcounter.h>
```

Inheritance diagram for OsclRefCounterSA< DeallocType >:



### Public Member Functions

- [OsclRefCounterSA \(OsclAny \\*p\)](#)
- virtual [~OsclRefCounterSA \(\)](#)
- void [addRef \(\)](#)
- void [removeRef \(\)](#)
- uint32 [getCount \(\)](#)

#### 7.192.1 Detailed Description

**template<class DeallocType> class OsclRefCounterSA< DeallocType >**

Implementation of an [OsclRefCounter](#) that uses a statically created deallocator.

#### 7.192.2 Constructor & Destructor Documentation

**7.192.2.1 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

References NULL, and OSCL\_ASSERT.

**7.192.2.2 template<class DeallocType> virtual OsclRefCounterSA< DeallocType >::~OsclRefCounterSA () [inline, virtual]**

Destructor empty

### 7.192.3 Member Function Documentation

**7.192.3.1 template<class DeallocType> void OsclRefCounterSA< DeallocType >::addRef ()  
[inline, virtual]**

Add to the reference count

Implements [OsclRefCounter](#).

**7.192.3.2 template<class DeallocType> uint32 OsclRefCounterSA< DeallocType >::getCount ()  
[inline, virtual]**

Gets the current number of references

Implements [OsclRefCounter](#).

**7.192.3.3 template<class DeallocType> void OsclRefCounterSA< DeallocType >::removeRef ()  
[inline, virtual]**

Remove from the reference count

Implements [OsclRefCounter](#).

The documentation for this class was generated from the following file:

- [oscl\\_refcounter.h](#)

## 7.193 OsclRegistryAccessClient Class Reference

```
#include <oscl_registry_access_client.h>
```

### Public Member Functions

- OSCL\_IMPORT\_REF OsclRegistryAccessClient ()
- OSCL\_IMPORT\_REF ~OsclRegistryAccessClient ()
- OSCL\_IMPORT\_REF int32 Connect ()
- OSCL\_IMPORT\_REF OsclComponentFactory GetFactory (OSCL\_String &aComponent)
- OSCL\_IMPORT\_REF void GetFactories (OSCL\_String &aRegistry, Oscl\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &aVec)
- OSCL\_IMPORT\_REF void Close ()

#### 7.193.1 Constructor & Destructor Documentation

**7.193.1.1 OSCL\_IMPORT\_REF OsclRegistryAccessClient::OsclRegistryAccessClient ()**

**7.193.1.2 OSCL\_IMPORT\_REF OsclRegistryAccessClient::~OsclRegistryAccessClient ()**

#### 7.193.2 Member Function Documentation

**7.193.2.1 OSCL\_IMPORT\_REF void OsclRegistryAccessClient::Close ()**

Close and cleanup session.

**7.193.2.2 OSCL\_IMPORT\_REF int32 OsclRegistryAccessClient::Connect ()**

Create a session.

#### Returns

OsclErrNone on success.

**7.193.2.3 OSCL\_IMPORT\_REF void OsclRegistryAccessClient::GetFactories (OSCL\_String &*aRegistry*, Oscl\_Vector< OsclRegistryAccessElement, OsclMemAllocator > & *aVec*)**

Get all factories for a given registry type.

#### Parameters

*aRegistry*,: registry MIME type

*aVec*,: output component factory + mimestring vector.

**7.193.2.4 OSCL\_IMPORT\_REF OsclComponentFactory OsclRegistryAccessClient::GetFactory (OSCL\_String & *aComponent*)**

Lookup a factory by registry and component mime type.

**Parameters**

*aComponent*,: registry+component MIME type

**Returns**

Factory. Factory will be NULL if not found.

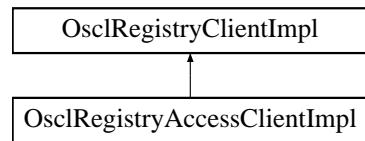
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_access\\_client.h](#)

## 7.194 OsclRegistryAccessClientImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryAccessClientImpl:



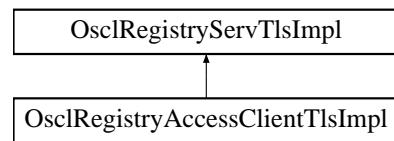
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 7.195 OsclRegistryAccessClientTlsImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryAccessClientTlsImpl:



The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 7.196 OsclRegistryAccessElement Class Reference

```
#include <oscl_registry_types.h>
```

### Data Fields

- [OsclComponentFactory iFactory](#)
- [OSCL\\_HeapString< OsclMemAllocator > iMimeType](#)

#### 7.196.1 Detailed Description

A class used to access the registry data

#### 7.196.2 Field Documentation

##### 7.196.2.1 OsclComponentFactory OsclRegistryAccessElement::iFactory

##### 7.196.2.2 OSCL\_HeapString<OsclMemAllocator> OsclRegistryAccessElement::iMimeType

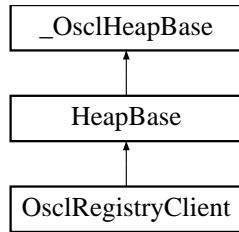
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_types.h](#)

## 7.197 OsclRegistryClient Class Reference

```
#include <oscl_registry_client.h>
```

Inheritance diagram for OsclRegistryClient:



### Public Member Functions

- OSCL\_IMPORT\_REF OsclRegistryClient ()
- OSCL\_IMPORT\_REF ~OsclRegistryClient ()
- OSCL\_IMPORT\_REF int32 Connect (bool aPerThread=false)
- OSCL\_IMPORT\_REF int32 Register (OSCL\_String &aComponentID, OsclComponentFactory aFactory)
- OSCL\_IMPORT\_REF int32 UnRegister (OSCL\_String &aComponentID)
- OSCL\_IMPORT\_REF void Close ()

#### 7.197.1 Constructor & Destructor Documentation

**7.197.1.1 OSCL\_IMPORT\_REF OsclRegistryClient::OsclRegistryClient ()**

**7.197.1.2 OSCL\_IMPORT\_REF OsclRegistryClient::~OsclRegistryClient ()**

#### 7.197.2 Member Function Documentation

**7.197.2.1 OSCL\_IMPORT\_REF void OsclRegistryClient::Close ()**

Close and cleanup. All components registered in this session are automatically unregistered.

**7.197.2.2 OSCL\_IMPORT\_REF int32 OsclRegistryClient::Connect (bool *aPerThread = false*)**

Create a session.

##### Parameters

*aPerThread*,: Select per-thread registry instead of global registry.

##### Returns

OsclErrNone on success.

**7.197.2.3 OSCL\_IMPORT\_REF int32 OsclRegistryClient::Register (OSCL\_String & *aComponentID*, OsclComponentFactory *aFactory*)**

Register a component factory by registry ID and component ID.

**Parameters**

*aComponentID*,: registry + component mime-string.

*aFactory*,: factory function pointer.

*aParam*,: component Create param.

**Returns**

OsclErrNone on success.

**7.197.2.4 OSCL\_IMPORT\_REF int32 OsclRegistryClient::UnRegister (OSCL\_String & *aComponentID*)**

Unregister a previously registered component.

**Returns**

OsclErrNone on success.

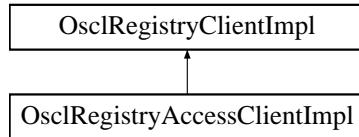
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client.h](#)

## 7.198 OsclRegistryClientImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryClientImpl:



### Protected Member Functions

- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL\\_String &, OsclComponentFactory\)](#)
- int32 [UnRegister \(OSCL\\_String &\)](#)
- [OsclComponentFactory GetFactory \(OSCL\\_String &\)](#)
- void [GetFactories \(OSCL\\_String &, Oscl\\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &\)](#)

### Friends

- class [OsclRegistryClient](#)
- class [OsclRegistryAccessClient](#)

#### 7.198.1 Member Function Documentation

**7.198.1.1 void OsclRegistryClientImpl::Close (void) [inline, protected]**

**7.198.1.2 int32 OsclRegistryClientImpl::Connect () [inline, protected]**

References OsclErrNotSupported.

**7.198.1.3 void OsclRegistryClientImpl::GetFactories (OSCL\_String &, Oscl\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &) [inline, protected]**

**7.198.1.4 OsclComponentFactory OsclRegistryClientImpl::GetFactory (OSCL\_String &) [inline, protected]**

References NULL.

**7.198.1.5 int32 OsclRegistryClientImpl::Register (OSCL\_String &, OsclComponentFactory) [inline, protected]**

References OsclErrNotSupported.

**7.198.1.6 int32 OsclRegistryClientImpl::UnRegister (OSCL\_String &) [inline, protected]**

References OsclErrNotSupported.

**7.198.2 Friends And Related Function Documentation****7.198.2.1 friend class OsclRegistryAccessClient [friend]****7.198.2.2 friend class OsclRegistryClient [friend]**

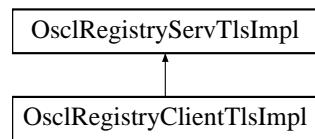
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 7.199 OsclRegistryClientTlsImpl Class Reference

```
#include <oscl_registry_client_impl.h>
```

Inheritance diagram for OsclRegistryClientTlsImpl:



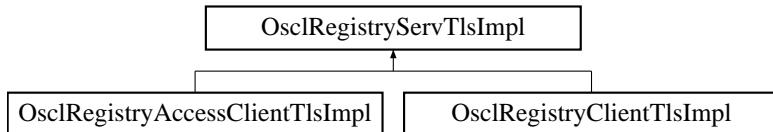
The documentation for this class was generated from the following file:

- [oscl\\_registry\\_client\\_impl.h](#)

## 7.200 OsclRegistryServTlsImpl Class Reference

```
#include <oscl_registry_serv_impl_tls.h>
```

Inheritance diagram for OsclRegistryServTlsImpl:



### Protected Member Functions

- [OsclRegistryServTlsImpl \(\)](#)
- virtual [~OsclRegistryServTlsImpl \(\)](#)
- int32 [Connect \(\)](#)
- void [Close \(\)](#)
- int32 [Register \(OSCL\\_String &aComponentID, OsclComponentFactory aFactory\)](#)
- int32 [UnRegister \(OSCL\\_String &aComponentID\)](#)
- [OsclComponentFactory GetFactory \(OSCL\\_String &aComponent\)](#)
- void [GetFactories \(OSCL\\_String &aRegistry, Oscl\\_Vector< OsclRegistryAccessElement, OsclMemAllocator > &aVec\)](#)

### Friends

- class [OsclRegistryClient](#)
- class [OsclRegistryAccessClient](#)

### 7.200.1 Constructor & Destructor Documentation

7.200.1.1 `OsclRegistryServTlsImpl::OsclRegistryServTlsImpl ()` [protected]

7.200.1.2 `virtual OsclRegistryServTlsImpl::~OsclRegistryServTlsImpl ()` [protected, virtual]

### 7.200.2 Member Function Documentation

7.200.2.1 `void OsclRegistryServTlsImpl::Close ()` [protected]

7.200.2.2 `int32 OsclRegistryServTlsImpl::Connect ()` [protected]

7.200.2.3 `void OsclRegistryServTlsImpl::GetFactories (OSCL_String & aRegistry, Oscl_Vector< OsclRegistryAccessElement, OsclMemAllocator > & aVec)` [protected]

7.200.2.4 `OsclComponentFactory OsclRegistryServTlsImpl::GetFactory (OSCL_String & aComponent)` [protected]

7.200.2.5 `int32 OsclRegistryServTlsImpl::Register (OSCL_String & aComponentID, OsclComponentFactory aFactory)` [protected]

7.200.2.6 `int32 OsclRegistryServTlsImpl::UnRegister (OSCL_String & aComponentID)` [protected]

### 7.200.3 Friends And Related Function Documentation

7.200.3.1 `friend class OsclRegistryAccessClient` [friend]

7.200.3.2 `friend class OsclRegistryClient` [friend]

The documentation for this class was generated from the following file:

- [oscl\\_registry\\_serv\\_impl\\_tls.h](#)

## 7.201 OsclScheduler Class Reference

```
#include <oscl_scheduler.h>
```

### Static Public Member Functions

- static OSCL\_IMPORT\_REF void [Init](#) (const char \*name, [Oscl\\_DefAlloc](#) \*alloc=NULL, int nreserve=20)
- static OSCL\_IMPORT\_REF void [Cleanup](#) ()

#### 7.201.1 Detailed Description

Per-thread scheduler initialization and cleanup.

#### 7.201.2 Member Function Documentation

##### 7.201.2.1 static OSCL\_IMPORT\_REF void OsclScheduler::Cleanup () [static]

This routine uninstalls and destroys Oscl scheduler for the calling thread.

##### 7.201.2.2 static OSCL\_IMPORT\_REF void OsclScheduler::Init (const char \* *name*, Oscl\_DefAlloc \* *alloc* = NULL, int *nreserve* = 20) [static]

This routine creates and installs a scheduler in the calling thread.

#### Parameters

- name*,: (input param) scheduler name.  
*alloc*,: (input param) optional allocator to use for the internal implementation.  
*nreserve*,: (input param) optional value for ready queue reserve size.

The documentation for this class was generated from the following file:

- [oscl\\_scheduler.h](#)

## 7.202 OsclSchedulerObserver Class Reference

```
#include <oscl_scheduler.h>
```

### Public Member Functions

- virtual void [OsclSchedulerTimerCallback](#) ([OsclAny](#) \*aContext, uint32 aDelayMsec)=0
- virtual void [OsclSchedulerReadyCallback](#) ([OsclAny](#) \*aContext)=0
- virtual ~[OsclSchedulerObserver](#) ()

### 7.202.1 Detailed Description

[OsclSchedulerObserver](#) is an observer class for use when running scheduler in non-blocking mode. The scheduler observer can register for callbacks so it will be notified when it is necessary to run scheduler again. Note: non-blocking mode and scheduler callbacks are not supported on Symbian.

### 7.202.2 Constructor & Destructor Documentation

7.202.2.1 virtual [OsclSchedulerObserver](#)::~[OsclSchedulerObserver](#) () [inline, virtual]

### 7.202.3 Member Function Documentation

7.202.3.1 virtual void [OsclSchedulerObserver::OsclSchedulerReadyCallback](#) ([OsclAny](#) \* *aContext*) [pure virtual]

[OsclSchedulerReadyCallback](#) is called when the ready queue is updated, meaning an AO is ready to run. Scheduler needs to be run ASAP. Calling context may be any thread, so be careful!

The current observer is cleared before making the callback, so the observer must call RegisterForCallback again if it wants further notifications.

7.202.3.2 virtual void [OsclSchedulerObserver::OsclSchedulerTimerCallback](#) ([OsclAny](#) \* *aContext*, uint32 *aDelayMsec*) [pure virtual]

[OsclSchedulerTimerCallback](#) is called when the front of the timer queue is updated. This means the minimum delay has changed and scheduler needs to be run again after *aDelayMsec*. Calling context is in-thread.

The current observer is cleared before making the callback, so the observer must call RegisterForCallback again if it wants further notifications.

The documentation for this class was generated from the following file:

- [oscl\\_scheduler.h](#)

## 7.203 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 Member Functions

- [OsclScopedLock](#) (LockClass &*inLock*)  
*Default constructor Initializes the pointer and takes ownership.*
- [~OsclScopedLock](#) ()  
*Destructor.*

### 7.203.1 Detailed Description

**template<class LockClass> class OsclScopedLock< LockClass >**

The [OsclScopedLock](#) class is a template class that handles unlocking an abstract class on destruction. This is very useful for ensuring that the lock is released when the [OsclScopedLock](#) goes out of scope. The purpose of this class is to provide a way to prevent accidental resource leaks in a class or a method, due to "not remembering to unlock" variables which might lead to deadlock conditions.

### 7.203.2 Constructor & Destructor Documentation

#### 7.203.2.1 **template<class LockClass > OsclScopedLock< LockClass >::OsclScopedLock (LockClass & *inLock*) [inline, explicit]**

Default constructor Initializes the pointer and takes ownership.

#### 7.203.2.2 **template<class LockClass > OsclScopedLock< LockClass >::~OsclScopedLock () [inline]**

Destructor.

The pointer is deleted in case this class still has ownership

The documentation for this class was generated from the following file:

- [oscl\\_lock\\_base.h](#)

## 7.204 OsclSelect Class Reference

```
#include <oscl_init.h>
```

### Public Member Functions

- [OsclSelect \(\)](#)
- [OsclSelect \(Oscl\\_DefAlloc \\*erralloc, Oscl\\_DefAlloc \\*schedalloc, const char \\*name, int32 reserve=10, bool heapcheck=false, FILE \\*output=NULL\)](#)

### Data Fields

- bool [iOsclBase](#)
- bool [iOsclMemory](#)
- bool [iOsclErrorTrap](#)
- bool [iOsclLogger](#)
- bool [iOsclScheduler](#)
- Oscl\_DefAlloc \* [iErrAlloc](#)
- Oscl\_DefAlloc \* [iSchedulerAlloc](#)
- const char \* [iSchedulerName](#)
- int32 [iSchedulerReserve](#)
- bool [iHeapCheck](#)
- FILE \* [iOutputFile](#)

### 7.204.1 Detailed Description

Oscl Module selection and Init/Cleanup options.

## 7.204.2 Constructor & Destructor Documentation

7.204.2.1 **OsclSelect::OsclSelect () [inline]**

7.204.2.2 **OsclSelect::OsclSelect (Oscl\_DefAlloc \* *erralloc*, Oscl\_DefAlloc \* *schedalloc*, const char \* *name*, int32 *reserve* = 10, bool *heapcheck* = **false**, FILE \* *output* = **NULL**) [inline]**

## 7.204.3 Field Documentation

7.204.3.1 **Oscl\_DefAlloc\* OsclSelect::iErrAlloc**

7.204.3.2 **bool OsclSelect::iHeapCheck**

7.204.3.3 **bool OsclSelect::iOsclBase**

7.204.3.4 **bool OsclSelect::iOsclErrorTrap**

7.204.3.5 **bool OsclSelect::iOsclLogger**

7.204.3.6 **bool OsclSelect::iOsclMemory**

7.204.3.7 **bool OsclSelect::iOsclScheduler**

7.204.3.8 **FILE\* OsclSelect::iOutputFile**

7.204.3.9 **Oscl\_DefAlloc\* OsclSelect::iSchedulerAlloc**

7.204.3.10 **const char\* OsclSelect::iSchedulerName**

7.204.3.11 **int32 OsclSelect::iSchedulerReserve**

The documentation for this class was generated from the following file:

- [oscl\\_init.h](#)

## 7.205 OsclSemaphore Class Reference

```
#include <oscl_semaphore.h>
```

### Public Member Functions

- OSCL\_IMPORT\_REF OsclSemaphore ()
- OSCL\_IMPORT\_REF ~OsclSemaphore ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Create (uint32 initVal=0)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Close ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Wait ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Wait (uint32 timeout\_msec)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError TryWait ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Signal ()

### 7.205.1 Detailed Description

Class Semaphore

### 7.205.2 Constructor & Destructor Documentation

#### 7.205.2.1 OSCL\_IMPORT\_REF OsclSemaphore::OsclSemaphore ()

Class constructor

#### 7.205.2.2 OSCL\_IMPORT\_REF OsclSemaphore::~OsclSemaphore ()

Class destructor

### 7.205.3 Member Function Documentation

#### 7.205.3.1 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Close ()

Closes the Semaphore

##### Parameters

*It* wont take any parameters

##### Returns

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

#### 7.205.3.2 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Create (uint32 initVal = 0)

Creates the Semaphore

**Parameters**

*Initialcount*

**Returns**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

**7.205.3.3 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Signal ()**

Signals that the thread is finished with the Semaphore

**Parameters**

*It* wont take any parameters

**Returns**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

**7.205.3.4 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::TryWait ()**

Try to acquire semaphore ,if the semaphore is already acquired by another thread, calling thread immediately returns with out blocking

**Parameters**

*It* wont take any parameters

**Returns**

Returns SUCCESS\_ERROR if the semaphore was acquired, SEM\_LOCKED\_ERROR if the semaphore cannot be acquired without waiting, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.205.3.5 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Wait (uint32 timeout\_msec)**

Makes the thread to wait on the Semaphore, with a timeout.

**Parameters**

*timeout* in milliseconds.

**Returns**

Returns SUCCESS\_ERROR if the semaphore was aquired, WAIT\_TIMEOUT\_ERROR if the timeout expired without acquiring the semaphore, or an error code if the operation failed. Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.205.3.6 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclSemaphore::Wait ()**

Makes the thread to wait on the Semaphore

**Parameters**

*It* wont take any parameters

**Returns**

Returns the Error whether it is success or failure incase of failure it will return what is the specific error

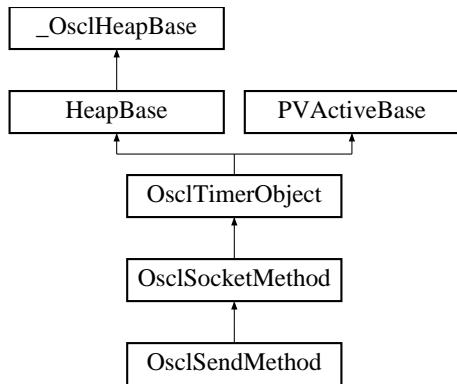
The documentation for this class was generated from the following file:

- [oscl\\_semaphore.h](#)

## 7.206 OsclSendMethod Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OsclSendMethod:



### Public Member Functions

- `~OsclSendMethod ()`
- `TPVSocketEvent Send (const uint8 *&aPtr, uint32 aLen, int32 aTimeout)`
- `uint8 * GetSendData (int32 *aLength)`
- `OsclSendRequest * SendRequest ()`

### Static Public Member Functions

- static `OsclSendMethod * NewL (OsclIPSocketI &c)`

#### 7.206.1 Constructor & Destructor Documentation

##### 7.206.1.1 OsclSendMethod::~OsclSendMethod ()

#### 7.206.2 Member Function Documentation

##### 7.206.2.1 uint8\* OsclSendMethod::GetSendData (int32 \* aLength)

Referenced by `OsclTCPSocketI::GetSendData()`.

##### 7.206.2.2 static OsclSendMethod\* OsclSendMethod::NewL (OsclIPSocketI & c) [static]

##### 7.206.2.3 TPVSocketEvent OsclSendMethod::Send (const uint8 \*& aPtr, uint32 aLen, int32 aTimeout)

Referenced by `OsclTCPSocketI::Send()`.

**7.206.2.4 OsclSendRequest\* OsclSendMethod::SendRequest () [inline]**

References OsclSocketMethod::iSocketRequestAO.

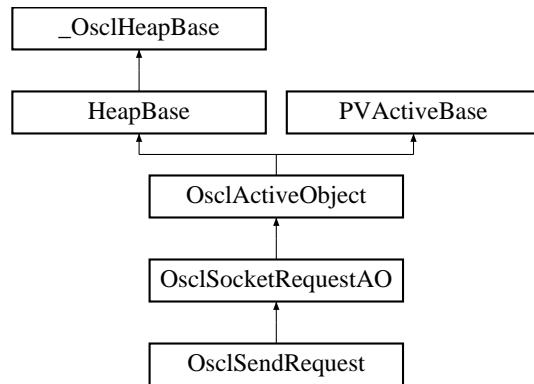
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_send.h](#)

## 7.207 OsclSendRequest Class Reference

```
#include <oscl_socket_send.h>
```

Inheritance diagram for OsclSendRequest:



### Public Member Functions

- [OsclSendRequest \(OsclSocketMethod &c\)](#)
- void [Send \(const uint8 \\*&aPtr, uint32 aLen\)](#)
- void [Success \(\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)

#### 7.207.1 Constructor & Destructor Documentation

[7.207.1.1 OsclSendRequest::OsclSendRequest \(OsclSocketMethod & c\) \[inline\]](#)

#### 7.207.2 Member Function Documentation

[7.207.2.1 uint8\\* OsclSendRequest::GetSendData \(int32 \\* aLength\)](#)

[7.207.2.2 void OsclSendRequest::Send \(const uint8 \\*& aPtr, uint32 aLen\)](#)

[7.207.2.3 void OsclSendRequest::Success \(\) \[virtual\]](#)

Reimplemented from [OsclSocketRequestAO](#).

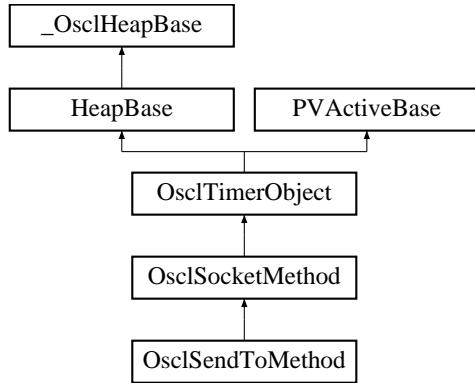
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_send.h](#)

## 7.208 OsclSendToMethod Class Reference

```
#include <oscl_socket_send_to.h>
```

Inheritance diagram for OsclSendToMethod:



### Public Member Functions

- [~OsclSendToMethod \(\)](#)
- [TPVSocketEvent SendTo \(const uint8 \\*&aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeout\)](#)
- [uint8 \\* GetSendData \(int32 \\*aLength\)](#)
- [OsclSendToRequest \\* SendToRequest \(\)](#)

### Static Public Member Functions

- static [OsclSendToMethod \\* NewL \(OsclIPSocketI &c\)](#)

#### 7.208.1 Constructor & Destructor Documentation

##### 7.208.1.1 OsclSendToMethod::~OsclSendToMethod ()

#### 7.208.2 Member Function Documentation

##### 7.208.2.1 uint8\* OsclSendToMethod::GetSendData (int32 \* aLength)

Referenced by OsclUDPSocketI::GetSendData().

##### 7.208.2.2 static OsclSendToMethod\* OsclSendToMethod::NewL (OsclIPSocketI &c) [static]

##### 7.208.2.3 TPVSocketEvent OsclSendToMethod::SendTo (const uint8 \*& aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeout)

Referenced by OsclUDPSocketI::SendTo().

**7.208.2.4 OsclSendToRequest\* OsclSendToMethod::SendToRequest () [inline]**

References OsclSocketMethod::iSocketRequestAO.

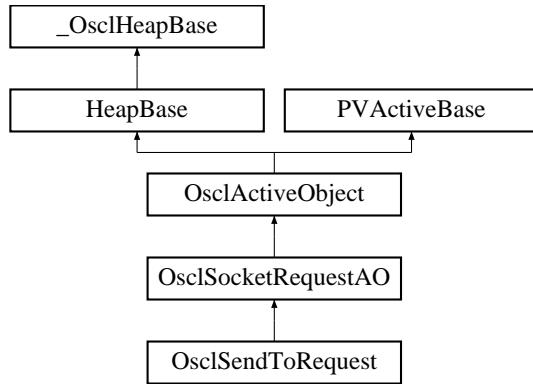
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_send\\_to.h](#)

## 7.209 OsclSendToRequest Class Reference

```
#include <oscl_socket_send_to.h>
```

Inheritance diagram for OsclSendToRequest:



### Public Member Functions

- [OsclSendToRequest \(OsclSocketMethod &c\)](#)
- void [SendTo \(const uint8 \\*&aPtr, uint32 aLen, OsclNetworkAddress &aAddress\)](#)
- void [Success \(\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)

#### 7.209.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.209.2 Constructor & Destructor Documentation

**7.209.2.1 OsclSendToRequest::OsclSendToRequest (OsclSocketMethod & c) [inline]**

#### 7.209.3 Member Function Documentation

**7.209.3.1 uint8\* OsclSendToRequest::GetSendData (int32 \* aLength)**

**7.209.3.2 void OsclSendToRequest::SendTo (const uint8 \*& aPtr, uint32 aLen, OsclNetworkAddress & aAddress)**

**7.209.3.3 void OsclSendToRequest::Success () [virtual]**

Reimplemented from [OsclSocketRequestAO](#).

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_send\\_to.h](#)

## 7.210 OsclSharedPtr< TheClass > Class Template Reference

A parameterized smart pointer class.

```
#include <oscl_shared_ptr.h>
```

### Public Member Functions

- **OsclSharedPtr ()**  
*Constructor.*
- **OsclSharedPtr (TheClass \*inClassPtr, OsclRefCounter \*in\_refcnt)**  
*Constructor.*
- **OsclSharedPtr (const OsclSharedPtr &inSharedPtr)**  
*Copy constructor.*
- **virtual ~OsclSharedPtr ()**  
*Destructor.*
- **TheClass \* operator-> ()**  
• **TheClass & operator\* ()**  
*The indirection operator returns a reference to an object of the parameterized type.*
- **operator TheClass \* ()**  
*Casting operator.*
- **TheClass \* GetRep ()**  
*Use this function to get a pointer to the wrapped object.*
- **OsclRefCounter \* GetRefCounter ()**  
*Get the refcount pointer. This should primarily be used for conversion operations.*
- **int get\_count ()**  
*Get a count of how many references to the object exist.*
- **void Bind (const OsclSharedPtr &inHandle)**  
*Use this function to bind an existing OsclSharedPtr to a already-wrapped object.*
- **void Bind (TheClass \*ptr, OsclRefCounter \*in\_refcnt)**  
*Use this function to bind an existing OsclSharedPtr to a new (unwrapped) object.*
- **void Unbind ()**  
*Use this function of unbind an existing OsclSharedPtr.*
- **OsclSharedPtr & operator= (const OsclSharedPtr &inSharedPtr)**  
*Assignment operator.*
- **bool operator== (const OsclSharedPtr &b) const**  
*Test for equality to see if two PVHandles wrap the same object.*

### 7.210.1 Detailed Description

**template<class TheClass> class OsclSharedPtr< TheClass >**

A parameterized smart pointer class.

### 7.210.2 Constructor & Destructor Documentation

#### 7.210.2.1 template<class TheClass> OsclSharedPtr< TheClass >::OsclSharedPtr () [inline]

Constructor.

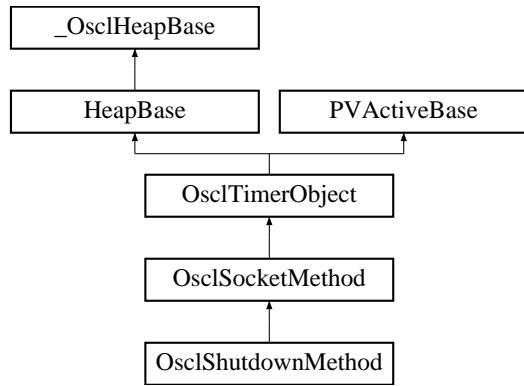
The documentation for this class was generated from the following file:

- [oscl\\_shared\\_ptr.h](#)

## 7.211 OsclShutdownMethod Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OsclShutdownMethod:



### Public Member Functions

- `~OsclShutdownMethod ()`
- `TPVSocketEvent Shutdown (TPVSocketShutdown aHow, int32 aTimeout)`
- `OsclShutdownRequest * ShutdownRequest ()`

### Static Public Member Functions

- static `OsclShutdownMethod * NewL (OsclIPSocketI &c)`

#### 7.211.1 Constructor & Destructor Documentation

##### 7.211.1.1 OsclShutdownMethod::~OsclShutdownMethod ()

#### 7.211.2 Member Function Documentation

##### 7.211.2.1 static OsclShutdownMethod\* OsclShutdownMethod::NewL (OsclIPSocketI &c) [static]

##### 7.211.2.2 TPVSocketEvent OsclShutdownMethod::Shutdown (TPVSocketShutdown aHow, int32 aTimeout)

Referenced by `OsclTCPSocketI::Shutdown()`.

##### 7.211.2.3 OsclShutdownRequest\* OsclShutdownMethod::ShutdownRequest () [inline]

References `OsclSocketMethod::iSocketRequestAO`.

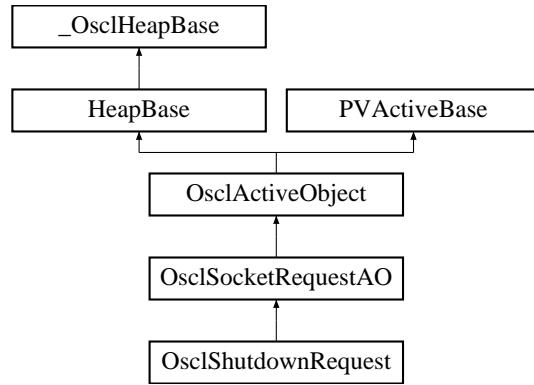
The documentation for this class was generated from the following file:

- `oscl_socket_shutdown.h`

## 7.212 OsclShutdownRequest Class Reference

```
#include <oscl_socket_shutdown.h>
```

Inheritance diagram for OsclShutdownRequest:



### Public Member Functions

- [OsclShutdownRequest \(OsclSocketMethod &c\)](#)
- void [Shutdown \(TPVSocketShutdown aHow\)](#)

#### 7.212.1 Detailed Description

This is the AO that interacts with the socket server

#### 7.212.2 Constructor & Destructor Documentation

7.212.2.1 [OsclShutdownRequest::OsclShutdownRequest \(OsclSocketMethod & c\) \[inline\]](#)

#### 7.212.3 Member Function Documentation

7.212.3.1 [void OsclShutdownRequest::Shutdown \(TPVSocketShutdown aHow\)](#)

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_shutdown.h](#)

## 7.213 OsclSingletonEx< T, ID, Registry > Class Template Reference

```
#include <oscl_error.h>
```

### Public Member Functions

- [OsclSingletonEx \(\)](#)
- [~OsclSingletonEx \(\)](#)
- [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 = OsclSingletonRegistryEx> class OsclSingletonEx< T, ID, Registry >
```

#### 7.213.1 Constructor & Destructor Documentation

**7.213.1.1 template<class T , uint32 ID, class Registry = OsclSingletonRegistryEx> OsclSingletonEx< T, ID, Registry >::OsclSingletonEx () [inline]**

**7.213.1.2 template<class T , uint32 ID, class Registry = OsclSingletonRegistryEx> OsclSingletonEx< T, ID, Registry >::~OsclSingletonEx () [inline]**

#### 7.213.2 Member Function Documentation

**7.213.2.1 template<class T , uint32 ID, class Registry = OsclSingletonRegistryEx> T& OsclSingletonEx< 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.

References OsclSingletonEx< T, ID, Registry >::\_Ptr.

**7.213.2.2 template<class T , uint32 ID, class Registry = OsclSingletonRegistryEx> T\* OsclSingletonEx< 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.

References OsclSingletonEx< T, ID, Registry >::\_Ptr.

**7.213.2.3 template<class T , uint32 ID, class Registry = OsclSingletonRegistryEx> bool OsclSingletonEx< 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.

References OsclSingletonEx< T, ID, Registry >::\_Ptr, and OSCL\_STATIC\_CAST.

### 7.213.3 Field Documentation

**7.213.3.1 template<class T , uint32 ID, class Registry = OsclSingletonRegistryEx> T\* OsclSingletonEx< T, ID, Registry >::\_Ptr [protected]**

Referenced by OsclSingletonEx< T, ID, Registry >::operator\*(), OsclSingletonEx< T, ID, Registry >::operator->(), and OsclSingletonEx< T, ID, Registry >::set().

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 7.214 OsclSingletonRegistryEx Class Reference

```
#include <oscl_error.h>
```

### Static Public Member Functions

- static `OsclAny * getInstance (uint32 ID)`
- static void `registerInstance (OsclAny *ptr, uint32 ID)`
- static `OsclAny * lockAndGetInstance (uint32 ID)`
- static void `registerInstanceAndUnlock (OsclAny *ptr, uint32 ID)`

#### 7.214.1 Member Function Documentation

**7.214.1.1 static OsclAny\* OsclSingletonRegistryEx::getInstance (uint32 *ID*) [inline, static]**

References OsclError::Leave(), and OSCL\_ASSERT.

**7.214.1.2 static OsclAny\* OsclSingletonRegistryEx::lockAndGetInstance (uint32 *ID*) [inline, static]**

References OsclError::Leave(), and OSCL\_ASSERT.

**7.214.1.3 static void OsclSingletonRegistryEx::registerInstance (OsclAny \* *ptr*, uint32 *ID*) [inline, static]**

References OsclError::Leave(), and OSCL\_ASSERT.

**7.214.1.4 static void OsclSingletonRegistryEx::registerInstanceAndUnlock (OsclAny \* *ptr*, uint32 *ID*) [inline, static]**

References OsclError::Leave(), and OSCL\_ASSERT.

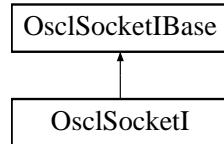
The documentation for this class was generated from the following file:

- `oscl_error.h`

## 7.215 OsclSocketI Class Reference

```
#include <oscl_socket_imp_pv.h>
```

Inheritance diagram for OsclSocketI:



### Public Member Functions

- `~OsclSocketI ()`
- `int32 Open (OsclSocketServI &aServer, uint32 addrFamily, uint32 sockType, uint32 protocol)`
- `int32 Open (OsclSocketServI &aServer)`
- `int32 Bind (OsclNetworkAddress &anAddr)`
- `int32 SetSockOpt (TPVSocketOptionLevel aOptionLevel, TPVSocketOptionName aOptionName, OsclAny *aOptionValue, int32 aOptionLen)`
- `int32 GetPeerName (OsclNetworkAddress &peerName)`
- `int32 Join (OsclNetworkAddress &anAddr)`
- `int32 Close ()`
- `int32 Listen (uint32 qSize)`
- `int32 SetRecvBufferSize (uint32 size)`
- `TPVSocketEvent ThreadLogoff ()`
- `TPVSocketEvent ThreadLogon (OsclSocketServI *aServ)`
- `void Connect (ConnectParam &, OsclSocketRequestAO &)`
- `void Accept (AcceptParam &, OsclSocketRequestAO &)`
- `void Shutdown (ShutdownParam &, OsclSocketRequestAO &)`
- `void Send (SendParam &, OsclSocketRequestAO &)`
- `void SendSuccess (SendParam &)`
- `void SendTo (SendToParam &, OsclSocketRequestAO &)`
- `void SendToSuccess (SendToParam &)`
- `void Recv (RecvParam &, OsclSocketRequestAO &)`
- `void RecvSuccess (RecvParam &)`
- `void RecvFrom (RecvFromParam &, OsclSocketRequestAO &)`
- `void RecvFromSuccess (RecvFromParam &)`
- `TOsclSocket Socket ()`
- `void ProcessConnect (OsclSocketServRequestQElem *)`
- `void ProcessShutdown (OsclSocketServRequestQElem *)`
- `void ProcessAccept (OsclSocketServRequestQElem *)`
- `void ProcessSendTo (OsclSocketServRequestQElem *)`
- `void ProcessRecvFrom (OsclSocketServRequestQElem *)`
- `void ProcessSend (OsclSocketServRequestQElem *)`
- `void ProcessRecv (OsclSocketServRequestQElem *)`
- `PVLogger * Logger ()`

## Static Public Member Functions

- static [OsclSocketI \\* NewL \(Oscl\\_DefAlloc &a\)](#)
- static bool [MakeAddr \(OsclNetworkAddress &in, TOsclSockAddr &addr\)](#)
- static void [MakeAddr \(TOsclSockAddr &in, OsclNetworkAddress &addr\)](#)
- static bool [MakeMulticastGroupInformation \(OsclIpMReq &in, TIpMReq &addr\)](#)
- static void [MakeMulticastGroupInformation \(TIpMReq &in, OsclIpMReq &addr\)](#)

## Friends

- class [OsclAcceptRequest](#)
- class [OsclConnectRequest](#)
- class [OsclRecvRequest](#)
- class [OsclRecvFromRequest](#)
- class [OsclSendRequest](#)
- class [OsclSendToRequest](#)
- class [OsclShutdownRequest](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

### 7.215.1 Detailed Description

Socket implementation class

### 7.215.2 Constructor & Destructor Documentation

#### 7.215.2.1 [OsclSocketI::~OsclSocketI \(\)](#)

### 7.215.3 Member Function Documentation

#### 7.215.3.1 [void OsclSocketI::Accept \(AcceptParam &, OsclSocketRequestAO &\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

#### 7.215.3.2 [int32 OsclSocketI::Bind \(OsclNetworkAddress & anAddr\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

#### 7.215.3.3 [int32 OsclSocketI::Close \(\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

#### 7.215.3.4 [void OsclSocketI::Connect \(ConnectParam &, OsclSocketRequestAO &\) \[virtual\]](#)

Implements [OsclSocketIBase](#).

**7.215.3.5 int32 OsclSocketI::GetPeerName (OsclNetworkAddress & *peerName*)**

**7.215.3.6 int32 OsclSocketI::Join (OsclNetworkAddress & *anAddr*) [virtual]**

Implements [OsclSocketIBase](#).

**7.215.3.7 int32 OsclSocketI::Listen (uint32 *qSize*) [virtual]**

Implements [OsclSocketIBase](#).

Referenced by [OsclTCPSocketI::Listen\(\)](#).

**7.215.3.8 PVLogger\* OsclSocketI::Logger () [inline]**

**7.215.3.9 static void OsclSocketI::MakeAddr (TOsclSockAddr & *in*, OsclNetworkAddress & *addr*) [static]**

**7.215.3.10 static bool OsclSocketI::MakeAddr (OsclNetworkAddress & *in*, TOsclSockAddr & *addr*) [static]**

**7.215.3.11 static void OsclSocketI::MakeMulticastGroupInformation (TIpMReq & *in*, OsclIpMReq & *addr*) [static]**

**7.215.3.12 static bool OsclSocketI::MakeMulticastGroupInformation (OsclIpMReq & *in*, TIpMReq & *addr*) [static]**

**7.215.3.13 static OsclSocketI\* OsclSocketI::NewL (Oscl\_DefAlloc & *a*) [static]**

**7.215.3.14 int32 OsclSocketI::Open (OsclSocketServI & *aServer*) [virtual]**

Implements [OsclSocketIBase](#).

**7.215.3.15 int32 OsclSocketI::Open (OsclSocketServI & *aServer*, uint32 *addrFamily*, uint32 *sockType*, uint32 *protocol*) [virtual]**

Implements [OsclSocketIBase](#).

**7.215.3.16 void OsclSocketI::ProcessAccept (OsclSocketServRequestQElem \*)**

**7.215.3.17 void OsclSocketI::ProcessConnect (OsclSocketServRequestQElem \*)**

**7.215.3.18 void OsclSocketI::ProcessRecv (OsclSocketServRequestQElem \*)**

**7.215.3.19 void OsclSocketI::ProcessRecvFrom (OsclSocketServRequestQElem \*)**

**7.215.3.20 void OsclSocketI::ProcessSend (OsclSocketServRequestQElem \*)**

**7.215.3.21 void OsclSocketI::ProcessSendTo (OsclSocketServRequestQElem \*)**

**7.215.3.22 void OsclSocketI::ProcessShutdown (OsclSocketServRequestQElem \*)**

**7.215.3.23 void OsclSocketI::Recv (RecvParam &, OsclSocketRequestAO &) [virtual]**

Implements [OsclSocketIBase](#).

**7.215.3.24 void OsclSocketI::RecvFrom (RecvFromParam &, OsclSocketRequestAO &) [virtual]**

Implements [OsclSocketIBase](#).

**7.215.3.25 void OsclSocketI::RecvFromSuccess (RecvFromParam &) [virtual]**

Implements [OsclSocketIBase](#).

**7.215.3.26 void OsclSocketI::RecvSuccess (RecvParam &) [virtual]**

Implements [OsclSocketIBase](#).

**7.215.3.27 void OsclSocketI::Send (SendParam &, OsclSocketRequestAO &) [virtual]**

Implements [OsclSocketIBase](#).

**7.215.3.28 void OsclSocketI::SendSuccess (SendParam &) [virtual]**

Implements [OsclSocketIBase](#).

**7.215.3.29 void OsclSocketI::SendTo (SendToParam &, OsclSocketRequestAO &) [virtual]**

Implements [OsclSocketIBase](#).

**7.215.3.30 void OsclSocketI::SendToSuccess (SendToParam &) [virtual]**

Implements [OsclSocketIBase](#).

7.215.3.31 **int32 OsclSocketI::SetRecvBufferSize (uint32 *size*)**

7.215.3.32 **int32 OsclSocketI::SetSockOpt (TPVSocketOptionLevel *aOptionLevel*, TPVSocketOptionName *aOptionName*, OsclAny \* *aOptionValue*, int32 *aOptionLen*)**

7.215.3.33 **void OsclSocketI::Shutdown (ShutdownParam &, OsclSocketRequestAO &)**  
[**virtual**]

Implements [OsclSocketIBase](#).

7.215.3.34 **TOsclSocket OsclSocketI::Socket () [inline]**

7.215.3.35 **TPVSocketEvent OsclSocketI::ThreadLogoff ()**

7.215.3.36 **TPVSocketEvent OsclSocketI::ThreadLogon (OsclSocketServI \* *aServ*)**

## 7.215.4 Friends And Related Function Documentation

7.215.4.1 **friend class OsclAcceptRequest [friend]**

7.215.4.2 **friend class OsclConnectRequest [friend]**

7.215.4.3 **friend class OsclRecvFromRequest [friend]**

7.215.4.4 **friend class OsclRecvRequest [friend]**

7.215.4.5 **friend class OsclSendRequest [friend]**

7.215.4.6 **friend class OsclSendToRequest [friend]**

7.215.4.7 **friend class OsclShutdownRequest [friend]**

7.215.4.8 **friend class OsclTCPSocket [friend]**

Reimplemented from [OsclSocketIBase](#).

7.215.4.9 **friend class OsclUDPSocket [friend]**

Reimplemented from [OsclSocketIBase](#).

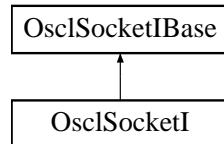
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_imp\\_pv.h](#)

## 7.216 OsclSocketIBase Class Reference

```
#include <oscl_socket_imp_base.h>
```

Inheritance diagram for OsclSocketIBase:



### Public Member Functions

- 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 Member Functions

- static bool HasAsyncBind ()
- static bool HasAsyncListen ()

### Protected Member Functions

- 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 Member Functions

- static int [GetShutdown \(TPVSocketShutdown aOsclVal\)](#)

## Protected Attributes

- [Oscl\\_DefAlloc & iAlloc](#)
- [OsclSocketServI \\* iSocketServ](#)

## Friends

- class [OsclSocketRequest](#)
- class [OsclSocketMethod](#)
- class [OsclSocketRequestAO](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

### 7.216.1 Detailed Description

Socket implementation base class

### 7.216.2 Constructor & Destructor Documentation

**7.216.2.1** virtual OsclSocketIBase::~OsclSocketIBase () [virtual]

**7.216.2.2** OsclSocketIBase::OsclSocketIBase (Oscl\_DefAlloc & *a*) [protected]

### 7.216.3 Member Function Documentation

**7.216.3.1** virtual void OsclSocketIBase::Accept (AcceptParam &, OsclSocketRequestAO &) [pure virtual]

Implemented in [OsclSocketI](#).

**7.216.3.2** virtual int32 OsclSocketIBase::Bind (OsclNetworkAddress & *anAddr*) [pure virtual]

Implemented in [OsclSocketI](#).

**7.216.3.3 virtual void OsclSocketIBase::BindAsync (BindParam &, OsclSocketRequestAO &)**  
[**inline, virtual**]

**7.216.3.4 virtual void OsclSocketIBase::CancelAccept ()** [**protected, pure virtual**]

**7.216.3.5 virtual void OsclSocketIBase::CancelBind ()** [**inline, protected, virtual**]

**7.216.3.6 virtual void OsclSocketIBase::CancelConnect ()** [**protected, pure virtual**]

**7.216.3.7 void OsclSocketIBase::CancelFxn (TPVSocketFxn)**

Referenced by OsclSocketRequestAO::DoCancel().

**7.216.3.8 virtual void OsclSocketIBase::CancelListen ()** [**inline, protected, virtual**]

**7.216.3.9 virtual void OsclSocketIBase::CancelRecv ()** [**protected, pure virtual**]

**7.216.3.10 virtual void OsclSocketIBase::CancelRecvFrom ()** [**protected, pure virtual**]

**7.216.3.11 virtual void OsclSocketIBase::CancelSend ()** [**protected, pure virtual**]

**7.216.3.12 virtual void OsclSocketIBase::CancelSendTo ()** [**protected, pure virtual**]

**7.216.3.13 virtual void OsclSocketIBase::CancelShutdown ()** [**protected, pure virtual**]

**7.216.3.14 virtual int32 OsclSocketIBase::Close ()** [**pure virtual**]

Implemented in [OsclSocketI](#).

**7.216.3.15 virtual void OsclSocketIBase::Connect (ConnectParam &, OsclSocketRequestAO &)**  
[**pure virtual**]

Implemented in [OsclSocketI](#).

**7.216.3.16 static int OsclSocketIBase::GetShutdown (TPVSocketShutdown aOsclVal)** [**static, protected**]

**7.216.3.17 static bool OsclSocketIBase::HasAsyncBind ()** [**static**]

Referenced by OsclUDPSocketI::BindAsync(), and OsclTCPSocketI::BindAsync().

**7.216.3.18 static bool OsclSocketIBase::HasAsyncListen ()** [**static**]

Referenced by OsclTCPSocketI::ListenAsync().

**7.216.3.19 virtual bool OsclSocketIBase::IsOpen () [protected, pure virtual]**

**7.216.3.20 virtual int32 OsclSocketIBase::Join (OsclNetworkAddress & *anAddr*) [pure virtual]**

Implemented in [OsclSocketI](#).

**7.216.3.21 virtual int32 OsclSocketIBase::Listen (uint32 *qSize*) [pure virtual]**

Implemented in [OsclSocketI](#).

**7.216.3.22 virtual void OsclSocketIBase::ListenAsync (ListenParam &, OsclSocketRequestAO & [inline, virtual])**

**7.216.3.23 virtual int32 OsclSocketIBase::Open (OsclSocketServI & *aServer*) [pure virtual]**

Implemented in [OsclSocketI](#).

**7.216.3.24 virtual int32 OsclSocketIBase::Open (OsclSocketServI & *aServer*, uint32 *addrFamily*, uint32 *sockType*, uint32 *protocol*) [pure virtual]**

Implemented in [OsclSocketI](#).

**7.216.3.25 virtual void OsclSocketIBase::Recv (RecvParam &, OsclSocketRequestAO & [pure virtual])**

Implemented in [OsclSocketI](#).

**7.216.3.26 virtual void OsclSocketIBase::RecvFrom (RecvFromParam &, OsclSocketRequestAO &) [pure virtual]**

Implemented in [OsclSocketI](#).

**7.216.3.27 virtual void OsclSocketIBase::RecvFromSuccess (RecvFromParam &) [pure virtual]**

Implemented in [OsclSocketI](#).

**7.216.3.28 virtual void OsclSocketIBase::RecvSuccess (RecvParam &) [pure virtual]**

Implemented in [OsclSocketI](#).

**7.216.3.29 virtual void OsclSocketIBase::Send (SendParam &, OsclSocketRequestAO & [pure virtual])**

Implemented in [OsclSocketI](#).

**7.216.3.30 virtual void OsclSocketIBase::SendSuccess (SendParam &) [pure virtual]**

Implemented in [OsclSocketI](#).

**7.216.3.31 virtual void OsclSocketIBase::SendTo (SendToParam &, OsclSocketRequestAO &) [pure virtual]**

Implemented in [OsclSocketI](#).

**7.216.3.32 virtual void OsclSocketIBase::SendToSuccess (SendToParam &) [pure virtual]**

Implemented in [OsclSocketI](#).

**7.216.3.33 virtual void OsclSocketIBase::Shutdown (ShutdownParam &, OsclSocketRequestAO &) [pure virtual]**

Implemented in [OsclSocketI](#).

## 7.216.4 Friends And Related Function Documentation

**7.216.4.1 friend class OsclSocketMethod [friend]**

**7.216.4.2 friend class OsclSocketRequest [friend]**

**7.216.4.3 friend class OsclSocketRequestAO [friend]**

**7.216.4.4 friend class OsclTCPSocket [friend]**

Reimplemented in [OsclSocketI](#).

**7.216.4.5 friend class OsclUDPSocket [friend]**

Reimplemented in [OsclSocketI](#).

## 7.216.5 Field Documentation

**7.216.5.1 Oscl\_DefAlloc& OsclSocketIBase::iAlloc [protected]**

**7.216.5.2 OsclSocketServI\* OsclSocketIBase::iSocketServ [protected]**

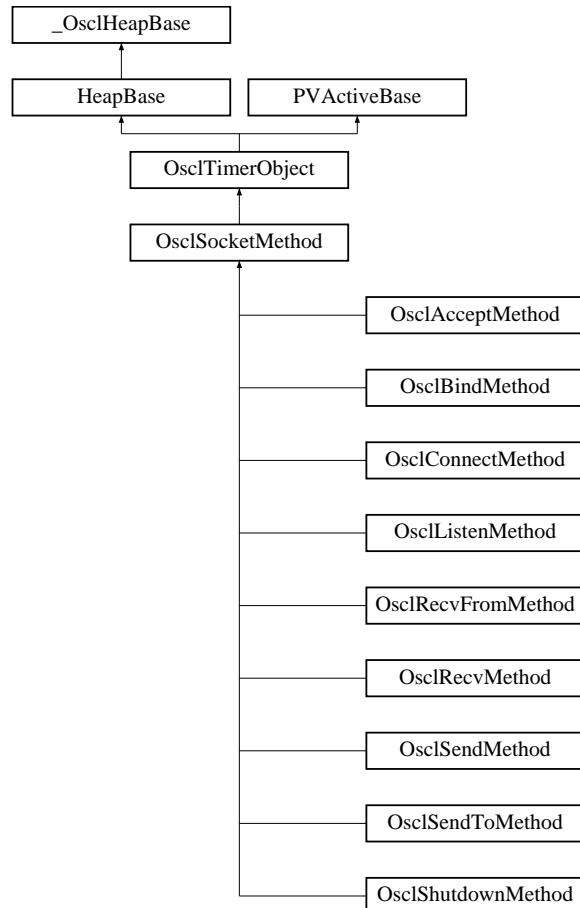
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_imp\\_base.h](#)

## 7.217 OsclSocketMethod Class Reference

```
#include <oscl_socket_method.h>
```

Inheritance diagram for OsclSocketMethod:



### Public Member Functions

- [OsclSocketMethod \(OsclIPSocketI &aContainer, const char \\*name, TPVSocketFxn fxn\)](#)
- virtual [~OsclSocketMethod \(\)](#)
- void [Abort \(\)](#)
- void [AbortAll \(\)](#)
- void [CancelMethod \(\)](#)
- [Oscl\\_DefAlloc & Alloc \(\)](#)
- [TPVSocketEvent ThreadLogon \(\)](#)
- [TPVSocketEvent ThreadLogoff \(\)](#)

### Data Fields

- [OsclIPSocketI & iContainer](#)
- [TPVSocketFxn iSocketFxn](#)

## Protected Member Functions

- void [ConstructL \(OsclSocketRequestAO \\*aAO\)](#)
- bool [StartMethod \(int32 aTimeoutMsec\)](#)
- void [MethodDone \(\)](#)
- void [Run \(\)](#)

## Protected Attributes

- [OsclSocketRequestAO \\* iSocketRequestAO](#)

### 7.217.1 Detailed Description

`OsclSocketMethod` is the base class for all socket methods. Two AOs are required for each socket operation-- one to provide a timeout, and one to detect request completion. The `OsclSocketMethod` class implements the timeout and contains the request completion AO.

### 7.217.2 Constructor & Destructor Documentation

**7.217.2.1 `OsclSocketMethod::OsclSocketMethod (OsclIPSocketI & aContainer, const char * name, TPVSocketFxn f xn) [inline]`**

**7.217.2.2 `virtual OsclSocketMethod::~OsclSocketMethod () [inline, virtual]`**

### 7.217.3 Member Function Documentation

**7.217.3.1 `void OsclSocketMethod::Abort () [inline]`**

References `OsclTimerObject::Cancel()`.

Referenced by `AbortAll()`, `CancelMethod()`, and `OsclSocketRequestAO::RequestDone()`.

**7.217.3.2 `void OsclSocketMethod::AbortAll () [inline]`**

References `OsclSocketRequestAO::Abort()`, `Abort()`, and `iSocketRequestAO`.

**7.217.3.3 `Oscl_DefAlloc& OsclSocketMethod::Alloc () [inline]`**

References `OsclIPSocketI::Alloc()`, and `iContainer`.

**7.217.3.4 `void OsclSocketMethod::CancelMethod () [inline]`**

References `Abort()`, `OsclSocketRequestAO::DoCancel()`, and `iSocketRequestAO`.

Referenced by `OsclTCPSocketI::CancelAccept()`, `OsclUDPSocketI::CancelBind()`, `OsclTCPSocketI::CancelBind()`, `OsclTCPSocketI::CancelConnect()`, `OsclTCPSocketI::CancelListen()`, `OsclTCPSocketI::CancelRecv()`, `OsclUDPSocketI::CancelRecvFrom()`, `OsclTCPSocketI::CancelSend()`, `OsclUDPSocketI::CancelSendTo()`, and `OsclTCPSocketI::CancelShutdown()`.

---

**7.217.3.5 void OsclSocketMethod::ConstructL (OsclSocketRequestAO \* aAO) [inline, protected]**

References iSocketRequestAO, OsclError::Leave(), and OsclErrGeneral.

**7.217.3.6 void OsclSocketMethod::MethodDone () [inline, protected]**

References OsclSocketRequestAO::Abort(), and iSocketRequestAO.

**7.217.3.7 void OsclSocketMethod::Run () [protected, virtual]**

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's WaitForAnyRequest() function completes.

Before calling this active object's [Run\(\)](#) function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

[Run\(\)](#) runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls ExecError() to handle the leave.

Note that once the active scheduler's Start() function has been called, all user code is run under one of the program's active object's [Run\(\)](#) or [RunError\(\)](#) functions.

Implements [PVActiveBase](#).

**7.217.3.8 bool OsclSocketMethod::StartMethod (int32 aTimeoutMsec) [protected]**
**7.217.3.9 TPVSocketEvent OsclSocketMethod::ThreadLogoff ()**
**7.217.3.10 TPVSocketEvent OsclSocketMethod::ThreadLogon ()**

## 7.217.4 Field Documentation

**7.217.4.1 OsclIPSocketI& OsclSocketMethod::iContainer**

Referenced by OsclSocketRequestAO::Alloc(), Alloc(), OsclSocketRequestAO::Id(), OsclSocketRequestAO::SocketI(), and OsclSocketRequestAO::SocketObserver().

**7.217.4.2 TPVSocketFxn OsclSocketMethod::iSocketFxn**

Referenced by OsclSocketRequestAO::DoCancel().

**7.217.4.3 OsclSocketRequestAO\* OsclSocketMethod::iSocketRequestAO [protected]**

Referenced by AbortAll(), OsclAcceptMethod::AcceptRequest(), OsclBindMethod::BindRequest(), CancelMethod(), OsclConnectMethod::ConnectRequest(), ConstructL(), OsclListenMethod::ListenRequest(),

MethodDone(), OsclRecvFromMethod::RecvFromRequest(), OsclRecvMethod::RecvRequest(), OsclSendMethod::SendRequest(), OsclSendToMethod::SendToRequest(), and OsclShutdownMethod::ShutdownRequest().

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_method.h](#)

## 7.218 OsclSocketObserver Class Reference

```
#include <oscl_socket_types.h>
```

### Public Member Functions

- virtual OSCL\_IMPORT\_REF void [HandleSocketEvent](#) (int32 aId, [TPVSocketFxn](#) aFxn, [TPVSocketEvent](#) aEvent, int32 aError)=0
- virtual [~OsclSocketObserver](#) ()

#### 7.218.1 Detailed Description

Socket event observer. The client implements this to get asynchronous command completion.

#### 7.218.2 Constructor & Destructor Documentation

7.218.2.1 virtual [OsclSocketObserver::~OsclSocketObserver](#) () [inline, virtual]

#### 7.218.3 Member Function Documentation

7.218.3.1 virtual OSCL\_IMPORT\_REF void [OsclSocketObserver::HandleSocketEvent](#) (int32 *aId*, [TPVSocketFxn](#) *aFxn*, [TPVSocketEvent](#) *aEvent*, int32 *aError*) [pure virtual]

Socket Event callback.

##### Parameters

*aId*,: The ID that was supplied when the socket was created.

*aFxn*,: Type of socket function call.

*aEvent*,: Function completion event. Will be EPVSocketSuccess, EPVSocketTimeout, or EPVSocketFailure.

*aError*,: When the event is EPVSocketFailure, this may contain a platform-specific error code, or zero if none is available.

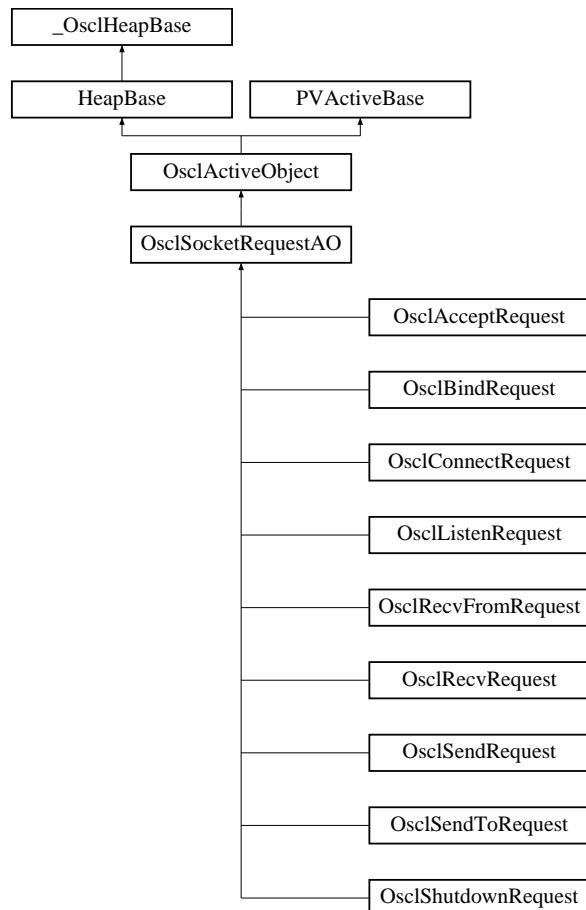
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_types.h](#)

## 7.219 OsclSocketRequestAO Class Reference

```
#include <oscl_socket_method.h>
```

Inheritance diagram for OsclSocketRequestAO:



### Public Member Functions

- void [ConstructL](#) ()

### Protected Member Functions

- [OsclSocketRequestAO](#) ([OsclSocketMethod](#) &aContainer, const char \*name)
- virtual ~[OsclSocketRequestAO](#) ()
- [OsclAny](#) \* [NewRequest](#) (const uint32 size)
- void [CleanupParam](#) (bool deallocate=false)
- void [Abort](#) ()
- void [RequestDone](#) ()
- int [GetSocketError](#) ()
- void [DoCancel](#) ()
- void [Run](#) ()

- virtual void [Success](#) ()
- [OsclSocketI \\* SocketI](#) ()
- [OsclSocketObserver \\* SocketObserver](#) ()
- uint32 [Id](#) ()
- [Oscl\\_DefAlloc & Alloc](#) ()

## Protected Attributes

- [OsclSocketMethod & iContainer](#)
- int32 [iSocketError](#)
- [SocketRequestParam \\* iParam](#)
- uint32 [iParamSize](#)

## Friends

- class [OsclSocketI](#)
- class [OsclSocketMethod](#)
- class [OsclSocketRequest](#)

### 7.219.1 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

### 7.219.2 Constructor & Destructor Documentation

**7.219.2.1 OsclSocketRequestAO::OsclSocketRequestAO ([OsclSocketMethod & aContainer](#), const char \* *name*) [inline, protected]**

**7.219.2.2 virtual OsclSocketRequestAO::~OsclSocketRequestAO () [inline, protected, virtual]**

References [CleanupParam\(\)](#).

### 7.219.3 Member Function Documentation

**7.219.3.1 void OsclSocketRequestAO::Abort () [inline, protected]**

References [OsclActiveObject::Cancel\(\)](#).

Referenced by [OsclSocketMethod::AbortAll\(\)](#), and [OsclSocketMethod::MethodDone\(\)](#).

**7.219.3.2 Oscl\_DefAlloc& OsclSocketRequestAO::Alloc () [inline, protected]**

References [OsclIPSocketI::Alloc\(\)](#), [OsclSocketMethod::iContainer](#), and [iContainer](#).

**7.219.3.3 void OsclSocketRequestAO::CleanupParam (bool *deallocate = false*) [protected]**

Referenced by [~OsclSocketRequestAO\(\)](#).

**7.219.3.4 void OsclSocketRequestAO::ConstructL () [inline]****7.219.3.5 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](#).

References OsclSocketIBase::CancelFxn(), iContainer, OsclSocketMethod::iSocketFxn, and SocketI().

Referenced by OsclSocketMethod::CancelMethod().

**7.219.3.6 int OsclSocketRequestAO::GetSocketError () [inline, protected]****7.219.3.7 uint32 OsclSocketRequestAO::Id () [inline, protected]**

References OsclSocketMethod::iContainer, iContainer, and OsclIPSocketI::iId.

**7.219.3.8 OsclAny\* OsclSocketRequestAO::NewRequest (const uint32 size) [protected]****7.219.3.9 void OsclSocketRequestAO::RequestDone () [inline, protected]**

References OsclSocketMethod::Abort(), and iContainer.

**7.219.3.10 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](#).

**7.219.3.11 OsclSocketI\* OsclSocketRequestAO::SocketI () [inline, protected]**

References OsclSocketMethod::iContainer, iContainer, and OsclIPSocketI::iSocket.

Referenced by DoCancel().

**7.219.3.12 OsclSocketObserver\* OsclSocketRequestAO::SocketObserver () [inline, protected]**

References OsclSocketMethod::iContainer, iContainer, and OsclIPSocketI::iObserver.

**7.219.3.13 virtual void OsclSocketRequestAO::Success () [inline, protected, virtual]**

Reimplemented in [OsclRecvRequest](#), [OsclRecvFromRequest](#), [OsclSendRequest](#), and [OsclSendToRequest](#).

## 7.219.4 Friends And Related Function Documentation

**7.219.4.1 friend class OsclSocketI [friend]**

**7.219.4.2 friend class OsclSocketMethod [friend]**

**7.219.4.3 friend class OsclSocketRequest [friend]**

## 7.219.5 Field Documentation

**7.219.5.1 OsclSocketMethod& OsclSocketRequestAO::iContainer [protected]**

Referenced by Alloc(), DoCancel(), Id(), RequestDone(), SocketI(), and SocketObserver().

**7.219.5.2 SocketRequestParam\* OsclSocketRequestAO::iParam [protected]**

**7.219.5.3 uint32 OsclSocketRequestAO::iParamSize [protected]**

**7.219.5.4 int32 OsclSocketRequestAO::iSocketError [protected]**

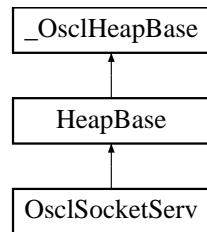
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_method.h](#)

## 7.220 OsclSocketServ Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclSocketServ:



### Public Member Functions

- OSCL\_IMPORT\_REF ~OsclSocketServ ()
- OSCL\_IMPORT\_REF int32 Connect (uint32 aMessageSlots=8, bool aShareSession=false)
- OSCL\_IMPORT\_REF void Close (bool aCleanup=true)

### Static Public Member Functions

- static OSCL\_IMPORT\_REF OsclSocketServ \* NewL (Oscl\_DefAlloc &alloc)

### Friends

- class OsclTCPSocket
- class OsclUDPSocket
- class OsclDNS

#### 7.220.1 Member Function Documentation

##### 7.220.1.1 static 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

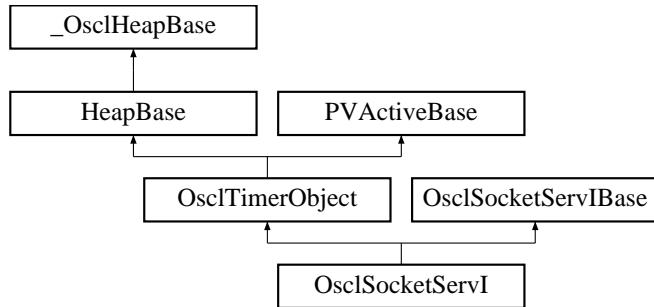
The documentation for this class was generated from the following file:

- [oscl\\_socket.h](#)

## 7.221 OsclSocketServI Class Reference

```
#include <oscl_socket_serv_imp_pv.h>
```

Inheritance diagram for OsclSocketServI:



### Public Member Functions

- int32 [Connect](#) (uint32 aMessageSlots, bool aShareSession)
- void [Close](#) (bool)
- bool [IsServerThread](#) ()

### Static Public Member Functions

- static [OsclSocketServI \\* NewL](#) ([Oscl\\_DefAlloc](#) &a)

### Friends

- class [OsclSocketServRequestList](#)
- class [LoopbackSocket](#)
- class [OsclTCPSocketI](#)
- class [OsclUDPSocketI](#)
- class [OsclSocketI](#)
- class [OsclDNSI](#)
- class [OsclSocketRequest](#)
- class [OsclSocketServ](#)

### 7.221.1 Detailed Description

PV socket server implementation

### 7.221.2 Member Function Documentation

#### 7.221.2.1 void OsclSocketServI::Close (bool) [virtual]

Implements [OsclSocketServIBase](#).

**7.221.2.2 int32 OsclSocketServI::Connect (uint32 *aMessageSlots*, bool *aShareSession*) [virtual]**

Implements [OsclSocketServIBase](#).

**7.221.2.3 bool OsclSocketServI::IsServerThread ()**

**7.221.2.4 static OsclSocketServI\* OsclSocketServI::NewL (Oscl\_DefAlloc & *a*) [static]**

### 7.221.3 Friends And Related Function Documentation

**7.221.3.1 friend class LoopbackSocket [friend]**

**7.221.3.2 friend class OsclDNSI [friend]**

**7.221.3.3 friend class OsclSocketI [friend]**

**7.221.3.4 friend class OsclSocketRequest [friend]**

**7.221.3.5 friend class OsclSocketServ [friend]**

**7.221.3.6 friend class OsclSocketServRequestList [friend]**

**7.221.3.7 friend class OsclTCPSocketI [friend]**

**7.221.3.8 friend class OsclUDPSocketI [friend]**

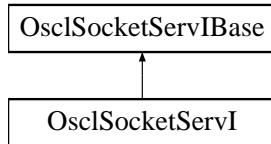
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_serv\\_imp\\_pv.h](#)

## 7.222 OsclSocketServIBase Class Reference

```
#include <oscl_socket_serv_imp_base.h>
```

Inheritance diagram for OsclSocketServIBase:



### Public Member Functions

- virtual ~OsclSocketServIBase ()
- virtual int32 Connect (uint32 aMessageSlots, bool aShareSession)=0
- virtual void Close (bool)=0

### Data Fields

- PVLogger \* iLogger

### Protected Types

- enum TSocketServState { ESocketServ\_Idle, ESocketServ\_Connected, ESocketServ\_Error }

### Protected Member Functions

- OsclSocketServIBase (Oscl\_DefAlloc &a)
- TSocketServState State () const
- bool IsServConnected () const

### Protected Attributes

- Oscl\_DefAlloc & iAlloc
- TSocketServState iServState
- int iServError

#### 7.222.1 Detailed Description

Socket Server implementation Base class common to all implementations

#### 7.222.2 Member Enumeration Documentation

##### 7.222.2.1 enum OsclSocketServIBase::TSocketServState [protected]

Enumerator:

*ESocketServ\_Idle*

*ESocketServ\_Connected*  
*ESocketServ\_Error*

### 7.222.3 Constructor & Destructor Documentation

7.222.3.1 **virtual OsclSocketServIBase::~OsclSocketServIBase () [inline, virtual]**

7.222.3.2 **OsclSocketServIBase::OsclSocketServIBase (Oscl\_DefAlloc & a) [inline, protected]**

References ESocketServ\_Idle, iLogger, iServError, iServState, and NULL.

### 7.222.4 Member Function Documentation

7.222.4.1 **virtual void OsclSocketServIBase::Close (bool) [pure virtual]**

Implemented in [OsclSocketServI](#).

7.222.4.2 **virtual int32 OsclSocketServIBase::Connect (uint32 aMessageSlots, bool aShareSession) [pure virtual]**

Implemented in [OsclSocketServI](#).

7.222.4.3 **bool OsclSocketServIBase::IsServConnected () const [inline, protected]**

References ESocketServ\_Connected, and iServState.

7.222.4.4 **TSocketServState OsclSocketServIBase::State () const [inline, protected]**

References iServState.

### 7.222.5 Field Documentation

7.222.5.1 **Oscl\_DefAlloc& OsclSocketServIBase::iAlloc [protected]**

7.222.5.2 **PVLogger\* OsclSocketServIBase::iLogger**

Referenced by OsclSocketServIBase().

7.222.5.3 **int OsclSocketServIBase::iServError [protected]**

Referenced by OsclSocketServIBase().

7.222.5.4 **TSocketServState OsclSocketServIBase::iServState [protected]**

Referenced by IsServConnected(), OsclSocketServIBase(), and State().

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_serv\\_imp\\_base.h](#)

## 7.223 OsclSocketServRequestList Class Reference

```
#include <oscl_socket_serv_imp_reqlist.h>
```

### Public Member Functions

- [OsclSocketServRequestList \(\)](#)
- void [Add \(OsclSocketRequest \\*\)](#)
- void [StartCancel \(OsclSocketRequest \\*\)](#)
- void [Open \(OsclSocketServI \\*s\)](#)
- void [Close \(\)](#)
- void [Wakeup \(\)](#)
- void [WaitOnRequests \(\)](#)
- void [Remove \(OsclSocketServRequestQElem \\*aElem\)](#)

### Friends

- class [OsclSocketServI](#)

### 7.223.1 Detailed Description

PV socket server request queue

### 7.223.2 Constructor & Destructor Documentation

#### 7.223.2.1 OsclSocketServRequestList::OsclSocketServRequestList ()

### 7.223.3 Member Function Documentation

#### 7.223.3.1 void OsclSocketServRequestList::Add (OsclSocketRequest \*)

#### 7.223.3.2 void OsclSocketServRequestList::Close ()

#### 7.223.3.3 void OsclSocketServRequestList::Open (OsclSocketServI \* s)

#### 7.223.3.4 void OsclSocketServRequestList::Remove (OsclSocketServRequestQElem \* aElem) [[inline](#)]

References [OsclSocketServRequestQElem::iSocketRequest](#), and [NULL](#).

**7.223.3.5 void OsclSocketServRequestList::StartCancel (OsclSocketRequest \*)**

**7.223.3.6 void OsclSocketServRequestList::WaitOnRequests ()**

**7.223.3.7 void OsclSocketServRequestList::Wakeup ()**

## **7.223.4 Friends And Related Function Documentation**

**7.223.4.1 friend class OsclSocketServI [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_serv\\_imp\\_reqlist.h](#)

## 7.224 OsclSocketServRequestQElem Class Reference

```
#include <oscl_socket_serv_imp_reqlist.h>
```

### Public Member Functions

- [OsclSocketServRequestQElem \(OsclSocketRequest \\*r\)](#)

### Data Fields

- OsclSocketRequest \* [iSocketRequest](#)
- uint8 [iSelect](#)
- bool [iCancel](#)

#### 7.224.1 Constructor & Destructor Documentation

7.224.1.1 **OsclSocketServRequestQElem::OsclSocketServRequestQElem (OsclSocketRequest \* r)**  
[[inline](#)]

#### 7.224.2 Field Documentation

7.224.2.1 **bool OsclSocketServRequestQElem::iCancel**

7.224.2.2 **uint8 OsclSocketServRequestQElem::iSelect**

7.224.2.3 **OsclSocketRequest\* OsclSocketServRequestQElem::iSocketRequest**

Referenced by [OsclSocketServRequestList::Remove\(\)](#).

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_serv\\_imp\\_reqlist.h](#)

## 7.225 OsclSocketTOS Class Reference

```
#include <oscl_socket_types.h>
```

### Public Types

- enum **TPVServicePrecedence** {
   
    **EPVRoutine** = 0, **EPVPriority** = 1, **EPVImmediate** = 2, **EPVFlash** = 3,
   
    **EPVOverrideFlash** = 4, **EPVCritic\_Ecp** = 5, **EPVInetControl** = 6, **EPVNetControl** = 7 }
- enum **TPVServicePriority** { **EPVNoTOS** = 0x0, **EPVLDelay** = (1 << 4), **EPVHiThrpt** = (1 << 3),
 **EPVHiRel** = (1 << 2) }

### Public Member Functions

- **OsclSocketTOS** ()
- void **SetPrecedence** (**TPVServicePrecedence** aPrecedence)
- void **SetPriority** (bool aMinimizeDelay, bool aMaximizeThroughput, bool MaximizeReliability)
- void **ClearTOS** ()
- uint8 **GetTOS** () const

#### 7.225.1 Member Enumeration Documentation

##### 7.225.1.1 enum OsclSocketTOS::TPVServicePrecedence

Enumerator:

*EPVRoutine*  
*EPVPriority*  
*EPVImmediate*  
*EPVFlash*  
*EPVOverrideFlash*  
*EPVCritic\_Ecp*  
*EPVInetControl*  
*EPVNetControl*

##### 7.225.1.2 enum OsclSocketTOS::TPVServicePriority

Enumerator:

*EPVNoTOS*  
*EPVLDelay*  
*EPVHiThrpt*  
*EPVHiRel*

## 7.225.2 Constructor & Destructor Documentation

### 7.225.2.1 OsclSocketTOS::OsclSocketTOS () [inline]

References ClearTOS().

## 7.225.3 Member Function Documentation

### 7.225.3.1 void OsclSocketTOS::ClearTOS () [inline]

References EPVNoTOS, and EPVRoutine.

Referenced by OsclSocketTOS().

### 7.225.3.2 uint8 OsclSocketTOS::GetTOS () const [inline]

Format of Ip Header's TOS field as specified in RFC 791  
0 1 2 3 4 5 6 7 +-----+-----+-----+-----+  
+-----+-----+ | PRECEDENCE | D | T | R | 0 | 0 | +-----+-----+-----+-----+-----+-----+

### 7.225.3.3 void OsclSocketTOS::SetPrecedence (TPVServicePrecedence *aPrecedence*) [inline]

### 7.225.3.4 void OsclSocketTOS::SetPriority (bool *aMinimizeDelay*, bool *aMaximizeThroughput*, bool *MaximizeReliability*) [inline]

References EPVHiRel, EPVHiThrpt, EPVLDelay, and EPVNoTOS.

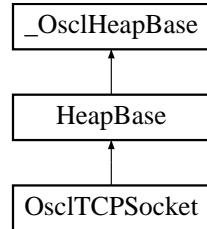
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_types.h](#)

## 7.226 OsclTCPSocket Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclTCPSocket:



### Public Member Functions

- OSCL\_IMPORT\_REF ~OsclTCPSocket ()
- OSCL\_IMPORT\_REF TPVSocketEvent ThreadLogoff ()
- OSCL\_IMPORT\_REF TPVSocketEvent ThreadLogon (OsclSocketServ &aServ, OsclSocketObserver \*aObserver)
- OSCL\_IMPORT\_REF int32 Close ()
- OSCL\_IMPORT\_REF int32 Bind (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF TPVSocketEvent BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void CancelBind ()
- OSCL\_IMPORT\_REF int32 SetOptionToReuseAddress ()
- OSCL\_IMPORT\_REF int32 SetTOS (const OsclSocketTOS &aTOS)
- OSCL\_IMPORT\_REF int32 GetPeerName (OsclNetworkAddress &aPeerName)
- OSCL\_IMPORT\_REF int32 Listen (int32 aQueueSize)
- OSCL\_IMPORT\_REF TPVSocketEvent ListenAsync (int32 aQueueSize, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void CancelListen ()
- OSCL\_IMPORT\_REF OsclTCPSocket \* GetAcceptedSocketL (uint32 aId)
- OSCL\_IMPORT\_REF uint8 \* GetRecvData (int32 \*aLength)
- OSCL\_IMPORT\_REF uint8 \* GetSendData (int32 \*aLength)
- OSCL\_IMPORT\_REF TPVSocketEvent Connect (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelConnect ()
- OSCL\_IMPORT\_REF TPVSocketEvent Shutdown (TPVSocketShutdown aHow, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelShutdown ()
- OSCL\_IMPORT\_REF TPVSocketEvent Accept (int32 aTimeout=-1)
- OSCL\_IMPORT\_REF void CancelAccept ()
- OSCL\_IMPORT\_REF TPVSocketEvent Send (const uint8 \*aPtr, uint32 aLen, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelSend ()
- OSCL\_IMPORT\_REF TPVSocketEvent Recv (uint8 \*aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelRecv ()

## Static Public Member Functions

- static OSCL\_IMPORT\_REF OsclTCPSocket \* NewL (Oscl\_DefAlloc &alloc, OsclSocketServ &aServ, OsclSocketObserver \*aObserver, uint32 aId)

### 7.226.1 Detailed Description

The TCP Socket class

### 7.226.2 Member Function Documentation

#### 7.226.2.1 static 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.

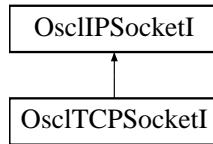
The documentation for this class was generated from the following file:

- [oscl\\_socket.h](#)

## 7.227 OsclTCPSocketI Class Reference

```
#include <oscl_tcp_socket.h>
```

Inheritance diagram for OsclTCPSocketI:



### Public Member Functions

- virtual ~OsclTCPSocketI ()
- [TPVSocketEvent ThreadLogoff \(\)](#)
- [TPVSocketEvent ThreadLogon \(OsclSocketServI \\*aServ, OsclSocketObserver \\*aObserver\)](#)
- int32 [Close \(\)](#)
- int32 [Listen \(int aQueueSize\)](#)
- [OsclTCPSocketI \\* GetAcceptedSocketL \(uint32 aId\)](#)
- uint8 \* [GetRecvData \(int32 \\*aLength\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)
- [TPVSocketEvent BindAsync \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelBind \(\)](#)
- [TPVSocketEvent ListenAsync \(uint32 qsize, int32 aTimeoutMsec=-1\)](#)
- void [CancelListen \(\)](#)
- [TPVSocketEvent Connect \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelConnect \(\)](#)
- [TPVSocketEvent Shutdown \(TPVSocketShutdown aHow, int32 aTimeoutMsec=-1\)](#)
- void [CancelShutdown \(\)](#)
- [TPVSocketEvent Accept \(int32 aTimeout=-1\)](#)
- void [CancelAccept \(\)](#)
- [TPVSocketEvent Send \(const uint8 \\*&aPtr, uint32 aLen, int32 aTimeoutMsec=-1\)](#)
- void [CancelSend \(\)](#)
- [TPVSocketEvent Recv \(uint8 \\*&aPtr, uint32 aMaxLen, int32 aTimeoutMsec=-1\)](#)
- void [CancelRecv \(\)](#)

### Static Public Member Functions

- static OsclTCPSocketI \* [NewL \(Oscl\\_DefAlloc &a, OsclSocketServI \\*aServ, OsclSocketObserver \\*aObserver, uint32 aId\)](#)

#### 7.227.1 Detailed Description

Internal implementation class for [OsclTCPSocket](#)

## 7.227.2 Constructor & Destructor Documentation

**7.227.2.1 virtual OsclTCPSocketI::~OsclTCPSocketI () [virtual]**

## 7.227.3 Member Function Documentation

**7.227.3.1 TPVSocketEvent OsclTCPSocketI::Accept (int32 *aTimeout* = -1) [inline]**

References OsclAcceptMethod::Accept(), EPVSocketFailure, and OsclIPSocketI::iObserver.

**7.227.3.2 TPVSocketEvent OsclTCPSocketI::BindAsync (OsclNetworkAddress & *aAddress*, int32 *aTimeoutMsec* = -1) [inline]**

References OsclBindMethod::Bind(), EPVSocketFailure, EPVSocketNotImplemented, OsclSocketIBase::HasAsyncBind(), OsclIPSocketI::iAddress, OsclIPSocketI::iObserver, OsclNetworkAddress::ipAddr, OsclNetworkAddress::port, OsclNameString< \_\_len >::Set(), and OsclNameString< \_\_len >::Str().

**7.227.3.3 void OsclTCPSocketI::CancelAccept () [inline]**

References OsclSocketMethod::CancelMethod().

**7.227.3.4 void OsclTCPSocketI::CancelBind () [inline]**

References OsclSocketMethod::CancelMethod().

**7.227.3.5 void OsclTCPSocketI::CancelConnect () [inline]**

References OsclSocketMethod::CancelMethod().

**7.227.3.6 void OsclTCPSocketI::CancelListen () [inline]**

References OsclSocketMethod::CancelMethod().

**7.227.3.7 void OsclTCPSocketI::CancelRecv () [inline]**

References OsclSocketMethod::CancelMethod().

**7.227.3.8 void OsclTCPSocketI::CancelSend () [inline]**

References OsclSocketMethod::CancelMethod().

**7.227.3.9 void OsclTCPSocketI::CancelShutdown () [inline]**

References OsclSocketMethod::CancelMethod().

**7.227.3.10 int32 OsclTCPSocketI::Close () [virtual]**

Implements [OsclIPSocketI](#).

**7.227.3.11 TPVSocketEvent OsclTCPSocketI::Connect (OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]**

References [OsclConnectMethod::Connect\(\)](#), [EPVSocketFailure](#), and [OsclIPSocketI::iObserver](#).

**7.227.3.12 OsclTCPSocketI\* OsclTCPSocketI::GetAcceptedSocketL (uint32 aId)****7.227.3.13 uint8 \* OsclTCPSocketI::GetRecvData (int32 \* aLength) [inline, virtual]**

Implements [OsclIPSocketI](#).

References [OsclRecvMethod::GetRecvData\(\)](#).

**7.227.3.14 uint8 \* OsclTCPSocketI::GetSendData (int32 \* aLength) [inline, virtual]**

Implements [OsclIPSocketI](#).

References [OsclSendMethod::GetSendData\(\)](#).

**7.227.3.15 int32 OsclTCPSocketI::Listen (int aQueueSize) [inline]**

References [OsclIPSocketI::iSocket](#), and [OsclSocketI::Listen\(\)](#).

**7.227.3.16 TPVSocketEvent OsclTCPSocketI::ListenAsync (uint32 qsize, int32 aTimeoutMsec = -1) [inline]**

References [EPVSocketFailure](#), [EPVSocketNotImplemented](#), [OsclSocketIBase::HasAsyncListen\(\)](#), [OsclIPSocketI::iObserver](#), and [OsclListenMethod::Listen\(\)](#).

**7.227.3.17 static OsclTCPSocketI\* OsclTCPSocketI::NewL (Oscl\_DefAlloc & a, OsclSocketServI \* aServ, OsclSocketObserver \* aObserver, uint32 aId) [static]****7.227.3.18 TPVSocketEvent OsclTCPSocketI::Recv (uint8 \*& aPtr, uint32 aMaxLen, int32 aTimeoutMsec = -1) [inline]**

References [EPVSocketFailure](#), [OsclIPSocketI::iObserver](#), and [OsclRecvMethod::Recv\(\)](#).

**7.227.3.19 TPVSocketEvent OsclTCPSocketI::Send (const uint8 \*& aPtr, uint32 aLen, int32 aTimeoutMsec = -1) [inline]**

References [EPVSocketFailure](#), [OsclIPSocketI::iObserver](#), and [OsclSendMethod::Send\(\)](#).

**7.227.3.20 TPVSocketEvent OsclTCPSocketI::Shutdown (TPVSocketShutdown *aHow*, int32 *aTimeoutMsec* = -1) [inline]**

References EPVSocketFailure, OsclIPSocketI::iObserver, and OsclShutdownMethod::Shutdown().

**7.227.3.21 TPVSocketEvent OsclTCPSocketI::ThreadLogoff ()**

Reimplemented from [OsclIPSocketI](#).

**7.227.3.22 TPVSocketEvent OsclTCPSocketI::ThreadLogon (OsclSocketServI \* *aServ*, OsclSocketObserver \* *aObserver*)**

The documentation for this class was generated from the following file:

- [oscl\\_tcp\\_socket.h](#)

## 7.228 OsclThread Class Reference

```
#include <oscl_thread.h>
```

### Public Member Functions

- OSCL\_IMPORT\_REF OsclThread ()
- OSCL\_IMPORT\_REF ~OsclThread ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Create (TOsclThreadFuncPtr func, int32 stack\_size, TOsclThreadFuncArg argument, OsclThread\_State state=Start\_on\_creation, bool oIsJoinable=false)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError GetPriority (OsclThreadPriority &refThreadPriority)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError SetPriority (OsclThreadPriority ePriority)
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Suspend ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Resume ()
- OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError Terminate (OsclAny \*exitcode)
- OSCL\_IMPORT\_REF TOsclThreadTerminate CanTerminate ()

### Static Public Member Functions

- static OSCL\_IMPORT\_REF void Exit (OsclAny \*exitcode)
- static OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError GetId (TOsclThreadId &refThreadId)
- static OSCL\_IMPORT\_REF bool CompareId (TOsclThreadId &t1, TOsclThreadId &t2)
- static OSCL\_IMPORT\_REF void SleepMillisec (const int32 msec)

### 7.228.1 Detailed Description

Thread Class. A subset of Thread APIs. It implements platform independent APIs for thread creation, exiting, suspend, resume, priority and termination. With the use of proper defines it implements the basic thread features. It provides an opaque layer through which user doesn't need to worry about OS specific data.

### 7.228.2 Constructor & Destructor Documentation

#### 7.228.2.1 OSCL\_IMPORT\_REF OsclThread::OsclThread ()

Class constructor

#### 7.228.2.2 OSCL\_IMPORT\_REF OsclThread::~OsclThread ()

Class destructor

### 7.228.3 Member Function Documentation

#### 7.228.3.1 OSCL\_IMPORT\_REF TOsclThreadTerminate OsclThread::CanTerminate ()

Tell if thread terminate will do join, immediate hard kill, or NOP.

**Returns**

Terminate behavior.

**7.228.3.2 static OSCL\_IMPORT\_REF bool OsclThread::CompareId (TOsclThreadId & *t1*, TOsclThreadId & *t2*) [static]**

Static routine to compare whether two thread ID's are equal.

**Parameters**

*t1,t2*: thread ID passed by the application

**Returns**

true if equal.

**7.228.3.3 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Create (TOsclThreadFuncPtr *func*, int32 *stack\_size*, TOsclThreadFuncArg *argument*, OsclThread\_State *state* = Start\_on\_creation, bool *oIsJoinable* = false)**

This routine will create a thread. The thread may be launched immediately or may be created in a suspended state and launched with a Resume call.

**Parameters**

*func* = Name of the thread Function  
*stack\_size* = Size of the thread stack. If zero, then the platform-specific default stack size will be used.  
*argument* = Argument to be passed to thread function  
*state* = Enumeration which specifies the state of the thread on creation with values Running and Suspend. Note: the Suspend option may not be available on all platforms. If it is not supported, the Create call will return INVALID\_PARAM\_ERROR.  
*oIsJoinable* = A boolean, which when set to true, creates a Joinable thread. The default value for this is false, which creates a Detached thread.  
Note 1: When a joinable thread is created, it is imperative to call thread Terminate. Otherwise, it would cause a memory leak.  
Note 2: This is currently available only for platforms that have support for pthreads.

**Returns**

eOsclProcError

**7.228.3.4 static OSCL\_IMPORT\_REF void OsclThread::Exit (OsclAny \* *exitcode*) [static]**

Exit is a static function which is used to end the current thread. When called it just ends the execution of the current thread. Note: on some platforms this may be a NOP.

**Parameters**

*exitcode* = Exitcode of the thread. This can be used by other threads to know the exit status of this thread.

**Returns**

None

**7.228.3.5 static OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::GetId  
(TOsclThreadId & refThreadId) [static]**

Static routine to retrieve ID of calling thread.

**Parameters**

*Thread* ID passed by the application

**Returns**

Error code

**7.228.3.6 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::GetPriority  
(OsclThreadPriority & refThreadPriority)**

GetThreadPriority gets the priority of the thread. It takes reference of the input argument and assigns priority to it from one of the already defined priorities.

**Parameters**

*int16&* refThreadPriority : Output Priority value

**Returns**

Error code

**7.228.3.7 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Resume ()**

ResumeThread resumes the suspended thread and brings it into execution.

**Parameters**

*None*

**Returns**

Error code Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.228.3.8 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::SetPriority  
(OsclThreadPriority *ePriority*)**

SetThreadPriority sets the priority of the thread. It takes priority as the input argument and assigns it to the thread referred.

**Parameters**

*ePriorityLevel* : Input Priority value

**Returns**

Error code Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.228.3.9 static OSCL\_IMPORT\_REF void OsclThread::SleepMillisec (const int32 msec)  
[static]**

Suspend current thread execution for specified time.

**Parameters**

*msec*,*t2*,: sleep time in milliseconds.

**7.228.3.10 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Suspend ()**

This API suspends the thread being referred. The thread can later be brought into execution by calling OSCL\_ResumeThread() on it.

**Parameters**

*None*

**Returns**

Error code Note: this function may not be supported on all platforms, and may return NOT\_IMPLEMENTED.

**7.228.3.11 OSCL\_IMPORT\_REF OsclProcStatus::eOsclProcError OsclThread::Terminate  
(OsclAny \* *exitcode*)**

Terminate a thread other than the calling thread.

This API may have multiple behaviors. It may do a hard kill, a "join" operation, or a do-nothing. Caller can use CanTerminate option to tell the behavior in advance.

**Parameters**

*exitcode* = Exitcode of the thread.

**Returns**

Error code

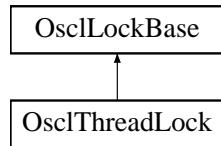
The documentation for this class was generated from the following file:

- [oscl\\_thread.h](#)

## 7.229 OsclThreadLock Class Reference

```
#include <oscl_mutex.h>
```

Inheritance diagram for OsclThreadLock:



### Public Member Functions

- OSCL\_IMPORT\_REF OsclThreadLock ()
- virtual OSCL\_IMPORT\_REF ~OsclThreadLock ()
- OSCL\_IMPORT\_REF void Lock ()
- OSCL\_IMPORT\_REF void Unlock ()

#### 7.229.1 Detailed Description

An implementation of [OsclLockBase](#) using a mutex

#### 7.229.2 Constructor & Destructor Documentation

**7.229.2.1 OSCL\_IMPORT\_REF OsclThreadLock::OsclThreadLock ()**

**7.229.2.2 virtual OSCL\_IMPORT\_REF OsclThreadLock::~OsclThreadLock () [virtual]**

#### 7.229.3 Member Function Documentation

**7.229.3.1 OSCL\_IMPORT\_REF void OsclThreadLock::Lock () [virtual]**

Implements [OsclLockBase](#).

**7.229.3.2 OSCL\_IMPORT\_REF void OsclThreadLock::Unlock () [virtual]**

Implements [OsclLockBase](#).

The documentation for this class was generated from the following file:

- [oscl\\_mutex.h](#)

## 7.230 OsclTickCount Class Reference

```
#include <oscl_tickcount.h>
```

### Static Public Member Functions

- static uint32 [TickCount \(\)](#)
- static uint32 [TickCountFrequency \(\)](#)
- static uint32 [TickCountPeriod \(\)](#)
- static uint32 [TicksToMsec \(uint32 ticks\)](#)
- static uint32 [MsecToTicks \(uint32 msec\)](#)

#### 7.230.1 Detailed Description

[OsclTickCount](#) class is used to retrieve the system tick count and the tick counter's frequency.

The maximum tick count value is equivalent to the maximum uint32 value.

#### 7.230.2 Member Function Documentation

##### 7.230.2.1 static uint32 OsclTickCount::MsecToTicks (uint32 *msec*) [static]

This function converts milliseconds to ticks

###### Returns

ticks

##### 7.230.2.2 static uint32 OsclTickCount::TickCount () [static]

This function returns the current system tick count

###### Returns

returns the tick count

Referenced by [OsclTimer< Alloc >::Request\(\)](#), and [OsclTimer< Alloc >::TimerBaseElapsed\(\)](#).

##### 7.230.2.3 static uint32 OsclTickCount::TickCountFrequency () [static]

This function returns the tick frequency in ticks per second

###### Returns

ticks per second

**7.230.2.4 static uint32 OsclTickCount::TickCountPeriod () [static]**

This function returns the tick period in microseconds per tick

**Returns**

microseconds per tick

Referenced by OsclTimer< Alloc >::SetExactFrequency(), and OsclTimer< Alloc >::SetFrequency().

**7.230.2.5 static uint32 OsclTickCount::TicksToMsec (uint32 *ticks*) [static]**

This function converts ticks to milliseconds

**Returns**

milliseconds

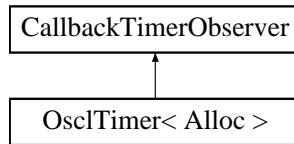
The documentation for this class was generated from the following file:

- [oscl\\_tickcount.h](#)

## 7.231 OsclTimer< Alloc > Class Template Reference

```
#include <oscl_timer.h>
```

Inheritance diagram for OsclTimer< Alloc >:



### Data Structures

- struct `_TimerEntry`

### Public Types

- typedef `CallbackTimer< Alloc > callback_timer_type`

### Public Member Functions

- `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 Member Functions

- void `TimerBaseElapsed ()`

### Friends

- class `CallbackTimer< Alloc >`

---

```
template<class Alloc> class OsclTimer< Alloc >
```

### 7.231.1 Member Typedef Documentation

7.231.1.1 **template<class Alloc > typedef CallbackTimer<Alloc> OsclTimer< Alloc >::callback\_timer\_type**

### 7.231.2 Constructor & Destructor Documentation

7.231.2.1 **template<class Alloc > OsclTimer< Alloc >::OsclTimer (const char \* *name*, uint32 *frequency* = 1, int32 *priority* = OsclActiveObject::EPriorityNominal) [inline]**

Constructor

#### Parameters

*frequency* The frequency of the timer in cycles/second. A value of 1 means the timer will cycle in 1 second intervals.

References OSCL\_LEAVE, OSCL\_PLACEMENT\_NEW, OsclErrArgument, and OsclTimer< Alloc >::SetFrequency().

7.231.2.2 **template<class Alloc > OsclTimer< Alloc >::~OsclTimer () [inline, virtual]**

References Oscl\_Vector< T, Alloc >::begin(), OsclTimerObject::Cancel(), Oscl\_TAlloc< T, Alloc >::deallocate(), Oscl\_Vector< T, Alloc >::end(), and NULL.

### 7.231.3 Member Function Documentation

7.231.3.1 **template<class Alloc > void OsclTimer< Alloc >::Cancel (int32 *timerID*, int32 *timeoutInfo* = -1) [inline]**

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.

References Oscl\_Vector< T, Alloc >::begin(), Oscl\_TAlloc< T, Alloc >::deallocate(), Oscl\_Vector< T, Alloc >::end(), Oscl\_Vector< T, Alloc >::erase(), and Oscl\_Vector< T, Alloc >::push\_back().

Referenced by OsclTimer< Alloc >::TimerBaseElapsed().

7.231.3.2 **template<class Alloc > void OsclTimer< Alloc >::Clear () [inline]**

Cancel all pending timers.

References Oscl\_Vector< T, Alloc >::begin(), Oscl\_Vector< T, Alloc >::clear(), Oscl\_TAlloc< T, Alloc >::deallocate(), and Oscl\_Vector< T, Alloc >::end().

**7.231.3.3 template<class Alloc > void OsclTimer< Alloc >::Request (int32 *timerID*, int32 *timeoutInfo*, int32 *cycles*, OsclTimerObserver \* *obs* = 0, bool *recurring* = 0) [inline]**

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.

References Oscl\_Vector< T, Alloc >::push\_back(), OsclTimerObject::RunIfNotReady(), and OsclTickCount::TickCount().

Referenced by OsclTimer< Alloc >::TimerBaseElapsed().

**7.231.3.4 template<class Alloc > void OsclTimer< Alloc >::SetExactFrequency (uint32 *frequency*) [inline]**

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.

References OsclTickCount::TickCountPeriod().

**7.231.3.5 template<class Alloc > void OsclTimer< Alloc >::SetFrequency (uint32 *frequency*) [inline]**

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.

References OsclTickCount::TickCountPeriod().

Referenced by OsclTimer< Alloc >::OsclTimer().

**7.231.3.6 template<class Alloc > void OsclTimer< Alloc >::SetObserver (OsclTimerObserver \* *obs*) [inline]**

Set the global observer. Each timer can request a local observer, which if set overrides the global observer.

#### Parameters

*obs* observer object.

**7.231.3.7 template<class Alloc > void OsclTimer< Alloc >::TimerBaseElapsed () [inline, protected, virtual]**

Implements [CallbackTimerObserver](#).

References `Oscl_Vector< T, Alloc >::begin()`, `OsclTimer< Alloc >::Cancel()`, `Oscl_Vector< T, Alloc >::clear()`, `Oscl_TAlloc< T, Alloc >::deallocate()`, `Oscl_Vector_Base::empty()`, `Oscl_Vector< T, Alloc >::end()`, `Oscl_Vector< T, Alloc >::erase()`, `OSCL_ABS`, `OSCL_MAX`, `OsclTimer< Alloc >::Request()`, `OsclTimerObject::RunIfNotReady()`, `OsclTickCount::TickCount()`, and `OsclTimerObserver::TimeoutOccurred()`.

## 7.231.4 Friends And Related Function Documentation

**7.231.4.1 template<class Alloc > friend class CallbackTimer< Alloc > [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_timer.h](#)

## 7.232 OsclTimerCompare Class Reference

```
#include <oscl_scheduler_readyq.h>
```

### Static Public Member Functions

- static int [compare \(TOsclReady &a, TOsclReady &b\)](#)

#### 7.232.1 Member Function Documentation

##### 7.232.1.1 static int OsclTimerCompare::compare (TOsclReady & a, TOsclReady & b) [static]

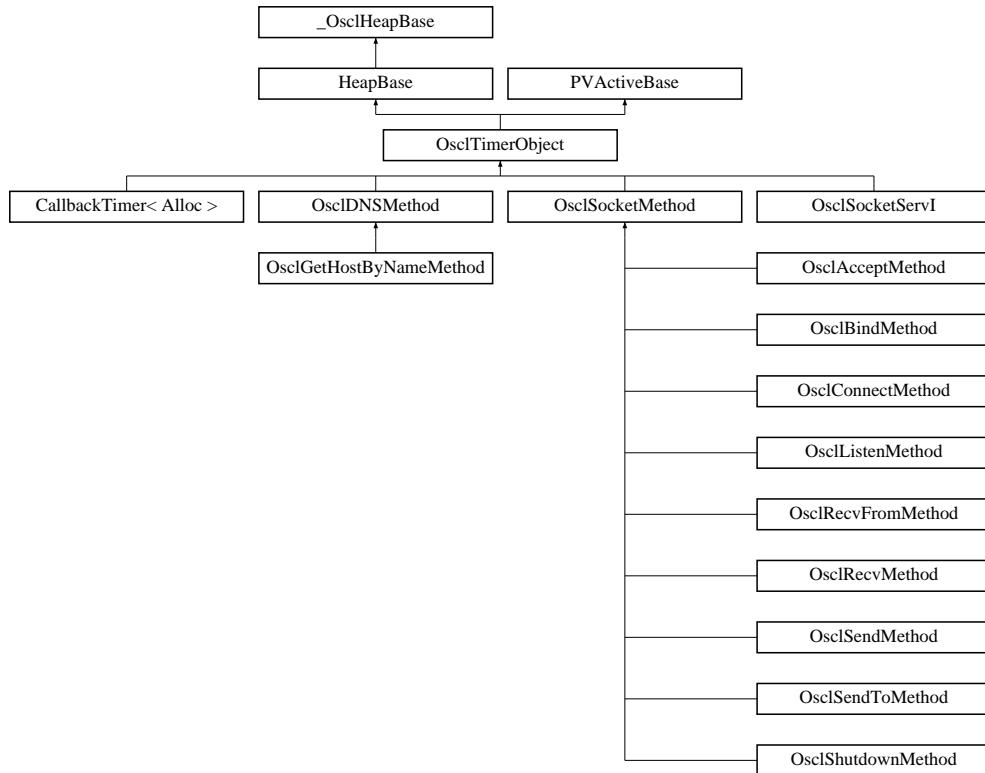
The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 7.233 OsclTimerObject Class Reference

```
#include <oscl_scheduler_ao.h>
```

Inheritance diagram for OsclTimerObject:



### Public Member Functions

- `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 Member Functions

- `virtual OSCL_IMPORT_REF void DoCancel ()`
- `virtual OSCL_IMPORT_REF int32 RunError (int32 aError)`

### 7.233.1 Detailed Description

User base class for execution objects. [OsclTimerObject](#) defines an exec object with a timer.

### 7.233.2 Constructor & Destructor Documentation

#### 7.233.2.1 OSCL\_IMPORT\_REF OsclTimerObject::OsclTimerObject (int32 *aPriority*, const char *name*[ ])

Constructor.

##### Parameters

*aPriority* (input param): scheduling priority

*name* (input param): optional name for this AO.

#### 7.233.2.2 virtual OSCL\_IMPORT\_REF OsclTimerObject::~OsclTimerObject () [virtual]

Destructor.

### 7.233.3 Member Function Documentation

#### 7.233.3.1 OSCL\_IMPORT\_REF void OsclTimerObject::AddToScheduler ()

Add this AO to the current thread's scheduler.

Reimplemented from [PVActiveBase](#).

Referenced by [CallbackTimer< Alloc >::CallbackTimer\(\)](#).

#### 7.233.3.2 OSCL\_IMPORT\_REF void OsclTimerObject::After (int32 *aDelayMicrosec*)

'After' sets the request ready, with request status OSCL\_REQUEST\_STATUS\_PENDING, and starts a timer. When the timer expires, the request will complete with status OSCL\_REQUEST\_ERR\_NONE. Must be called from the same thread in which the active object is scheduled. Will leave if the request is already readied, the object is not added to any scheduler, or the calling thread does not match the scheduling thread.

##### Parameters

*anInterval*,: timeout interval in microseconds.

#### 7.233.3.3 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](#).

Referenced by [OsclSocketMethod::Abort\(\)](#), and [OsclTimer< Alloc >::~OsclTimer\(\)](#).

---

**7.233.3.4 OSCL\_IMPORT\_REF void OsclTimerObject::DoCancel () [protected, virtual]**

Cancel request handler. This gets called by scheduler when the request is cancelled. The default routine will cancel the timer. If any additional action is needed, the derived class may override this. If the derived class does override this, it should explicitly call [OsclTimerObject::DoCancel](#) in its own DoCancel routine.

Implements [PVActiveBase](#).

**7.233.3.5 OSCL\_IMPORT\_REF bool OsclTimerObject::IsBusy () const**

Return true if this AO is active, false otherwise.

**7.233.3.6 OSCL\_IMPORT\_REF int32 OsclTimerObject::Priority () const**

Return scheduling priority of this exec object.

**7.233.3.7 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](#).

Referenced by [CallbackTimer< Alloc >::~CallbackTimer\(\)](#).

**7.233.3.8 virtual OSCL\_IMPORT\_REF int32 OsclTimerObject::RunError (int32 *aError*) [protected, virtual]**

Run Leave handler. This gets called by scheduler when the Run routine leaves. The default implementation simply returns the leave code. If the derived class wants to handle errors from Run, it may override this. The ExecError should return OsclErrNone if it handles the error, otherwise it should return the input error code.

#### Parameters

*aError*: the leave code generated by the Run.

Implements [PVActiveBase](#).

**7.233.3.9 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.

Referenced by [OsclTimer< Alloc >::Request\(\)](#), and [OsclTimer< Alloc >::TimerBaseElapsed\(\)](#).

**7.233.3.10 OSCL\_IMPORT\_REF void OsclTimerObject::SetBusy ()**

Set request ready for this AO. Will leave if the request is already readied, or the exec object is not added to any scheduler, or the calling thread context does not match the scheduler thread.

**7.233.3.11 OSCL\_IMPORT\_REF void OsclTimerObject::SetStatus (int32)****7.233.3.12 OSCL\_IMPORT\_REF int32 OsclTimerObject::Status () const**

Request status access

Referenced by `CallbackTimer< Alloc >::Run()`.

**7.233.3.13 OSCL\_IMPORT\_REF OsclAOStatus& OsclTimerObject::StatusRef ()**

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_ao.h](#)

## 7.234 OsclTimerObserver Class Reference

```
#include <oscl_timer.h>
```

### Public Member Functions

- virtual void [TimeoutOccurred](#) (int32 timerID, int32 timeoutInfo)=0
- virtual [~OsclTimerObserver](#) ()

#### 7.234.1 Detailed Description

The observer class to receive timeout callbacks

#### 7.234.2 Constructor & Destructor Documentation

7.234.2.1 virtual OsclTimerObserver::[~OsclTimerObserver](#) () [inline, virtual]

#### 7.234.3 Member Function Documentation

7.234.3.1 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.

Referenced by OsclTimer< Alloc >::TimerBaseElapsed().

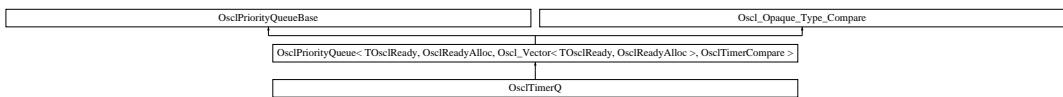
The documentation for this class was generated from the following file:

- [oscl\\_timer.h](#)

## 7.235 OsclTimerQ Class Reference

```
#include <oscl_scheduler_readyq.h>
```

Inheritance diagram for OsclTimerQ:



### Public Member Functions

- void [Construct](#) (int)
- void [Add](#) ([TOsclReady](#))
- void [Remove](#) ([TOsclReady](#))
- [TOsclReady](#) [PopTop](#) ()
- [TOsclReady](#) [Top](#) ()
- void [Pop](#) ([TOsclReady](#))
- bool [IsIn](#) ([TOsclReady](#))

#### 7.235.1 Member Function Documentation

**7.235.1.1 void OsclTimerQ::Add (TOsclReady)**

**7.235.1.2 void OsclTimerQ::Construct (int)**

**7.235.1.3 bool OsclTimerQ::IsIn (TOsclReady)**

**7.235.1.4 void OsclTimerQ::Pop (TOsclReady)**

**7.235.1.5 TOsclReady OsclTimerQ::PopTop ()**

**7.235.1.6 void OsclTimerQ::Remove (TOsclReady)**

**7.235.1.7 TOsclReady OsclTimerQ::Top ()**

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 7.236 OsclTLS< T, ID, Registry > Class Template Reference

```
#include <oscl_tls.h>
```

### Public Member Functions

- [OsclTLS \(\)](#)
- [~OsclTLS \(\)](#)
- [T & operator\\* \(\) const](#)

*The indirection operator (\*) accesses a value indirectly, through a pointer.*

- [T \\* operator-> \(\) const](#)

*The indirection operator (->) accesses a value indirectly, through a pointer.*

- [bool set \(\)](#)

*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- [T \\* \\_Ptr](#)

**template<class T, uint32 ID, class Registry = OsclTLSRegistry> class OsclTLS< T, ID, Registry >**

#### 7.236.1 Constructor & Destructor Documentation

**7.236.1.1 template<class T , uint32 ID, class Registry = OsclTLSRegistry> OsclTLS< T, ID, Registry >::OsclTLS () [inline]**

References OsclTLS< T, ID, Registry >::\_Ptr, and OSCL\_STATIC\_CAST.

**7.236.1.2 template<class T , uint32 ID, class Registry = OsclTLSRegistry> OsclTLS< T, ID, Registry >::~OsclTLS () [inline]**

#### 7.236.2 Member Function Documentation

**7.236.2.1 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.

References OsclTLS< T, ID, Registry >::\_Ptr.

**7.236.2.2 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.  
References [OsclTLS< T, ID, Registry >::\\_Ptr](#).

#### 7.236.2.3 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.

References [OsclTLS< T, ID, Registry >::\\_Ptr](#), and [OSCL\\_STATIC\\_CAST](#).

### 7.236.3 Field Documentation

#### 7.236.3.1 template<class T , uint32 ID, class Registry = OsclTLSRegistry> T\* OsclTLS< T, ID, Registry >::\_Ptr [protected]

Referenced by [OsclTLS< T, ID, Registry >::operator\\*\(\)](#), [OsclTLS< T, ID, Registry >::operator->\(\)](#), [OsclTLS< T, ID, Registry >::OsclTLS\(\)](#), and [OsclTLS< T, ID, Registry >::set\(\)](#).

The documentation for this class was generated from the following file:

- [oscl\\_tls.h](#)

## 7.237 OsclTLSEEx< T, ID, Registry > Class Template Reference

```
#include <oscl_error.h>
```

### Public Member Functions

- [OsclTLSEEx \(\)](#)
- [~OsclTLSEEx \(\)](#)
- [T & operator\\* \(\) const](#)

*The indirection operator (\*) accesses a value indirectly, through a pointer.*

- [T \\* operator-> \(\) const](#)

*The indirection operator (->) accesses a value indirectly, through a pointer.*

- [bool set \(\)](#)

*set() method sets ownership to the pointer, passed. This method is needed when the class is created with a default constructor. Returns false in case the class is non-empty.*

### Protected Attributes

- [T \\* \\_Ptr](#)

```
template<class T, uint32 ID, class Registry = OsclTLSRegistryEx> class OsclTLSEEx< T, ID, Registry >
```

#### 7.237.1 Constructor & Destructor Documentation

7.237.1.1 [template<class T , uint32 ID, class Registry = OsclTLSRegistryEx> OsclTLSEEx< T, ID, Registry >::OsclTLSEEx \(\) \[inline\]](#)

7.237.1.2 [template<class T , uint32 ID, class Registry = OsclTLSRegistryEx> OsclTLSEEx< T, ID, Registry >::~OsclTLSEEx \(\) \[inline\]](#)

#### 7.237.2 Member Function Documentation

7.237.2.1 [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.

References [OsclTLSEEx< T, ID, Registry >::\\_Ptr](#).

7.237.2.2 [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.

References OsclTLSEEx< T, ID, Registry >::\_Ptr.

#### 7.237.2.3 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.

References OsclTLSEEx< T, ID, Registry >::\_Ptr, and OSCL\_STATIC\_CAST.

### 7.237.3 Field Documentation

#### 7.237.3.1 template<class T , uint32 ID, class Registry = OsclTLSRegistryEx> T\* OsclTLSEEx< T, ID, Registry >::\_Ptr [protected]

Referenced by OsclTLSEEx< T, ID, Registry >::operator\*(), OsclTLSEEx< T, ID, Registry >::operator->(), and OsclTLSEEx< T, ID, Registry >::set().

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 7.238 OsclTLSRegistry Class Reference

```
#include <oscl_tls.h>
```

### Static Public Member Functions

- static OSCL\_IMPORT\_REF [OsclAny](#) \* [getInstance](#) (uint32 ID, int32 &error)
- static OSCL\_IMPORT\_REF void [registerInstance](#) ([OsclAny](#) \*ptr, uint32 ID, int32 &error)

### Friends

- class [OsclBase](#)

### 7.238.1 Member Function Documentation

**7.238.1.1 static OSCL\_IMPORT\_REF OsclAny\* OsclTLSRegistry::getInstance (uint32 *ID*, int32 &*error*) [static]**

**7.238.1.2 static OSCL\_IMPORT\_REF void OsclTLSRegistry::registerInstance (OsclAny \**ptr*, uint32 *ID*, int32 &*error*) [static]**

### 7.238.2 Friends And Related Function Documentation

**7.238.2.1 friend class OsclBase [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_tls.h](#)

## 7.239 OsclTLSRegistryEx Class Reference

```
#include <oscl_error.h>
```

### Static Public Member Functions

- static [OsclAny \\* getInstance \(uint32 ID\)](#)
- static void [registerInstance \(OsclAny \\*ptr, uint32 ID\)](#)

#### 7.239.1 Member Function Documentation

##### 7.239.1.1 static OsclAny\* OsclTLSRegistryEx::getInstance (uint32 ID) [inline, static]

References OsclError::Leave(), and OSCL\_ASSERT.

##### 7.239.1.2 static void OsclTLSRegistryEx::registerInstance (OsclAny \* ptr, uint32 ID) [inline, static]

References OsclError::Leave(), and OSCL\_ASSERT.

The documentation for this class was generated from the following file:

- [oscl\\_error.h](#)

## 7.240 OsclTrapItem Class Reference

```
#include <oscl_heapbase.h>
```

### Public Member Functions

- OSCL\_INLINE OsclTrapItem (OsclTrapOperation anOperation)
- OSCL\_INLINE OsclTrapItem (OsclTrapOperation anOperation, OsclAny \*aPtr)

### Friends

- class OsclTrapStackItem
- class OsclTrapStack

#### 7.240.1 Constructor & Destructor Documentation

7.240.1.1 OSCL\_INLINE OsclTrapItem::OsclTrapItem (OsclTrapOperation *anOperation*)

7.240.1.2 OSCL\_INLINE OsclTrapItem::OsclTrapItem (OsclTrapOperation *anOperation*, OsclAny \* *aPtr*)

#### 7.240.2 Friends And Related Function Documentation

7.240.2.1 friend class OsclTrapStack [friend]

7.240.2.2 friend class OsclTrapStackItem [friend]

The documentation for this class was generated from the following file:

- oscl\_heapbase.h

## 7.241 OsclTrapStack Class Reference

```
#include <oscl_error_trapcleanup.h>
```

### Friends

- class [OsclError](#)
- class [OsclErrorTrap](#)
- class [OsclErrorTrapImp](#)

### 7.241.1 Detailed Description

A common type for cleanup stack and trap mark stack. for internal use only.

### 7.241.2 Friends And Related Function Documentation

**7.241.2.1 friend class OsclError [friend]**

**7.241.2.2 friend class OsclErrorTrap [friend]**

**7.241.2.3 friend class OsclErrorTrapImp [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_error\\_trapcleanup.h](#)

## 7.242 OsclTrapStackItem Class Reference

```
#include <oscl_error_trapcleanup.h>
```

### Public Member Functions

- [OsclTrapStackItem \(\)](#)
- [OsclTrapStackItem \(\\_OsclHeapBase \\*aCBase\)](#)
- [OsclTrapStackItem \(OsclAny \\*aTAny\)](#)
- [OsclTrapStackItem \(OsclTrapItem aItem\)](#)

### Data Fields

- [\\_OsclHeapBase \\* iCBase](#)
- [OsclAny \\* iTAny](#)
- [OsclTrapOperation iTrapOperation](#)
- [OsclTrapStackItem \\* iNext](#)

#### 7.242.1 Detailed Description

Internal cleanup stack item type.

#### 7.242.2 Constructor & Destructor Documentation

##### 7.242.2.1 OsclTrapStackItem::OsclTrapStackItem () [inline]

##### 7.242.2.2 OsclTrapStackItem::OsclTrapStackItem (\_OsclHeapBase \* aCBase) [inline]

References iCBase, iNext, iTAny, iTrapOperation, and NULL.

##### 7.242.2.3 OsclTrapStackItem::OsclTrapStackItem (OsclAny \* aTAny) [inline]

References iCBase, iNext, iTAny, iTrapOperation, and NULL.

##### 7.242.2.4 OsclTrapStackItem::OsclTrapStackItem (OsclTrapItem aItem) [inline]

References iCBase, iNext, iTAny, iTrapOperation, and NULL.

#### 7.242.3 Field Documentation

##### 7.242.3.1 \_OsclHeapBase\* OsclTrapStackItem::iCBase

Referenced by OsclTrapStackItem().

##### 7.242.3.2 OsclTrapStackItem\* OsclTrapStackItem::iNext

Referenced by OsclTrapStackItem().

**7.242.3.3 OsclAny\* OsclTrapStackItem::iTAny**

Referenced by OsclTrapStackItem().

**7.242.3.4 OsclTrapOperation OsclTrapStackItem::iTrapOperation**

Referenced by OsclTrapStackItem().

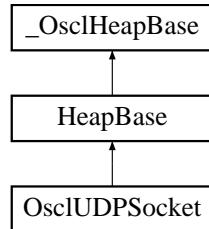
The documentation for this class was generated from the following file:

- [oscl\\_error\\_trapcleanup.h](#)

## 7.243 OsclUDPSocket Class Reference

```
#include <oscl_socket.h>
```

Inheritance diagram for OsclUDPSocket:



### Public Member Functions

- OSCL\_IMPORT\_REF ~OsclUDPSocket ()
- OSCL\_IMPORT\_REF TPVSocketEvent ThreadLogoff ()
- OSCL\_IMPORT\_REF TPVSocketEvent ThreadLogon (OsclSocketServ &aServ, OsclSocketObserver \*aObserver)
- OSCL\_IMPORT\_REF int32 Close ()
- OSCL\_IMPORT\_REF int32 Bind (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF int32 Join (OsclNetworkAddress &aAddress)
- OSCL\_IMPORT\_REF int32 JoinMulticastGroup (OsclIpMReq &aMReq)
- OSCL\_IMPORT\_REF int32 SetMulticastTTL (int32 aTTL)
- OSCL\_IMPORT\_REF int32 SetOptionToReuseAddress ()
- OSCL\_IMPORT\_REF int32 SetTOS (const OsclSocketTOS &aTOS)
- OSCL\_IMPORT\_REF int32 GetPeerName (OsclNetworkAddress &aPeerName)
- OSCL\_IMPORT\_REF TPVSocketEvent BindAsync (OsclNetworkAddress &aAddress, int32 aTimeoutMsec=(-1))
- OSCL\_IMPORT\_REF void CancelBind ()
- OSCL\_IMPORT\_REF uint8 \* GetRecvData (int32 \*aLength)
- OSCL\_IMPORT\_REF uint8 \* GetSendData (int32 \*aLength)
- OSCL\_IMPORT\_REF TPVSocketEvent SendTo (const uint8 \*aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1)
- OSCL\_IMPORT\_REF void CancelSendTo ()
- OSCL\_IMPORT\_REF TPVSocketEvent RecvFrom (uint8 \*aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiRecvLimit=0, Oscl\_Vector< uint32, OsclMemAllocator > \*aPacketLen=NULL, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \*aPacketSource=NULL)
- OSCL\_IMPORT\_REF void CancelRecvFrom ()
- OSCL\_IMPORT\_REF int32 SetRecvBufferSize (uint32 size)

### Static Public Member Functions

- static OSCL\_IMPORT\_REF OsclUDPSocket \* NewL (Oscl\_DefAlloc &alloc, OsclSocketServ &aServ, OsclSocketObserver \*aObserver, uint32 aId)

### 7.243.1 Detailed Description

The UDP Socket class

### 7.243.2 Member Function Documentation

**7.243.2.1 static 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.

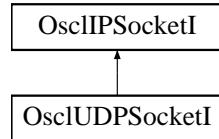
The documentation for this class was generated from the following file:

- [oscl\\_socket.h](#)

## 7.244 OsclUDPSocketI Class Reference

```
#include <oscl_udp_socket.h>
```

Inheritance diagram for OsclUDPSocketI:



### Public Member Functions

- virtual ~OsclUDPSocketI ()
- int32 [Close \(\)](#)
- int32 [JoinMulticastGroup \(OsclIpMReq &aMReq\)](#)
- int32 [SetMulticastTTL \(int32 aTTL\)](#)
- uint8 \* [GetRecvData \(int32 \\*aLength\)](#)
- uint8 \* [GetSendData \(int32 \\*aLength\)](#)
- [TPVSocketEvent ThreadLogoff \(\)](#)
- [TPVSocketEvent ThreadLogon \(OsclSocketServI \\*aServ, OsclSocketObserver \\*aObserver\)](#)
- [TPVSocketEvent BindAsync \(OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelBind \(\)](#)
- [TPVSocketEvent SendTo \(const uint8 \\*&aPtr, uint32 aLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1\)](#)
- void [CancelSendTo \(\)](#)
- [TPVSocketEvent RecvFrom \(uint8 \\*&aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, int32 aTimeoutMsec=-1, uint32 aMultiMaxLen=0, Oscl\\_Vector< uint32, OsclMemAllocator > \\*aPacketLen=NULL, Oscl\\_Vector< OsclNetworkAddress, OsclMemAllocator > \\*aPacketSource=NULL\)](#)
- void [CancelRecvFrom \(\)](#)

### Static Public Member Functions

- static OsclUDPSocketI \* [NewL \(Oscl\\_DefAlloc &a, OsclSocketServI \\*aServ, OsclSocketObserver \\*aObserver, uint32 aId\)](#)

#### 7.244.1 Detailed Description

Internal implementation class for [OsclUDPSocket](#)

## 7.244.2 Constructor & Destructor Documentation

7.244.2.1 **virtual OsclUDPSocketI::~OsclUDPSocketI () [virtual]**

## 7.244.3 Member Function Documentation

7.244.3.1 **TPVSocketEvent OsclUDPSocketI::BindAsync (OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]**

References OsclBindMethod::Bind(), EPVSocketFailure, EPVSocketNotImplemented, OsclSocketIBase::HasAsyncBind(), OsclIPSocketI::iAddress, OsclIPSocketI::iObserver, OsclNetworkAddress::ipAddr, OsclNetworkAddress::port, OsclNameString< \_\_len >::Set(), and OsclNameString< \_\_len >::Str().

7.244.3.2 **void OsclUDPSocketI::CancelBind () [inline]**

References OsclSocketMethod::CancelMethod().

7.244.3.3 **void OsclUDPSocketI::CancelRecvFrom () [inline]**

References OsclSocketMethod::CancelMethod().

7.244.3.4 **void OsclUDPSocketI::CancelSendTo () [inline]**

References OsclSocketMethod::CancelMethod().

7.244.3.5 **int32 OsclUDPSocketI::Close () [virtual]**

Implements [OsclIPSocketI](#).

7.244.3.6 **uint8 \* OsclUDPSocketI::GetRecvData (int32 \* aLength) [inline, virtual]**

Implements [OsclIPSocketI](#).

References OsclRecvFromMethod::GetRecvData().

7.244.3.7 **uint8 \* OsclUDPSocketI::GetSendData (int32 \* aLength) [inline, virtual]**

Implements [OsclIPSocketI](#).

References OsclSendToMethod::GetSendData().

**7.244.3.8 int32 OsclUDPSocketI::JoinMulticastGroup (OsclIpMReq & aMReq)**

**7.244.3.9 static OsclUDPSocketI\* OsclUDPSocketI::NewL (Oscl\_DefAlloc & a, OsclSocketServI \* aServ, OsclSocketObserver \* aObserver, uint32 aId) [static]**

**7.244.3.10 TPVSocketEvent OsclUDPSocketI::RecvFrom (uint8 \*& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1, uint32 aMultiMaxLen = 0, Oscl\_Vector< uint32, OsclMemAllocator > \* aPacketLen = NULL, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \* aPacketSource = NULL) [inline]**

References EPVSocketFailure, OsclIPSocketI::iObserver, and OsclRecvFromMethod::RecvFrom().

**7.244.3.11 TPVSocketEvent OsclUDPSocketI::SendTo (const uint8 \*& aPtr, uint32 aLen, OsclNetworkAddress & aAddress, int32 aTimeoutMsec = -1) [inline]**

References EPVSocketFailure, OsclIPSocketI::iObserver, and OsclSendToMethod::SendTo().

**7.244.3.12 int32 OsclUDPSocketI::SetMulticastTTL (int32 aTTL)**

**7.244.3.13 TPVSocketEvent OsclUDPSocketI::ThreadLogoff ()**

Reimplemented from [OsclIPSocketI](#).

**7.244.3.14 TPVSocketEvent OsclUDPSocketI::ThreadLogon (OsclSocketServI \* aServ, OsclSocketObserver \* aObserver)**

The documentation for this class was generated from the following file:

- [oscl\\_udp\\_socket.h](#)

## 7.245 OsclUuid Struct Reference

```
#include <oscl_uuid.h>
```

### Public Member Functions

- [OsclUuid \(\)](#)
- [OsclUuid \(uint32 l, uint16 w1, uint16 w2, uint8 b1, uint8 b2, uint8 b3, uint8 b4, uint8 b5, uint8 b6, uint8 b7, uint8 b8\)](#)
- [OsclUuid \(const OsclUuid &uuid\)](#)
- [OsclUuid & operator= \(const OsclUuid &src\)](#)
- [bool operator== \(const OsclUuid &src\) const](#)
- [bool operator!= \(const OsclUuid &src\) const](#)

### Data Fields

- [uint32 data1](#)
- [uint16 data2](#)
- [uint16 data3](#)
- [uint8 data4 \[BYTES\\_IN\\_UUID\\_ARRAY\]](#)

#### 7.245.1 Detailed Description

OSCL UUID structure used for unique identification of modules and interfaces.

#### 7.245.2 Constructor & Destructor Documentation

##### 7.245.2.1 OsclUuid::OsclUuid () [inline]

References oscl\_memset().

##### 7.245.2.2 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]

References data1, data2, data3, and data4.

##### 7.245.2.3 OsclUuid::OsclUuid (const OsclUuid & uuid) [inline]

References oscl\_memcpy().

#### 7.245.3 Member Function Documentation

##### 7.245.3.1 bool OsclUuid::operator!= (const OsclUuid & src) const [inline]

##### 7.245.3.2 OsclUuid& OsclUuid::operator= (const OsclUuid & src) [inline]

References oscl\_memcpy().

**7.245.3.3 bool OsclUuid::operator==(const OsclUuid & src) const [inline]**

References data1, data2, data3, and data4.

## 7.245.4 Field Documentation

### 7.245.4.1 uint32 OsclUuid::data1

Referenced by operator==( ), and OsclUuid( ).

### 7.245.4.2 uint16 OsclUuid::data2

Referenced by operator==( ), and OsclUuid( ).

### 7.245.4.3 uint16 OsclUuid::data3

Referenced by operator==( ), and OsclUuid( ).

### 7.245.4.4 uint8 OsclUuid::data4[BYTES\_IN\_UUID\_ARRAY]

Referenced by operator==( ), and OsclUuid( ).

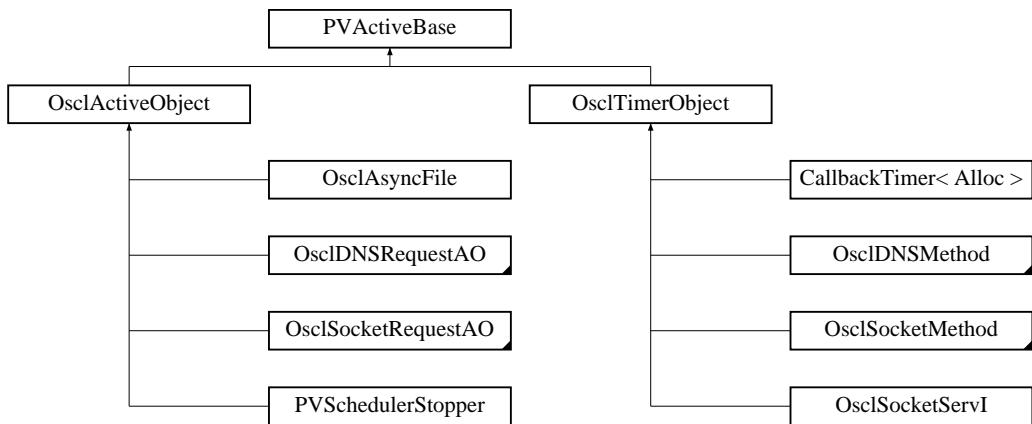
The documentation for this struct was generated from the following file:

- [oscl\\_uuid.h](#)

## 7.246 PVActiveBase Class Reference

```
#include <oscl_scheduler_aobase.h>
```

Inheritance diagram for PVActiveBase:



### Public Member Functions

- [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](#)
- [TReadyQueLink](#) [iPVReadyQLink](#)
- bool [iBusy](#)
- [OsclAOStatus](#) [iStatus](#)

### Friends

- class [OsclSchedulerCommonBase](#)
- class [OsclActiveObject](#)
- class [OsclTimerObject](#)

- class [OsclReadyQ](#)
- class [OsclReadyCompare](#)
- class [OsclReadySetPosition](#)
- class [OsclExecScheduler](#)

## 7.246.1 Detailed Description

PV Scheduler internal AO base class. Both [OsclActiveObject](#) and [OsclTimerObject](#) derive from this class. For Symbian, this just container has the desired additions to the basic CTimer or OsclActiveObj functionality. For non-Symbian, this class contains the entire AO implementation.

## 7.246.2 Constructor & Destructor Documentation

**7.246.2.1 `PVActiveBase::PVActiveBase (const char name[], int32 pri)`**

**7.246.2.2 `virtual PVActiveBase::~PVActiveBase () [virtual]`**

## 7.246.3 Member Function Documentation

**7.246.3.1 `void PVActiveBase::Activate ()`**

**7.246.3.2 `void PVActiveBase::AddToScheduler ()`**

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

**7.246.3.3 `void PVActiveBase::Cancel ()`**

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

**7.246.3.4 `void PVActiveBase::Destroy ()`**

**7.246.3.5 `virtual void PVActiveBase::DoCancel () [pure virtual]`**

Implements cancellation of an outstanding request.

This function is called as part of the active object's [Cancel\(\)](#).

It must call the appropriate cancel function offered by the active object's asynchronous service provider. The asynchronous service provider's cancel is expected to act immediately.

[DoCancel\(\)](#) must not wait for event completion; this is handled by [Cancel\(\)](#).

Implemented in [OsclDNSRequestAO](#), [OsclSocketRequestAO](#), [OsclActiveObject](#), and [OsclTimerObject](#).

**7.246.3.6 `OSCL_IMPORT_REF bool PVActiveBase::IsAdded () const`**

**7.246.3.7 `bool PVActiveBase::IsInAnyQ () [inline]`**

References TReadyQueLink::iIsIn, iPVReadyQLink, and NULL.

**7.246.3.8 void PVActiveBase::RemoveFromScheduler ()**

Reimplemented in [OsclActiveObject](#), and [OsclTimerObject](#).

**7.246.3.9 virtual void PVActiveBase::Run () [pure virtual]**

Handles an active object's request completion event.

A derived class must provide an implementation to handle the completed request. If appropriate, it may issue another request.

The function is called by the active scheduler when a request completion event occurs, i.e. after the active scheduler's `WaitForAnyRequest()` function completes.

Before calling this active object's `Run()` function, the active scheduler has:

1. decided that this is the highest priority active object with a completed request
2. marked this active object's request as complete (i.e. the request is no longer outstanding)

`Run()` runs under a trap harness in the active scheduler. If it leaves, then the active scheduler calls `ExecError()` to handle the leave.

Note that once the active scheduler's `Start()` function has been called, all user code is run under one of the program's active object's `Run()` or `RunError()` functions.

Implemented in [OsclDNSMethod](#), [OsclDNSRequestAO](#), [OsclSocketMethod](#), [OsclSocketRequestAO](#), and [CallbackTimer< Alloc >](#).

**7.246.3.10 virtual int32 PVActiveBase::RunError (int32 *aError*) [pure virtual]**

Virtual routine that gets called if the active object's Run routine leaves.

**Parameters**

*aError*,: the leave code generated by the Run.

**Returns**

:returns `OsclErrNone` if the error was handled, or returns the input *aError* value if not handled.

Implemented in [OsclActiveObject](#), and [OsclTimerObject](#).

## 7.246.4 Friends And Related Function Documentation

- 7.246.4.1 **friend class OsclActiveObject [friend]**
- 7.246.4.2 **friend class OsclExecScheduler [friend]**
- 7.246.4.3 **friend class OsclReadyCompare [friend]**
- 7.246.4.4 **friend class OsclReadyQ [friend]**
- 7.246.4.5 **friend class OsclReadySetPosition [friend]**
- 7.246.4.6 **friend class OsclSchedulerCommonBase [friend]**
- 7.246.4.7 **friend class OsclTimerObject [friend]**

## 7.246.5 Field Documentation

- 7.246.5.1 **uint32 PVActiveBase::iAddedNum**
- 7.246.5.2 **bool PVActiveBase::iBusy**
- 7.246.5.3 **OsclNameString<PVEXECNAMELEN> PVActiveBase::iName**
- 7.246.5.4 **TReadyQueLink PVActiveBase::iPVReadyQLink**

Referenced by IsInAnyQ().

### 7.246.5.5 OsclAOStatus PVActiveBase::iStatus

The request status associated with an asynchronous request.

This is passed as a parameter to all asynchronous service providers.

The active scheduler uses this to check whether the active object's request has completed.

The function can use the completion code to judge the success or otherwise of the request.

Request status contains one of the values OSCL\_REQUEST\_ERR\_NONE: request completed with no error, or request is not active. OSCL\_REQUEST\_PENDING: request is active & pending OSCL\_REQUEST\_ERR\_CANCEL: request was canceled before completion. or any user-defined value.

### 7.246.5.6 PVThreadContext PVActiveBase::iThreadContext

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_aobase.h](#)

## 7.247 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 Member Functions

- `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 Member Functions

- `static OSCL_IMPORT_REF void Init ()`
- `static OSCL_IMPORT_REF void Cleanup ()`
- `static OSCL_IMPORT_REF PVLogger * GetLoggerObject (const char *inputTag)`

### Protected Member Functions

- `void SetParent (PVLogger *parentLogger)`
- `PVLogger * GetParent ()`

### Friends

- class `PVLoggerRegistry`

### 7.247.1 Member Typedef Documentation

7.247.1.1 `typedef _OsclBasicAllocator PVLogger::alloc_type`

7.247.1.2 `typedef int32 PVLogger::filter_status_type`

7.247.1.3 `typedef int32 PVLogger::log_level_type`

7.247.1.4 `typedef int32 PVLogger::message_id_type`

### 7.247.2 Constructor & Destructor Documentation

7.247.2.1 `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

7.247.2.2 `virtual PVLogger::~PVLogger () [inline, virtual]`

References `Oscl_TAlloc< T, Alloc >::deallocate()`.

### 7.247.3 Member Function Documentation

7.247.3.1 `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

References `OSCL_UNUSED_ARG`, and `Oscl_Vector< T, Alloc >::push_back()`.

**7.247.3.2 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

References OSCL\_UNUSED\_ARG, and Oscl\_Vector< T, Alloc >::push\_back().

**7.247.3.3 static OSCL\_IMPORT\_REF void PVLogger::Cleanup () [static]**

Frees the [PVLogger](#) singleton used by the current thread. This must be called before thread exit. No messages can be logged after cleanup.

**Returns****7.247.3.4 void PVLogger::DisableAppenderInheritance () [inline]**

This method disables appender inheritance for the logging control point

**7.247.3.5 static 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

Referenced by OsclDNSMethod::OsclDNSMethod().

**7.247.3.6 `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`

**7.247.3.7 `uint32 PVLogger::GetNumAppenders () [inline]`**

This method returns the number of appenders attached to the logging control point.

References `Oscl_Vector_Base::size()`.

**7.247.3.8 `PVLogger* PVLogger::GetParent () [inline, protected]`****7.247.3.9 `static OSCL_IMPORT_REF void PVLogger::Init () [static]`**

`PVLogger` needs to be initialized once per thread. This creates the `PVLogger` singleton that is used throughout the duration of the thread. Initialization must occur before the first message is logged.

**Exceptions**

`leaves` if out of memory

**7.247.3.10 `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`

**7.247.3.11 `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

### 7.247.3.12 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

### 7.247.3.13 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

### 7.247.3.14 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

### 7.247.3.15 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

References Oscl\_Vector< T, Alloc >::begin(), Oscl\_Vector< T, Alloc >::end(), Oscl\_Vector< T, Alloc >::erase(), OsclSharedPtr< TheClass >::GetRep(), and OSCL\_UNUSED\_ARG.

### 7.247.3.16 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

References OSCL\_UNUSED\_ARG.

### 7.247.3.17 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

**7.247.3.18 void PVLogger::SetParent (PVLogger \**parentLogger*) [inline, protected]**

## 7.247.4 Friends And Related Function Documentation

### 7.247.4.1 friend class PVLoggerRegistry [friend]

The documentation for this class was generated from the following file:

- [pvlogger.h](#)

## 7.248 PVLoggerAppender Class Reference

```
#include <pvlogger_accessories.h>
```

### Public Types

- `typedef PVLogger::message_id_type message_id_type`

### Public Member Functions

- `virtual ~PVLoggerAppender ()`
- `virtual void AppendString (message_id_type msgID, const char *fmt, va_list va)=0`
- `virtual void AppendBuffers (message_id_type msgID, int32 numPairs, va_list va)=0`

#### 7.248.1 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.

#### 7.248.2 Member Typedef Documentation

##### 7.248.2.1 `typedef PVLogger::message_id_type PVLoggerAppender::message_id_type`

#### 7.248.3 Constructor & Destructor Documentation

##### 7.248.3.1 `virtual PVLoggerAppender::~PVLoggerAppender () [inline, virtual]`

#### 7.248.4 Member Function Documentation

##### 7.248.4.1 `virtual void PVLoggerAppender::AppendBuffers (message_id_type msgID, int32 numPairs, va_list va) [pure virtual]`

##### 7.248.4.2 `virtual void PVLoggerAppender::AppendString (message_id_type msgID, const char *fmt, va_list va) [pure virtual]`

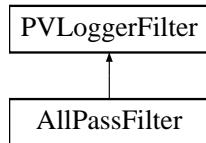
The documentation for this class was generated from the following file:

- `pvlogger_accessories.h`

## 7.249 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 Member Functions

- `virtual ~PVLoggerFilter ()`
- `virtual filter_status_type FilterString (char *tag, message_id_type msgID, log_level_type level)=0`
- `virtual filter_status_type FilterOpaqueMessage (char *tag, message_id_type msgID, log_level_type level)=0`

#### 7.249.1 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.

#### 7.249.2 Member Typedef Documentation

##### 7.249.2.1 `typedef PVLogger::filter_status_type PVLoggerFilter::filter_status_type`

Reimplemented in [AllPassFilter](#).

##### 7.249.2.2 `typedef PVLogger::log_level_type PVLoggerFilter::log_level_type`

Reimplemented in [AllPassFilter](#).

##### 7.249.2.3 `typedef PVLogger::message_id_type PVLoggerFilter::message_id_type`

Reimplemented in [AllPassFilter](#).

### 7.249.3 Constructor & Destructor Documentation

7.249.3.1 `virtual PVLoggerFilter::~PVLoggerFilter () [inline, virtual]`

### 7.249.4 Member Function Documentation

7.249.4.1 `virtual filter_status_type PVLoggerFilter::FilterOpaqueMessge (char * tag, message_id_type msgID, log_level_type level) [pure virtual]`

Implemented in [AllPassFilter](#).

7.249.4.2 `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](#)

## 7.250 PVLoggerLayout Class Reference

```
#include <pvlogger_accessories.h>
```

### Public Types

- `typedef PVLogger::message_id_type message_id_type`

### Public Member Functions

- `virtual ~PVLoggerLayout()`
- `virtual int32 FormatString (char *formatBuf, int32 formatBufSize, message_id_type msgID, const char *fmt, va_list va)=0`
- `virtual int32 FormatOpaqueMessage (char *formatBuf, int32 formatBufSize, message_id_type msgID, int32 numPairs, va_list va)=0`

#### 7.250.1 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.

#### 7.250.2 Member Typedef Documentation

##### 7.250.2.1 `typedef PVLogger::message_id_type PVLoggerLayout::message_id_type`

#### 7.250.3 Constructor & Destructor Documentation

##### 7.250.3.1 `virtual PVLoggerLayout::~PVLoggerLayout () [inline, virtual]`

#### 7.250.4 Member Function Documentation

##### 7.250.4.1 `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.

##### 7.250.4.2 `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 ”

The documentation for this class was generated from the following file:

- [pvlogger\\_accessories.h](#)

## 7.251 PVLoggerRegistry Class Reference

```
#include <pvlogger_registry.h>
```

### Public Types

- `typedef PVLogger::log_level_type log_level_type`
- `typedef PVLogger::alloc_type alloc_type`

### Public Member Functions

- `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 Member Functions

- `static OSCL_IMPORT_REF PVLoggerRegistry * GetPVLoggerRegistry()`

#### 7.251.1 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.

#### 7.251.2 Member Typedef Documentation

##### 7.251.2.1 `typedef PVLogger::alloc_type PVLoggerRegistry::alloc_type`

##### 7.251.2.2 `typedef PVLogger::log_level_type PVLoggerRegistry::log_level_type`

#### 7.251.3 Constructor & Destructor Documentation

##### 7.251.3.1 `OSCL_IMPORT_REF PVLoggerRegistry::PVLoggerRegistry()`

`PVLoggerRegistry` Constructor

##### 7.251.3.2 `virtual OSCL_IMPORT_REF PVLoggerRegistry::~PVLoggerRegistry () [virtual]`

`PVLoggerRegistry` Destructor

## 7.251.4 Member Function Documentation

### 7.251.4.1 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

### 7.251.4.2 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

### 7.251.4.3 static OSCL\_IMPORT\_REF PVLoggerRegistry\* PVLoggerRegistry::GetPVLoggerRegistry () [static]

Get the logger registry. There is only one logger registry instance per thread.

### 7.251.4.4 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

**7.251.4.5 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.

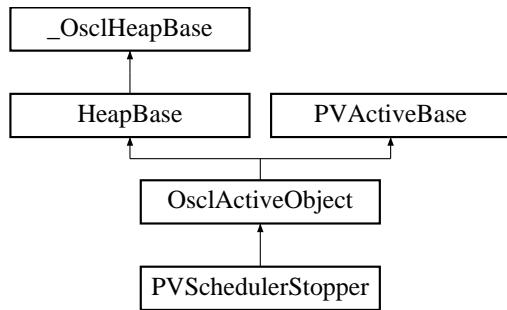
The documentation for this class was generated from the following file:

- [pvlogger\\_registry.h](#)

## 7.252 PVSchedulerStopper Class Reference

```
#include <oscl_scheduler.h>
```

Inheritance diagram for PVSchedulerStopper:



### Public Member Functions

- [PVSchedulerStopper \(\)](#)
- [~PVSchedulerStopper \(\)](#)

#### 7.252.1 Detailed Description

Scheduler stopper AO class, for internal use by scheduler.

#### 7.252.2 Constructor & Destructor Documentation

##### 7.252.2.1 PVSchedulerStopper::PVSchedulerStopper ()

##### 7.252.2.2 PVSchedulerStopper::~PVSchedulerStopper ()

The documentation for this class was generated from the following file:

- [oscl\\_scheduler.h](#)

## 7.253 PVSockBufRecv Class Reference

```
#include <oscl_socket_request.h>
```

### Public Member Functions

- [PVSockBufRecv \(\)](#)
- [PVSockBufRecv \(uint8 \\*aPtr, uint32 aLen, uint32 aMax\)](#)
- [PVSockBufRecv \(const PVSockBufRecv &a\)](#)

### Data Fields

- [uint8 \\* iPtr](#)
- [uint32 iLen](#)
- [uint32 iMaxLen](#)

#### 7.253.1 Constructor & Destructor Documentation

7.253.1.1 [PVSockBufRecv::PVSockBufRecv \(\) \[inline\]](#)

7.253.1.2 [PVSockBufRecv::PVSockBufRecv \(uint8 \\* aPtr, uint32 aLen, uint32 aMax\) \[inline\]](#)

7.253.1.3 [PVSockBufRecv::PVSockBufRecv \(const PVSockBufRecv & a\) \[inline\]](#)

#### 7.253.2 Field Documentation

7.253.2.1 [uint32 PVSockBufRecv::iLen](#)

7.253.2.2 [uint32 PVSockBufRecv::iMaxLen](#)

7.253.2.3 [uint8\\* PVSockBufRecv::iPtr](#)

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 7.254 PVSockBufSend Class Reference

```
#include <oscl_socket_request.h>
```

### Public Member Functions

- [PVSockBufSend \(\)](#)
- [PVSockBufSend \(const uint8 \\*aPtr, uint32 aLen\)](#)
- [PVSockBufSend \(const PVSockBufSend &a\)](#)

### Data Fields

- [const uint8 \\* iPtr](#)
- [uint32 iLen](#)

#### 7.254.1 Constructor & Destructor Documentation

7.254.1.1 [PVSockBufSend::PVSockBufSend \(\) \[inline\]](#)

7.254.1.2 [PVSockBufSend::PVSockBufSend \(const uint8 \\* aPtr, uint32 aLen\) \[inline\]](#)

7.254.1.3 [PVSockBufSend::PVSockBufSend \(const PVSockBufSend & a\) \[inline\]](#)

#### 7.254.2 Field Documentation

7.254.2.1 [uint32 PVSockBufSend::iLen](#)

7.254.2.2 [const uint8\\* PVSockBufSend::iPtr](#)

The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 7.255 PVThreadContext Class Reference

```
#include <oscl_scheduler_threadcontext.h>
```

### Public Member Functions

- 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 Member Functions

- static OSCL\_IMPORT\_REF uint32 Id()
- static OSCL\_IMPORT\_REF bool ThreadHasScheduler()

### Friends

- class PVActiveBase
- class OsclActiveObject
- class OsclTimerObject
- class OsclExecScheduler
- class OsclCoeActiveScheduler
- class OsclExecSchedulerCommonBase
- class OsclExecSchedulerBase
- class OsclCoeActiveSchedulerBase

#### 7.255.1 Constructor & Destructor Documentation

**7.255.1.1 OSCL\_IMPORT\_REF PVThreadContext::PVThreadContext()**

**7.255.1.2 OSCL\_IMPORT\_REF PVThreadContext::~PVThreadContext()**

#### 7.255.2 Member Function Documentation

**7.255.2.1 OSCL\_IMPORT\_REF void PVThreadContext::EnterThreadContext()**

enter and exit thread context.

**7.255.2.2 OSCL\_IMPORT\_REF void PVThreadContext::ExitThreadContext()**

**7.255.2.3 static OSCL\_IMPORT\_REF uint32 PVThreadContext::Id () [static]**

static routine to get a unique thread ID for caller's thread context.

**7.255.2.4 OSCL\_IMPORT\_REF bool PVThreadContext::IsSameThreadContext ()**

compare caller's thread context to this one.

**7.255.2.5 static OSCL\_IMPORT\_REF bool PVThreadContext::ThreadHasScheduler ()  
[static]**

a static utility to tell whether the calling thread has any scheduler-- either Oscl scheduler or native scheduler.

**7.255.3 Friends And Related Function Documentation****7.255.3.1 friend class OsclActiveObject [friend]****7.255.3.2 friend class OsclCoeActiveScheduler [friend]****7.255.3.3 friend class OsclCoeActiveSchedulerBase [friend]****7.255.3.4 friend class OsclExecScheduler [friend]****7.255.3.5 friend class OsclExecSchedulerBase [friend]****7.255.3.6 friend class OsclExecSchedulerCommonBase [friend]****7.255.3.7 friend class OsclTimerObject [friend]****7.255.3.8 friend class PVActiveBase [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_threadcontext.h](#)

## 7.256 Oscl\_TAlloc< T, Alloc >::rebind< U, V > Struct Template Reference

```
#include <oscl_defalloc.h>
```

### Public Types

- **typedef Oscl\_TAlloc< U, V > other**

```
template<class T, class Alloc>template<class U, class V> struct Oscl_TAlloc< T, Alloc >::rebind< U, V >
```

#### 7.256.1 Member Typedef Documentation

```
7.256.1.1 template<class T, class Alloc> template<class U , class V > typedef Oscl_TAlloc<U, V> Oscl_TAlloc< T, Alloc >::rebind< U, V >::other
```

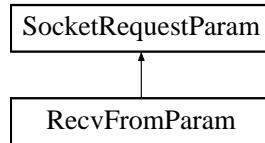
The documentation for this struct was generated from the following file:

- [oscl\\_defalloc.h](#)

## 7.257 RecvFromParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for RecvFromParam:



### Public Member Functions

- `RecvFromParam (uint8 *aPtr, uint32 aMaxLen, OsclNetworkAddress &aAddress, uint32 flags, uint32 aMultiMax, Oscl_Vector< uint32, OsclMemAllocator > *aPacketLen, Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > *aPacketSource)`

### Data Fields

- `PVSockBufRecv iBufRecv`
- `uint32 iFlags`
- `OsclNetworkAddress & iAddr`
- `uint32 iMultiMaxLen`
- `Oscl_Vector< uint32, OsclMemAllocator > * iPacketLen`
- `Oscl_Vector< OsclNetworkAddress, OsclMemAllocator > * iPacketSource`

#### 7.257.1 Constructor & Destructor Documentation

**7.257.1.1 RecvFromParam::RecvFromParam (uint8 \*& aPtr, uint32 aMaxLen, OsclNetworkAddress & aAddress, uint32 flags, uint32 aMultiMax, Oscl\_Vector< uint32, OsclMemAllocator > \* aPacketLen, Oscl\_Vector< OsclNetworkAddress, OsclMemAllocator > \* aPacketSource) [inline]**

#### 7.257.2 Field Documentation

**7.257.2.1 OsclNetworkAddress& RecvFromParam::iAddr**

**7.257.2.2 PVSockBufRecv RecvFromParam::iBufRecv**

**7.257.2.3 uint32 RecvFromParam::iFlags**

**7.257.2.4 uint32 RecvFromParam::iMultiMaxLen**

**7.257.2.5 Oscl\_Vector<uint32, OsclMemAllocator>\* RecvFromParam::iPacketLen**

**7.257.2.6 Oscl\_Vector<OsclNetworkAddress, OsclMemAllocator>\* RecvFromParam::iPacketSource**

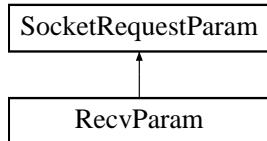
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 7.258 RecvParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for RecvParam:



### Public Member Functions

- [RecvParam \(uint8 \\*&aPtr, uint32 aMaxLen, uint32 flags\)](#)

### Data Fields

- [PVSockBufRecv iBufRecv](#)
- [uint32 iFlags](#)

#### 7.258.1 Constructor & Destructor Documentation

7.258.1.1 [RecvParam::RecvParam \(uint8 \\*& aPtr, uint32 aMaxLen, uint32 flags\) \[inline\]](#)

#### 7.258.2 Field Documentation

7.258.2.1 [PVSockBufRecv RecvParam::iBufRecv](#)

7.258.2.2 [uint32 RecvParam::iFlags](#)

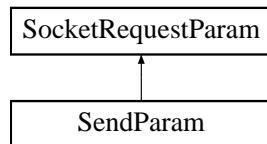
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 7.259 SendParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SendParam:



### Public Member Functions

- [SendParam \(const uint8 \\*aPtr, uint32 aLen, uint32 aFlags\)](#)

### Data Fields

- [PVSockBufSend iBufSend](#)
- [uint32 iFlags](#)
- [uint32 iXferLen](#)

#### 7.259.1 Detailed Description

Socket method parameter sets

#### 7.259.2 Constructor & Destructor Documentation

7.259.2.1 [SendParam::SendParam \(const uint8 \\*aPtr, uint32 aLen, uint32 aFlags\) \[inline\]](#)

#### 7.259.3 Field Documentation

7.259.3.1 [PVSockBufSend SendParam::iBufSend](#)

7.259.3.2 [uint32 SendParam::iFlags](#)

7.259.3.3 [uint32 SendParam::iXferLen](#)

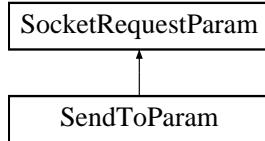
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 7.260 SendToParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SendToParam:



### Public Member Functions

- [SendToParam \(const uint8 \\*&aPtr, uint32 aLen, OsclNetworkAddress &anAddr, uint32 flags\)](#)
- [~SendToParam \(\)](#)

### Data Fields

- [PVSockBufSend iBufSend](#)
- [uint32 iFlags](#)
- [OsclNetworkAddress iAddr](#)
- [uint32 iXferLen](#)

#### 7.260.1 Constructor & Destructor Documentation

**7.260.1.1 SendToParam::SendToParam (const uint8 \*& aPtr, uint32 aLen, OsclNetworkAddress & anAddr, uint32 flags) [inline]**

**7.260.1.2 SendToParam::~SendToParam () [inline]**

#### 7.260.2 Field Documentation

**7.260.2.1 OsclNetworkAddress SendToParam::iAddr**

**7.260.2.2 PVSockBufSend SendToParam::iBufSend**

**7.260.2.3 uint32 SendToParam::iFlags**

**7.260.2.4 uint32 SendToParam::iXferLen**

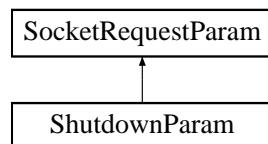
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 7.261 ShutdownParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for ShutdownParam:



### Public Member Functions

- [ShutdownParam \(TPVSocketShutdown aHow\)](#)

### Data Fields

- [TPVSocketShutdown iHow](#)

#### 7.261.1 Constructor & Destructor Documentation

7.261.1.1 [ShutdownParam::ShutdownParam \(TPVSocketShutdown \*aHow\*\) \[inline\]](#)

#### 7.261.2 Field Documentation

7.261.2.1 [TPVSocketShutdown ShutdownParam::iHow](#)

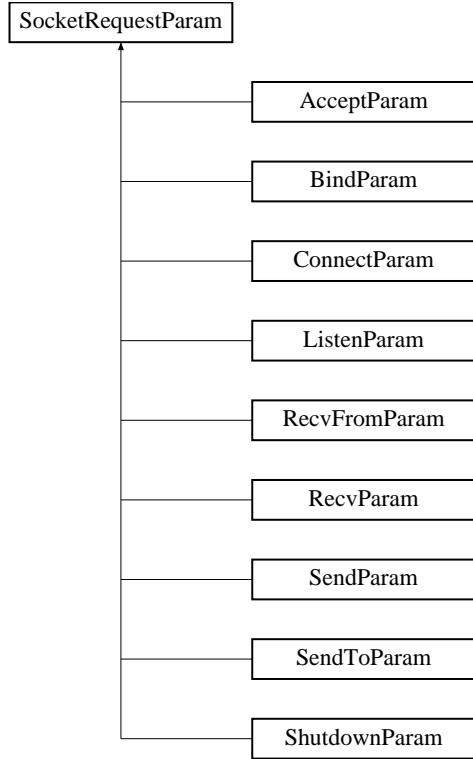
The documentation for this class was generated from the following file:

- [oscl\\_socket\\_request.h](#)

## 7.262 SocketRequestParam Class Reference

```
#include <oscl_socket_request.h>
```

Inheritance diagram for SocketRequestParam:



### Public Member Functions

- [SocketRequestParam \(TPVSocketFxn aFxn\)](#)

### Data Fields

- [TPVSocketFxn iFxn](#)

### 7.262.1 Detailed Description

Base class for all socket method parameter sets

## 7.262.2 Constructor & Destructor Documentation

### 7.262.2.1 SocketRequestParam::SocketRequestParam (TPVSocketFxn *aFxn*) [inline]

## 7.262.3 Field Documentation

### 7.262.3.1 TPVSocketFxn SocketRequestParam::iFxn

The documentation for this class was generated from the following file:

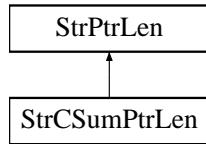
- [oscl\\_socket\\_request.h](#)

## 7.263 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 Member Functions

- `void setPtrLen (const char *newPtr, uint32 newLen)`
- `CheckSumType getCheckSum () const`
- `OSCL_IMPORT_REF void setCheckSum ()`
- `StrCSumPtrLen ()`
- `StrCSumPtrLen (const char *newPtr)`
- `StrCSumPtrLen (const char *newPtr, uint32 newLen)`
- `StrCSumPtrLen (const StrCSumPtrLen &rhs)`
- `StrCSumPtrLen (const StrPtrLen &rhs)`
- `c_bool isCIEquivalentTo (const StrCSumPtrLen &rhs) const`
- `c_bool operator== (const StrCSumPtrLen &rhs) const`
- `c_bool operator!= (const StrCSumPtrLen &rhs) const`
- `StrCSumPtrLen & operator= (const StrCSumPtrLen &rhs)`
- `StrCSumPtrLen & operator= (const StrPtrLen &rhs)`
- `StrCSumPtrLen & operator= (const char *rhs)`

### Protected Attributes

- `CheckSumType checkSum`

#### 7.263.1 Detailed Description

same as [StrPtrLen](#), but includes checksum field and method to speed up querying

### 7.263.2 Member Typedef Documentation

7.263.2.1 `typedef int16 StrCSumPtrLen::CheckSumType`

### 7.263.3 Constructor & Destructor Documentation

7.263.3.1 `StrCSumPtrLen::StrCSumPtrLen () [inline]`

7.263.3.2 `StrCSumPtrLen::StrCSumPtrLen (const char * newPtr) [inline]`

References `setCheckSum()`.

7.263.3.3 `StrCSumPtrLen::StrCSumPtrLen (const char * newPtr, uint32 newLen) [inline]`

References `setCheckSum()`.

7.263.3.4 `StrCSumPtrLen::StrCSumPtrLen (const StrCSumPtrLen & rhs) [inline]`

7.263.3.5 `StrCSumPtrLen::StrCSumPtrLen (const StrPtrLen & rhs) [inline]`

References `setCheckSum()`.

### 7.263.4 Member Function Documentation

7.263.4.1 `CheckSumType StrCSumPtrLen::getCheckSum () const [inline]`

References `checkSum`.

Referenced by `isCIEquivalentTo()`, and `operator==()`.

7.263.4.2 `c_bool StrCSumPtrLen::isCIEquivalentTo (const StrCSumPtrLen & rhs) const [inline]`

References `getCheckSum()`.

7.263.4.3 `c_bool StrCSumPtrLen::operator!= (const StrCSumPtrLen & rhs) const [inline]`

7.263.4.4 `StrCSumPtrLen& StrCSumPtrLen::operator= (const char * rhs) [inline]`

Reimplemented from [StrPtrLen](#).

References `operator=()`, and `setCheckSum()`.

7.263.4.5 `StrCSumPtrLen& StrCSumPtrLen::operator= (const StrPtrLen & rhs) [inline]`

Reimplemented from [StrPtrLen](#).

References `operator=()`, and `setCheckSum()`.

**7.263.4.6 StrCSumPtrLen& StrCSumPtrLen::operator= (const StrCSumPtrLen & *rhs*) [inline]**

References checkSum.

Referenced by operator=().

**7.263.4.7 c\_bool StrCSumPtrLen::operator== (const StrCSumPtrLen & *rhs*) const [inline]**

References getCheckSum().

**7.263.4.8 OSCL\_IMPORT\_REF void StrCSumPtrLen::setCheckSum ()**

Referenced by operator=(), setPtrLen(), and StrCSumPtrLen().

**7.263.4.9 void StrCSumPtrLen::setPtrLen (const char \* *newPtr*, uint32 *newLen*) [inline]**

Reimplemented from [StrPtrLen](#).

References setCheckSum().

## 7.263.5 Field Documentation

**7.263.5.1 CheckSumType StrCSumPtrLen::checkSum [protected]**

Referenced by getCheckSum(), and operator=().

The documentation for this struct was generated from the following file:

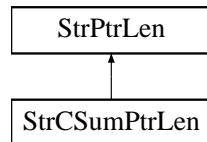
- [oscl\\_str\\_ptr\\_len.h](#)

## 7.264 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 Member Functions

- [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 Member Functions

- [bool isLetter \(const char c\) const](#)

### Protected Attributes

- [const char \\* ptr](#)
- [int32 len](#)

#### 7.264.1 Detailed Description

This data structure encapsulates a set of functions used to perform standard string operations. It should be used for null-terminated constant (non-modifiable) strings of char type.

#### 7.264.2 Constructor & Destructor Documentation

##### 7.264.2.1 StrPtrLen::StrPtrLen (const char \* *newPtr*) [inline]

References len, and oscl\_strlen().

**7.264.2.2 StrPtrLen::StrPtrLen (const char \* *newPtr*, uint32 *newLen*) [inline]**

**7.264.2.3 StrPtrLen::StrPtrLen () [inline]**

**7.264.2.4 StrPtrLen::StrPtrLen (const StrPtrLen & *rhs*) [inline]**

### **7.264.3 Member Function Documentation**

**7.264.3.1 const char\* StrPtrLen::c\_str () const [inline]**

References ptr.

**7.264.3.2 c\_bool StrPtrLen::isCIEquivalentTo (const StrPtrLen & *rhs*) const [inline]**

References isCIPrefixOf(), and len.

**7.264.3.3 c\_bool StrPtrLen::isCIPrefixOf (const StrPtrLen & *rhs*) const [inline]**

References isLetter(), len, OSCL\_ASCII\_CASE\_MAGIC\_BIT, and ptr.

Referenced by isCIEquivalentTo().

**7.264.3.4 bool StrPtrLen::isLetter (const char *c*) const [inline, protected]**

Referenced by isCIPrefixOf().

**7.264.3.5 int32 StrPtrLen::length () const [inline]**

References len.

**7.264.3.6 int32 StrPtrLen::operator!= (const StrPtrLen & *rhs*) const [inline]**

**7.264.3.7 StrPtrLen& StrPtrLen::operator= (const char \* *rhs*) [inline]**

Reimplemented in [StrCSumPtrLen](#).

References len, oscl\_strlen(), and ptr.

**7.264.3.8 StrPtrLen& StrPtrLen::operator= (const StrPtrLen & *rhs*) [inline]**

Reimplemented in [StrCSumPtrLen](#).

References len, and ptr.

**7.264.3.9 int32 StrPtrLen::operator== (const StrPtrLen & *rhs*) const [inline]**

References len, oscl\_strcmp(), and ptr.

**7.264.3.10 void StrPtrLen::setPtrLen (const char \* *newPtr*, uint32 *newLen*) [inline]**

Reimplemented in [StrCSumPtrLen](#).

References len, and ptr.

**7.264.3.11 int32 StrPtrLen::size () const [inline]**

References len.

## 7.264.4 Field Documentation

**7.264.4.1 int32 StrPtrLen::len [protected]**

Referenced by [isCIEquivalentTo\(\)](#), [isCIPrefixOf\(\)](#), [length\(\)](#), [operator=\(\)](#), [operator==\(\)](#), [setPtrLen\(\)](#), [size\(\)](#), and [StrPtrLen\(\)](#).

**7.264.4.2 const char\* StrPtrLen::ptr [protected]**

Referenced by [c\\_str\(\)](#), [isCIPrefixOf\(\)](#), [operator=\(\)](#), [operator==\(\)](#), and [setPtrLen\(\)](#).

The documentation for this struct was generated from the following file:

- [oscl\\_str\\_ptr\\_len.h](#)

## 7.265 TimeValue Class Reference

The [TimeValue](#) class represents a time value in a format native to the system.

```
#include <oscl_time.h>
```

### Public Member Functions

- **OSCL\_COND\_IMPORT\_REF TimeValue ()**  
*Create a [TimeValue](#) representing the current time.*
- **OSCL\_COND\_IMPORT\_REF TimeValue (const TimeValue &Tv)**  
*Copy constructor.*
- **OSCL\_COND\_IMPORT\_REF TimeValue (long tv, TimeUnits units)**  
*Create a [TimeValue](#) representing an interval of tv units.*
- **OSCL\_COND\_IMPORT\_REF TimeValue (const OsclBasicTimeStruct &in\_tv)**  
*Create a [TimeValue](#) representing the absolute time specified by the BasicTimeStruct.*
- **OSCL\_COND\_IMPORT\_REF TimeValue (const ISO8601timeStrBuf time\_strbuf)**
- **OSCL\_COND\_IMPORT\_REF TimeValue (uint16 aYear, uint16 aMonth, uint16 aDay, uint16 aHour, uint16 aMinute, uint16 aSecond, uint16 aMilliseconds)**
- **OSCL\_COND\_IMPORT\_REF TimeValue (OsclBasicDateTimeStruct in\_ts)**  
*Create a [TimeValue](#) representing the absolute time specified by the BasicDateTimeStruct.*
- **OSCL\_COND\_IMPORT\_REF int32 get\_local\_time ()**  
*Get the local time after having adjusted for daylight saving.*
- **OSCL\_COND\_IMPORT\_REF void set\_to\_zero ()**  
*Set the time value to zero (represents a zero interval).*
- **OSCL\_COND\_IMPORT\_REF void set\_to\_current\_time ()**  
*Set the time value to the current system time.*
- **OSCL\_COND\_IMPORT\_REF void set\_from\_ntp\_time (const uint32 ntp\_offset)**  
*This method converts a 32-bit NTP offset to system time.*
- **OSCL\_COND\_IMPORT\_REF uint32 get\_sec () const**  
*Get a 32 bit value representing the seconds since the (system dependent) epoch.*
- **OSCL\_COND\_IMPORT\_REF int32 to\_msec () const**
- **OSCL\_COND\_IMPORT\_REF uint32 get\_usec () const**  
*Get a 32 bit value representing the number of microseconds in the time value.*
- **OSCL\_COND\_IMPORT\_REF uint64 get\_timevalue\_in\_usec () const**  
*Get a 64 bit value representing the time value converted to microseconds.*
- **OSCL\_IMPORT\_REF char \* get\_str\_ctime (CtimeStrBuf ctime\_strbuf)**  
*Get a string containing a text representation of this [TimeValue](#) object.*

- OSCL\_IMPORT\_REF int [get\\_pv8601\\_str\\_time](#) (PV8601timeStrBuf time\_strbuf)  
*Get a PV extended text representation of the Time based on the PV 8601 format.*
- OSCL\_IMPORT\_REF int [get\\_ISO8601\\_str\\_time](#) (ISO8601timeStrBuf time\_strbuf)  
*Get a PV extended text representation of the Time based on the ISO 8601 format.*
- OSCL\_IMPORT\_REF char \* [get\\_rfc822\\_gmtime\\_str](#) (int max\_time\_strlen, char \*time\_str)  
*Get a text representation of the time in the GMT timezone based on the RFC 822 / RFC 1123 (also described in the HTTP spec RFC 2068 and RFC 2616).*
- OSCL\_COND\_IMPORT\_REF bool [is\\_zero](#) ()  
*Determine if the time value is zero.*
- OSCL\_COND\_IMPORT\_REF bool [is\\_zulu](#) () const  
*Manipulate internal flags to mark the time value as being in "zulu" (GMT) time.*
- OSCL\_COND\_IMPORT\_REF void [set\\_zulu](#) (bool is\_zulu)
- OSCL\_COND\_IMPORT\_REF [TimeValue & operator=](#) (const [TimeValue &a](#))  
*Assignment operator.*
- OSCL\_COND\_IMPORT\_REF [TimeValue & operator+=](#) (const [TimeValue &a](#))  
*+ = operator*
- OSCL\_COND\_IMPORT\_REF [TimeValue & operator-=](#) (const [TimeValue &a](#))  
*- = operator*
- OSCL\_COND\_IMPORT\_REF [TimeValue & operator\\*=](#) (const int scale)  
*This operator scales the time value by a constant.*
- OSCL\_COND\_IMPORT\_REF [OsclBasicTimeStruct \\* get\\_timeval\\_ptr](#) ()
- OSCL\_COND\_IMPORT\_REF [TimeValue & operator+=](#) (const int32 aSeconds)
- OSCL\_COND\_IMPORT\_REF [TimeValue & operator-=](#) (const int32 aSeconds)

## Friends

- class [NTPTime](#)
- OSCL\_COND\_IMPORT\_REF friend bool [operator==](#) (const [TimeValue &a](#), const [TimeValue &b](#))
- OSCL\_COND\_IMPORT\_REF friend bool [operator!=](#) (const [TimeValue &a](#), const [TimeValue &b](#))
- OSCL\_COND\_IMPORT\_REF friend bool [operator<=](#) (const [TimeValue &a](#), const [TimeValue &b](#))
- OSCL\_COND\_IMPORT\_REF friend bool [operator>=](#) (const [TimeValue &a](#), const [TimeValue &b](#))
- OSCL\_COND\_IMPORT\_REF friend bool [operator<](#) (const [TimeValue &a](#), const [TimeValue &b](#))
- OSCL\_COND\_IMPORT\_REF friend bool [operator>](#) (const [TimeValue &a](#), const [TimeValue &b](#))

### 7.265.1 Detailed Description

The [TimeValue](#) class represents a time value in a format native to the system. This class provides common time functions independent of any OS. The actual representation used is native to the system that the class is compiled on to increase efficiency. Macros used in this class:

- OSCL\_HAS\_ANSI\_STRING\_SUPPORT:

Definitions to determine the type of basic time structure used to store the time

- OSCL\_HAS\_UNIX\_TIME\_FUNCS
- OSCL\_HAS\_SYMBIAN\_SUPPORT
- OSCL\_HAS\_MSWIN\_SUPPORT

### 7.265.2 Constructor & Destructor Documentation

#### 7.265.2.1 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue ()

Create a [TimeValue](#) representing the current time.

#### 7.265.2.2 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (const TimeValue & *Tv*)

Copy constructor.

#### 7.265.2.3 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (long *tv*, TimeUnits *units*)

Create a [TimeValue](#) representing an interval of tv units.

##### Parameters

*tv* The number of units in the interval to be represented by this [TimeValue](#).

*units* The units of the tv argument. Must be in the enumeration TimeUnits.

#### 7.265.2.4 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (const OsclBasicTimeStruct & *in\_tv*)

Create a [TimeValue](#) representing the absolute time specified by the BasicTimeStruct.

##### Parameters

*in\_tv* OsclBasicTimeStruct as defined for each platform.

#### 7.265.2.5 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (const ISO8601timeStrBuf *time\_strbuf*)

#### 7.265.2.6 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (uint16 *aYear*, uint16 *aMonth*, uint16 *aDay*, uint16 *aHour*, uint16 *aMinute*, uint16 *aSecond*, uint16 *aMilliseconds*)

Create a [TimeValue](#) representing the absolute time specified by the year/month/day/hours/minutes/seconds/microseconds values passed as argument.

[TimeValue](#) constructor that sets time according to following input parameter for a specific date time. Please note that no argument is check for its validity (range etc) It might assert incase wrong argument are passed by user of this api.

#### Parameters

- ← **uint16** wYear;
- ← **uint16** wMonth; Jan = 1 to Dec = 12
- ← **uint16** wDay; 1-28/29/30/31
- ← **uint16** wHour; 0 to 23
- ← **uint16** wMinute; 0 to 59
- ← **uint16** wSecond; 0 to 59
- ← **uint16** wMilliseconds; 0 to 999

### 7.265.2.7 OSCL\_COND\_IMPORT\_REF TimeValue::TimeValue (OsclBasicDateTimeStruct *in\_ts*)

Create a [TimeValue](#) representing the absolute time specified by the BasicDateTimeStruct.

#### Parameters

*in\_ts* OsclBasicDateTimeStruct as defined for each platform provides the date in a readable format (i.e. day, date , week etc.) Notes: Implementation incomplete (= not done) on Win32, Wince, Symbian

## 7.265.3 Member Function Documentation

### 7.265.3.1 OSCL\_IMPORT\_REF int TimeValue::get\_ISO8601\_str\_time (ISO8601timeStrBuf *time\_strbuf*)

Get a PV extended text representation of the Time based on the ISO 8601 format.

#### Parameters

*time\_strbuf* A ISO8601timeStrBuf object to which the text representation will be written,

#### Returns

The number of characters copied to the buffer, not including the terminating null. The returned string is of the form "1985-04-12 10:15:30Z".

### 7.265.3.2 OSCL\_COND\_IMPORT\_REF int32 TimeValue::get\_local\_time ()

Get the local time after having adjusted for daylight saving.

Notes: Implementation incomplete (= not done) on Win32, Wince, Symbian

### 7.265.3.3 OSCL\_IMPORT\_REF int TimeValue::get\_pv8601\_str\_time (PV8601timeStrBuf *time\_strbuf*)

Get a PV extended text representation of the Time based on the PV 8601 format.

**Parameters**

*time\_strbuf* A PV8601timeStrBuf object to which the text representation will be written,

**Returns**

The number of characters copied to the buffer, not including the terminating null. The returned string is of the form "19850412T101530.047Z".

#### 7.265.3.4 OSCL\_IMPORT\_REF char\* TimeValue::get\_rfc822\_gmtime\_str (int *max\_time\_strlen*, char \* *time\_str*)

Get a text representation of the time in the GMT timezone based on the RFC 822 / RFC 1123 (also described in the HTTP spec RFC 2068 and RFC 2616).

**Parameters**

*max\_time\_strlen* The maximum number of characters that can be written to the buffer.

*time\_str* A pointer to the buffer to which the characters will be written.

**Returns**

Returns a pointer to the buffer (same as *time\_str*) containing a null terminated (c-style) string of the form "Wed, 30 Jun 1993 21:49:08 GMT".

#### 7.265.3.5 OSCL\_COND\_IMPORT\_REF uint32 TimeValue::get\_sec () const

Get a 32 bit value representing the seconds since the (system dependent) epoch.

**Returns**

This call returns a 32 bit value representing the number of seconds since the epoch. On unix systems this represents the number of seconds since the unix epoch Jan 1 1970. On Win32 this represents the number of seconds since Jan 1, 1601. This is intended to be used for intervals rather than for absolute time. (On Win32 for example, a 32 bit value would be too small to represent the number of seconds from the epoch until the current time.)

#### 7.265.3.6 OSCL\_IMPORT\_REF char\* TimeValue::get\_str\_ctime (CtimeStrBuf *ctime\_strbuf*)

Get a string containing a text representation of this [TimeValue](#) object.

**Parameters**

*ctime\_strbuf* A CtimeStrBuf object to which the text representation will be written,

**Returns**

A pointer to the input CtimeStrBuf is returned. This string is null terminated of the form "Wed Jun 30 21:49:08 1993".

**7.265.3.7 OSCL\_COND\_IMPORT\_REF OsclBasicTimeStruct\* TimeValue::get\_timeval\_ptr ()****7.265.3.8 OSCL\_COND\_IMPORT\_REF uint64 TimeValue::get\_timevalue\_in\_usec () const**

Get a 64 bit value representing the time value converted to microseconds.

**Returns**

Returns a uint64 value representing the time value in terms of microseconds. The time origin is dependent on platform for which OSCL is compiled. For example for symbian it is midnight, January 1st, 0 AD for windows it is January 1, 1601 (UTC)

**7.265.3.9 OSCL\_COND\_IMPORT\_REF uint32 TimeValue::get\_usec () const**

Get a 32 bit value representing the number of microseconds in the time value.

**Returns**

Returns a uint32 value representing the number of microseconds in the time value after subtracting off the whole seconds.

**7.265.3.10 OSCL\_COND\_IMPORT\_REF bool TimeValue::is\_zero ()**

Determine if the time value is zero.

**7.265.3.11 OSCL\_COND\_IMPORT\_REF bool TimeValue::is\_zulu () const**

Manipulate internal flags to mark the time value as being in "zulu" (GMT) time.

**7.265.3.12 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator\*= (const int *scale*)**

This operator scales the time value by a constant.

**7.265.3.13 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator+= (const int32 *aSeconds*)****7.265.3.14 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator+= (const TimeValue & *a*)**

*+=* operator

**7.265.3.15 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator-= (const int32 *aSeconds*)****7.265.3.16 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator-= (const TimeValue & *a*)**

*-=* operator

---

**7.265.3.17 OSCL\_COND\_IMPORT\_REF TimeValue& TimeValue::operator= (const TimeValue & a)**

Assignment operator.

**7.265.3.18 OSCL\_COND\_IMPORT\_REF void TimeValue::set\_from\_ntp\_time (const uint32 ntp\_offset)**

This method converts a 32-bit NTP offset to system time.

This method takes a 32-bit ntp offset which is the number of seconds from 0 h Jan 1, 1900 and converts it to the system time

**7.265.3.19 OSCL\_COND\_IMPORT\_REF void TimeValue::set\_to\_current\_time ()**

Set the time value to the current system time.

**7.265.3.20 OSCL\_COND\_IMPORT\_REF void TimeValue::set\_to\_zero ()**

Set the time value to zero (represents a zero interval).

**7.265.3.21 OSCL\_COND\_IMPORT\_REF void TimeValue::set\_zulu (bool is\_zulu)**

**7.265.3.22 OSCL\_COND\_IMPORT\_REF int32 TimeValue::to\_msec () const**

## 7.265.4 Friends And Related Function Documentation

**7.265.4.1 friend class NTPTime [friend]**

**7.265.4.2 OSCL\_COND\_IMPORT\_REF friend bool operator!= (const TimeValue & a, const TimeValue & b) [friend]**

**7.265.4.3 OSCL\_COND\_IMPORT\_REF friend bool operator< (const TimeValue & a, const TimeValue & b) [friend]**

**7.265.4.4 OSCL\_COND\_IMPORT\_REF friend bool operator<= (const TimeValue & a, const TimeValue & b) [friend]**

**7.265.4.5 OSCL\_COND\_IMPORT\_REF friend bool operator== (const TimeValue & a, const TimeValue & b) [friend]**

**7.265.4.6 OSCL\_COND\_IMPORT\_REF friend bool operator> (const TimeValue & a, const TimeValue & b) [friend]**

**7.265.4.7 OSCL\_COND\_IMPORT\_REF friend bool operator>= (const TimeValue & a, const TimeValue & b) [friend]**

The documentation for this class was generated from the following file:

- [oscl\\_time.h](#)

## 7.266 TLSStorageOps Class Reference

```
#include <oscl_tls.h>
```

### Static Public Member Functions

- static OSCL\_IMPORT\_REF void [save\\_registry \(TOsclTlsKey \\*key, OsclAny \\*ptr, int32 &\)](#)
- static OSCL\_IMPORT\_REF [OsclAny \\* get\\_registry \(TOsclTlsKey \\*key\)](#)

#### 7.266.1 Member Function Documentation

**7.266.1.1 static OSCL\_IMPORT\_REF OsclAny\* TLSStorageOps::get\_registry (TOsclTlsKey \**key*) [static]**

**7.266.1.2 static OSCL\_IMPORT\_REF void TLSStorageOps::save\_registry (TOsclTlsKey \* *key*, OsclAny \* *ptr*, int32 &) [static]**

The documentation for this class was generated from the following file:

- [oscl\\_tls.h](#)

## 7.267 TReadyQueLink Class Reference

```
#include <oscl_scheduler_readyq.h>
```

### Public Member Functions

- [TReadyQueLink \(\)](#)

### Data Fields

- int32 [iAOPriority](#)
- uint32 [iTimeToRunTicks](#)
- uint32 [iTimeQueuedTicks](#)
- uint32 [iSeqNum](#)
- [OsclAny \\* iIsIn](#)

#### 7.267.1 Detailed Description

This class defines the queue link, which is common to both ready Q and timer Q. Each AO contains its own queue link object.

#### 7.267.2 Constructor & Destructor Documentation

##### 7.267.2.1 [TReadyQueLink::TReadyQueLink \(\) \[inline\]](#)

References [iAOPriority](#), [iIsIn](#), [iSeqNum](#), [iTimeToRunTicks](#), and [NULL](#).

#### 7.267.3 Field Documentation

##### 7.267.3.1 [int32 TReadyQueLink::iAOPriority](#)

Referenced by [TReadyQueLink\(\)](#).

##### 7.267.3.2 [OsclAny\\* TReadyQueLink::iIsIn](#)

Referenced by [PVActiveBase::IsInAnyQ\(\)](#), and [TReadyQueLink\(\)](#).

##### 7.267.3.3 [uint32 TReadyQueLink::iSeqNum](#)

Referenced by [TReadyQueLink\(\)](#).

##### 7.267.3.4 [uint32 TReadyQueLink::iTimeQueuedTicks](#)

##### 7.267.3.5 [uint32 TReadyQueLink::iTimeToRunTicks](#)

Referenced by [TReadyQueLink\(\)](#).

The documentation for this class was generated from the following file:

- [oscl\\_scheduler\\_readyq.h](#)

## 7.268 Oscl\_Map< Key, T, Alloc, Compare >::value\_compare Class Reference

```
#include <oscl_map.h>
```

### Public Member Functions

- bool [operator\(\)](#) (const value\_type &x, const value\_type &y) const

### Protected Member Functions

- [value\\_compare](#) (Compare c)

### Protected Attributes

- Compare [comp](#)

### Friends

- class [Oscl\\_Map< Key, T, Alloc, Compare >](#)

**template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>> class Oscl\_Map< Key, T, Alloc, Compare >::value\_compare**

#### 7.268.1 Constructor & Destructor Documentation

**7.268.1.1 template<class Key, class T, class Alloc, class Compare = Oscl\_Less<Key>>  
Oscl\_Map< Key, T, Alloc, Compare >::value\_compare::value\_compare (Compare c)  
[[inline](#), [protected](#)]**

#### 7.268.2 Member Function Documentation

**7.268.2.1 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](#)]**

References [Oscl\\_Map< Key, T, Alloc, Compare >::value\\_compare::comp](#), and [Oscl\\_Pair< T1, T2 >::first](#).

### 7.268.3 Friends And Related Function Documentation

7.268.3.1 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> friend class Oscl_Map< Key, T, Alloc, Compare > [friend]`

### 7.268.4 Field Documentation

7.268.4.1 `template<class Key, class T, class Alloc, class Compare = Oscl_Less<Key>> Compare Oscl_Map< Key, T, Alloc, Compare >::value_compare::comp [protected]`

Referenced by `Oscl_Map< Key, T, Alloc, Compare >::value_compare::operator()()`.

The documentation for this class was generated from the following file:

- [oscl\\_map.h](#)

## 7.269 WStrPtrLen Struct Reference

This data structure encapsulates a set of functions used to perform.

```
#include <oscl_str_ptr_len.h>
```

### Public Member Functions

- `WStrPtrLen (const oscl_wchar *newPtr)`
- `WStrPtrLen (const oscl_wchar *newPtr, uint32 newLen)`
- `WStrPtrLen ()`
- `WStrPtrLen (const WStrPtrLen &rhs)`
- `const oscl_wchar * c_str () const`
- `int32 length () const`
- `int32 size () const`
- `void setPtrLen (const oscl_wchar *newPtr, uint32 newLen)`
- `c_bool isCIEquivalentTo (const WStrPtrLen &rhs) const`
- `int32 operator== (const WStrPtrLen &rhs) const`
- `int32 operator!= (const WStrPtrLen &rhs) const`
- `WStrPtrLen & operator= (const WStrPtrLen &rhs)`
- `WStrPtrLen & operator= (const oscl_wchar *rhs)`

### Protected Attributes

- `const oscl_wchar * ptr`
- `int32 len`

#### 7.269.1 Detailed Description

This data structure encapsulates a set of functions used to perform standard string operations. It should be used for null-terminated constant strings (non-modifiable) of wchar type.

#### 7.269.2 Constructor & Destructor Documentation

##### 7.269.2.1 WStrPtrLen::WStrPtrLen (const oscl\_wchar \* *newPtr*) [inline]

References len, and oscl\_strlen().

##### 7.269.2.2 WStrPtrLen::WStrPtrLen (const oscl\_wchar \* *newPtr*, uint32 *newLen*) [inline]

##### 7.269.2.3 WStrPtrLen::WStrPtrLen () [inline]

##### 7.269.2.4 WStrPtrLen::WStrPtrLen (const WStrPtrLen & *rhs*) [inline]

#### 7.269.3 Member Function Documentation

##### 7.269.3.1 const oscl\_wchar\* WStrPtrLen::c\_str () const [inline]

References ptr.

**7.269.3.2 c\_bool WStrPtrLen::isCIEquivalentTo (const WStrPtrLen & rhs) const [inline]**

References len, OSCL\_ASCII\_CASE\_MAGIC\_BIT, and ptr.

**7.269.3.3 int32 WStrPtrLen::length () const [inline]**

References len.

**7.269.3.4 int32 WStrPtrLen::operator!= (const WStrPtrLen & rhs) const [inline]****7.269.3.5 WStrPtrLen& WStrPtrLen::operator= (const oscl\_wchar \* rhs) [inline]**

References len, oscl\_strlen(), and ptr.

**7.269.3.6 WStrPtrLen& WStrPtrLen::operator= (const WStrPtrLen & rhs) [inline]**

References len, and ptr.

**7.269.3.7 int32 WStrPtrLen::operator== (const WStrPtrLen & rhs) const [inline]**

References len, oscl\_strncmp(), and ptr.

**7.269.3.8 void WStrPtrLen::setPtrLen (const oscl\_wchar \* newPtr, uint32 newLen) [inline]**

References len, and ptr.

**7.269.3.9 int32 WStrPtrLen::size () const [inline]**

References len.

## 7.269.4 Field Documentation

**7.269.4.1 int32 WStrPtrLen::len [protected]**

Referenced by isCIEquivalentTo(), length(), operator=(), operator==( ), setPtrLen(), size(), and WStrP-trLen().

**7.269.4.2 const oscl\_wchar\* WStrPtrLen::ptr [protected]**

Referenced by c\_str(), isCIEquivalentTo(), operator=(), operator==( ), and setPtrLen().

The documentation for this struct was generated from the following file:

- [oscl\\_str\\_ptr\\_len.h](#)



# Chapter 8

## File Documentation

### 8.1 oscl\_aostatus.h File Reference

Some basic types used with active objects.

```
#include "osclconfig_proc.h"
#include "oscl_base.h"
#include "oscl_aostatus.inl"
```

#### Data Structures

- class [OsclAOStatus](#)

#### Variables

- const int32 [OSCL\\_REQUEST\\_ERR\\_NONE](#) = 0
- const int32 [OSCL\\_REQUEST\\_PENDING](#) = (-0x7fffffff)
- const int32 [OSCL\\_REQUEST\\_ERR\\_CANCEL](#) = (-1)
- const int32 [OSCL\\_REQUEST\\_ERR\\_GENERAL](#) = (-2)

#### 8.1.1 Detailed Description

Some basic types used with active objects.

## 8.2 oscl\_assert.h File Reference

The file [oscl\\_assert.h](#) provides an OSCL\_ASSERT macro to document assumptions and test them during development.

```
#include "oscl_base.h"
#include "osclconfig.h"
#include "osclconfig_check.h"
#include "pv_config.h"
```

### Defines

- #define [OSCL\\_ASSERT](#)(\_expr) ((\_expr)?((void)0):OSCL\_Assert(# \_expr,\_\_FILE\_\_,\_\_LINE\_\_))

### Functions

- OSCL\_COND\_IMPORT\_REF void [\\_OSCL\\_Abort](#) ()  
*This function terminates the current process abnormally.*
- OSCL\_IMPORT\_REF void [OSCL\\_Assert](#) (const char \*expr, const char \*filename, int line\_number)  
*OSCL\_ASSERT macro evaluates an expression and when the result is false, prints a diagnostic message and aborts the program.*

### 8.2.1 Detailed Description

The file [oscl\\_assert.h](#) provides an OSCL\_ASSERT macro to document assumptions and test them during development.

## 8.3 oscl\_base.h File Reference

The file `oscl_base.h` is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros.

```
#include "osclconfig.h"
#include "oscl_base_macros.h"
#include "oscl_types.h"
#include "osclconfig_check.h"
#include "pv_config.h"
```

### Defines

- `#define OSCL_HAS_SINGLETON_SUPPORT 1`

### Functions

- `void PVOsclBase_Init ()`
- `void PVOsclBase_Cleanup ()`

#### 8.3.1 Detailed Description

The file `oscl_base.h` is the public header that should be included to pick up the platform configuration, basic type definitions, and common macros.

## 8.4 oscl\_base\_alloc.h File Reference

A basic allocator that does not rely on other modules.

```
#include "osclconfig.h"  
#include "oscl_defalloc.h"  
#include "oscl_base.h"  
#include "osclconfig_compiler_warnings.h"  
#include "osclconfig_memory.h"
```

### Data Structures

- class [\\_OsclBasicAllocator](#)

#### 8.4.1 Detailed Description

A basic allocator that does not rely on other modules.

## 8.5 oscl\_base\_macros.h File Reference

This file defines common macros and constants for basic compilation support.

```
#include "osclconfig.h"
```

### Defines

- #define **NULL** (0)  
*The NULL\_TERM\_CHAR is used to terminate c-style strings.*
- #define **OSCL\_INLINE** inline
- #define **OSCL\_COND\_EXPORT\_REF**
- #define **OSCL\_COND\_IMPORT\_REF**
- #define **OSCL\_CONST\_CAST**(type, exp) ((type)(exp))  
*Type casting macros.*
- #define **OSCL\_STATIC\_CAST**(type, exp) ((type)(exp))
- #define **OSCL\_REINTERPRET\_CAST**(type, exp) ((type)(exp))
- #define **OSCL\_DYNAMIC\_CAST**(type, exp) ((type)(exp))
- #define **OSCL\_VIRTUAL\_BASE**(type) type
- #define **OSCL\_UNUSED\_ARG**(vbl) (void)(vbl)
- #define **OSCL\_UNUSED\_RETURN**(value) return value
- #define **OSCL\_MIN**(a, b) ((a) < (b) ? (a) : (b))
- #define **OSCL\_MAX**(a, b) ((a) > (b) ? (a) : (b))
- #define **OSCL\_ABS**(a) ((a) > (0) ? (a) : -(a))
- #define **EPV\_ARM\_GNUC** 1
- #define **EPV\_ARM\_RVCT** 2
- #define **EPV\_ARM\_MSEVC** 3

### 8.5.1 Detailed Description

This file defines common macros and constants for basic compilation support.

## 8.6 oscl\_bin\_stream.h File Reference

Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.

```
#include "oscl_base.h"
#include "oscl_bin_stream.inl"
#include "oscl_mem_basic_functions.h"
```

### Data Structures

- class [OsclBinStream](#)
- class [OsclBinIStream](#)
- class [OsclBinIStreamLittleEndian](#)
- class [OsclBinIStreamBigEndian](#)
- class [OsclBinOStream](#)

*Class [OsclBinOStream](#) implements the basic stream functions for an output stream.*

- class [OsclBinOStreamLittleEndian](#)

*Class [OsclBinOStreamLittleEndian](#) implements a binary output stream using little endian byte ordering.*

- class [OsclBinOStreamBigEndian](#)

*Class [OsclBinOStreamBigEndian](#) implements a binary output stream using big endian byte ordering.*

### 8.6.1 Detailed Description

Defines a set of binary stream classes which handle portable input / output of binary data regardless of the native byte order.

## 8.7 oscl\_byte\_order.h File Reference

This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).

```
#include "oscl_base.h"
#include "oscl_byte_order.inl"
```

### Functions

- void [little\\_endian\\_to\\_host](#) (char \*data, uint32 size)  
*Convert little endian to host format.*
- void [host\\_to\\_little\\_endian](#) (char \*data, unsigned int size)  
*Convert host to little endian format.*
- void [big\\_endian\\_to\\_host](#) (char \*data, unsigned int size)  
*Convert big endian to host format.*
- void [host\\_to\\_big\\_endian](#) (char \*data, unsigned int size)  
*Convert host to big endian format.*

### 8.7.1 Detailed Description

This file defines functions providing byte ordering utility (e.g., switching between big and little endian orders).

## 8.8 oscl\_defalloc.h File Reference

The file defines simple default memory allocator classes. These allocators are used by the [Oscl\\_Vector](#) and [Oscl\\_Map](#) class, etc.

```
#include "oscl_base.h"
#include "osclconfig_compiler_warnings.h"
#include "oscl_mem_inst.h"
```

### Data Structures

- class [Oscl\\_Alloc](#)
- class [Oscl\\_Dalloc](#)
- class [Oscl\\_DefAlloc](#)
- class [OsclDestructDalloc](#)
- class [OsclAllocDestructDalloc](#)
- class [Oscl\\_TAlloc< T, Alloc >](#)
- struct [Oscl\\_TAlloc< T, Alloc >::rebind< U, V >](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [ALLOCATE\(n\)](#) allocate(n)
- #define [ALLOC\\_AND\\_CONSTRUCT\(n\)](#) alloc\_and\_construct(n)

### 8.8.1 Detailed Description

The file defines simple default memory allocator classes. These allocators are used by the [Oscl\\_Vector](#) and [Oscl\\_Map](#) class, etc.

## 8.9 oscl\_dll.h File Reference

Defines a DLL entry point.

```
#include "osclconfig.h"
```

### Defines

- #define **OSCL\_DLL\_ENTRY\_POINT()** void oscl\_dll\_entry\_point() {}
- #define **OSCL\_DLL\_ENTRY\_POINT\_DEFAULT()**

### 8.9.1 Detailed Description

Defines a DLL entry point.

## 8.10 oscl\_dns.h File Reference

The file [oscl\\_socket.h](#) defines the OSCL DNS APIs.

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_types.h"
#include "oscl_scheduler_types.h"
#include "oscl_namestring.h"
#include "oscl_stdstring.h"
#include "oscl_defalloc.h"
#include "oscl_socket_types.h"
#include "oscl_heapbase.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclDNSObserver](#)
- class [OsclDNS](#)

### Enumerations

- enum [TPVDNSFxn](#) { [EPVDNSGetHostByName](#) }
- enum [TPVDNSEvent](#) {  
    [EPVDNSSuccess](#), [EPVDNSPending](#), [EPVDNSTimeout](#), [EPVDNSFailure](#),  
    [EPVDNSCancel](#) }

### 8.10.1 Detailed Description

The file [oscl\\_socket.h](#) defines the OSCL DNS APIs.

## 8.11 oscl\_dns\_gethostbyname.h File Reference

```
#include "oscl_dns_method.h"
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_defalloc.h"
#include "oscl_socket.h"
#include "pvlogger.h"
#include "oscl_dns.h"
```

### Data Structures

- class [OsclGetHostByNameMethod](#)
- class [OsclGetHostByNameRequest](#)

## 8.12 oscl\_dns\_imp.h File Reference

```
#include "oscl_dns_tuneables.h"
#include "osclconfig_io.h"
#include "osclconfig_proc.h"
```

## 8.13 oscl\_dns\_imp\_base.h File Reference

```
#include "oscl_socket_imp.h"
#include "osclconfig_io.h"
#include "osclconfig_proc.h"
#include "oscl_socket_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_tuneables.h"
#include "oscl_defalloc.h"
#include "oscl_mutex.h"
#include "oscl_base.h"
#include "oscl_dns_tuneables.h"
#include "oscl_dns.h"
```

### Data Structures

- class [OsclDNSIBase](#)

## 8.14 oscl\_dns\_imp\_pv.h File Reference

```
#include "oscl_socket_imp_base.h"
#include "oscl_dns_request.h"
#include "oscl_dns_imp_base.h"
#include "oscl_socket_imp.h"
#include "oscl_dns.h"
```

### Data Structures

- class [OsclDNSI](#)

## 8.15 oscl\_dns\_method.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_dns.h"
#include "pvlogger.h"
```

### Data Structures

- class [OsclDNSMethod](#)
- class [OsclDNSRequestAO](#)

## 8.16 oscl\_dns\_param.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_dns_tuneables.h"
#include "oscl_namestring.h"
#include "oscl_dns.h"
#include "oscl_mutex.h"
#include "oscl_semaphore.h"
```

### Data Structures

- class [DNSRequestParam](#)
- class [GetHostByNameParam](#)

### Typedefs

- typedef [OsclMemAllocator TDNSRequestParamAllocator](#)

#### 8.16.1 Typedef Documentation

##### 8.16.1.1 typedef OsclMemAllocator TDNSRequestParamAllocator

## 8.17 oscl\_dns\_request.h File Reference

```
#include "oscl_dns_tuneables.h"
```

## 8.18 oscl\_dns\_tuneables.h File Reference

```
#include "osclconfig_io.h"  
#include "osclconfig_proc.h"
```

## 8.19 oscl\_double\_list.h File Reference

Internal use types for scheduler.

```
#include "osclconfig_proc.h"
#include "oscl_base.h"
#include "oscl_assert.h"
#include "oscl_double_list.inl"
```

### Data Structures

- class [OsclDoubleLink](#)
- class [OsclPriorityLink](#)
- class [OsclDoubleListBase](#)
- class [OsclDoubleList< T >](#)
- class [OsclPriorityList< T >](#)
- class [OsclDoubleRunner< T >](#)

### Defines

- #define [QUE\\_ITER\\_BEGIN](#)(\_type, \_qname)
- #define [QUE\\_ITER\\_END](#)(\_qname)

### Functions

- template<class T , class S >  
T \* [OsclPtrAdd](#) (T \*aPtr, S aVal)
- template<class T , class S >  
T \* [OsclPtrSub](#) (T \*aPtr, S aVal)

#### 8.19.1 Detailed Description

Internal use types for scheduler.

## 8.20 oscl\_errno.h File Reference

Defines functions to access additional information on errors where supported through an errno or similar service.

```
#include "oscl_base.h"
#include "osclconfig_error.h"
#include "oscl_errno.inl"
```

### Functions

- OSCL\_IMPORT\_REF bool **OSCL\_IsErrnoSupported** ()  
*oscl\_errno.h contains functions to access the global errno*
- OSCL\_IMPORT\_REF int **OSCL\_GetLastError** ()  
*This function returns the value of the system's global error number variable.*
- OSCL\_IMPORT\_REF bool **OSCL\_SetLastError** (int newVal)  
*This function sets the last error code for the system.*
- OSCL\_IMPORT\_REF char \* **OSCL\_StrError** (int errnum)  
*This function maps an error number to an error-message string.*

### 8.20.1 Detailed Description

Defines functions to access additional information on errors where supported through an errno or similar service.

## 8.21 oscl\_error.h File Reference

OSCL Error trap and cleanup include file.

```
#include "oscl_heapbase.h"
#include "osclconfig_error.h"
#include "oscl_base.h"
#include "oscl_defalloc.h"
#include "oscl_singleton.h"
#include "oscl_assert.h"
#include "oscl_tls.h"
```

### Data Structures

- class [OsclErrorTrap](#)
- class [OsclError](#)
- class [OsclSingletonRegistryEx](#)
- class [OsclSingletonEx< T, ID, Registry >](#)
- class [OsclTLSRegistryEx](#)
- class [OsclTLSEEx< T, ID, Registry >](#)

### Defines

- #define [OSCL\\_TRAPSTACK\\_PUSH\(a\)](#) OsclError::PushL(a)
- #define [OSCL\\_TRAPSTACK\\_POP\(\)](#) OsclError::Pop()
- #define [OSCL\\_TRAPSTACK\\_POPDEALLOC\(\)](#) OsclError::PopDealloc()

#### 8.21.1 Detailed Description

OSCL Error trap and cleanup include file.

## 8.22 oscl\_error\_allocator.h File Reference

Defines a memory allocation class used by the oscl error layer.

```
#include "oscl_base.h"  
#include "oscl_base_macros.h"  
#include "osclconfig_error.h"  
#include "oscl_assert.h"  
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclErrorAllocator](#)

*This class provides static methods to invoke the user defined memory allocation routines.*

#### 8.22.1 Detailed Description

Defines a memory allocation class used by the oscl error layer.

## 8.23 oscl\_error\_codes.h File Reference

Defines basic error and leave codes.

### Defines

- #define `OsclErrNone` 0
- #define `OsclErrGeneral` 100
- #define `OsclErrNoMemory` 101
- #define `OsclErrCancelled` 102
- #define `OsclErrNotSupported` 103
- #define `OsclErrArgument` 104
- #define `OsclErrBadHandle` 105
- #define `OsclErrAlreadyExists` 106
- #define `OsclErrBusy` 107
- #define `OsclErrNotReady` 108
- #define `OsclErrCorrupt` 109
- #define `OsclErrTimeout` 110
- #define `OsclErrOverflow` 111
- #define `OsclErrUnderflow` 112
- #define `OsclErrInvalidState` 113
- #define `OsclErrNoResources` 114
- #define `OsclErrNotInstalled` 115
- #define `OsclErrAlreadyInstalled` 116
- #define `OsclErrSystemCallFailed` 117
- #define `OsclErrNoHandler` 118
- #define `OsclErrThreadContextIncorrect` 119
- #define `OSCL_ERR_NONE` `OsclErrNone`
- #define `OSCL_BAD_ALLOC_EXCEPTION_CODE` `OsclErrNoMemory`
- #define `OsclSuccess` 0
- #define `OsclPending` 1
- #define `OsclFailure` -1

### Typedefs

- typedef int32 `OsclLeaveCode`
- typedef int32 `OsclReturnCode`

### 8.23.1 Detailed Description

Defines basic error and leave codes.

## 8.24 oscl\_error\_imp.h File Reference

Internal error implementation support.

```
#include "osclconfig_error.h"
#include "oscl_error_imp_jumps.h"
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_assert.h"
#include "oscl_error_codes.h"
#include "oscl_singleton.h"
#include "oscl_tls.h"
#include "oscl_base_alloc.h"
#include "oscl_error_trapcleanup.h"
#include "oscl_error.h"
```

### Defines

- #define PVERROR\_IMP\_JUMPS

#### 8.24.1 Detailed Description

Internal error implementation support.

## 8.25 oscl\_error\_imp\_cppexceptions.h File Reference

Implementation File for Leave using C++ exceptions.

```
#include "oscl_error_trapcleanup.h"
```

### Data Structures

- class [internalLeave](#)

### Defines

- #define [PVError\\_DoLeave\(\)](#) [internalLeave](#) \_\_ilv;\_\_ilv.a=0;throw(\_\_ilv)
- #define [\\_PV\\_TRAP\(](#)[\\_r](#), [\\_s](#))
- #define [\\_PV\\_TRAP\\_NO\\_TLS\(](#)[\\_trapimp](#), [\\_r](#), [\\_s](#))

### 8.25.1 Detailed Description

Implementation File for Leave using C++ exceptions.

## 8.26 oscl\_error\_imp\_fatalerror.h File Reference

Implementation File for Leave using system fatal error.

```
#include "oscl_assert.h"
```

### Defines

- #define [PVError\\_DoLeave\(\)](#) \_OSCL\_Abort()
- #define [\\_PV\\_TRAP\(\\_\\_r, \\_\\_s\)](#)
- #define [\\_PV\\_TRAP\\_NO\\_TLS\(\\_\\_tr, \\_\\_r, \\_\\_s\)](#)

### 8.26.1 Detailed Description

Implementation File for Leave using system fatal error.

## 8.27 oscl\_error\_imp\_jumps.h File Reference

Implementation of using Setjmp / Longjmp.

```
#include "oscl_error_trapcleanup.h"
#include "oscl_assert.h"
#include "osclconfig_error.h"
#include "oscl_defalloc.h"
#include "oscl_error.h"
```

### Data Structures

- class [OsclJump](#)

### Defines

- #define OSCL\_JUMP\_MAX\_JUMP\_MARKS OSCL\_MAX\_TRAP\_LEVELS
- #define internalLeave (-1)
- #define PVError\_DoLeave() OsclJump::StaticJump(internalLeave)
- #define \_PV\_TRAP(\_\_r, \_\_s)
- #define \_PV\_TRAP\_NO\_TLS(\_\_trapimp, \_\_r, \_\_s)

### 8.27.1 Detailed Description

Implementation of using Setjmp / Longjmp.

## 8.28 oscl\_error\_trapcleanup.h File Reference

OSCL Error trap and cleanup implementation include file.

```
#include "osclconfig_error.h"
#include "oscl_heapbase.h"
#include "oscl_error_imp.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"
```

### Data Structures

- class [OsclTrapStackItem](#)
- class [OsclTrapStack](#)
- class [OsclErrorTrapImp](#)

### Defines

- #define [OSCL\\_MAX\\_TRAP\\_LEVELS](#) 20
- #define [PVERRORTRAP\\_REGISTRY\\_ID](#) [OSCL\\_TLS\\_ID\\_PVERRORTRAP](#)
- #define [PVERRORTRAP\\_REGISTRY](#) [OsclTLSRegistry](#)

### 8.28.1 Detailed Description

OSCL Error trap and cleanup implementation include file.

## 8.29 oscl\_exception.h File Reference

contains all the exception handling macros and classes

```
#include "oscl_error.h"
#include "oscl_error_imp.h"
```

### Data Structures

- class [OsclException< LeaveCode >](#)

*oscl\_exception.h contains all the exception handling macros and classes*

### 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.*

- #define [OSCL\\_FIRST\\_CATCH\(\\_leave\\_status, \\_catch\\_value, \\_statements\)](#) if (\_leave\_-status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}

*Use this macro to define a block of code that catches the first exception type thrown in the preceding try block.*

- #define [OSCL\\_CATCH\(\\_leave\\_status, \\_catch\\_value, \\_statements\)](#) else if (\_leave\_-status!=OsclErrNone && \_leave\_status == \_catch\_value){\_statements;}

*Use this macro to define a block of code for catching additional exception types.*

- #define [OSCL\\_CATCH\\_ANY\(\\_leave\\_status, \\_statements\)](#) else if (\_leave\_status!=OsclErrNone){ \_-statements; }

*Use this macro to call a function that will catch all remaining exception types.*

- #define [OSCL\\_LAST\\_CATCH\(\\_leave\\_status\)](#) else if (\_leave\_status!=OsclErrNone){OSCL\_-LEAVE(\_leave\_status);}

*Use this macro if OSCL\_CATCH\_ANY has not been used. It will mark the end of the catch block.*

### 8.29.1 Detailed Description

contains all the exception handling macros and classes

## 8.30 oscl\_exclusive\_ptr.h File Reference

This file defines the [OsclExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

```
#include "oscl_defalloc.h"
#include "oscl_base.h"
```

### Data Structures

- 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 or indirectly) by new. When the OsclExclusivePtr expires, its destructor uses delete to free the memory.*

- 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 or indirectly) by new. When the OsclExclusiveArrayPtr expires, its destructor uses delete to free the memory.*

- 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 or indirectly) through Alloc. When the OsclExclusivePtrA expires, Alloc is used to free the memory.*

### 8.30.1 Detailed Description

This file defines the [OsclExclusivePtr](#) template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

## 8.31 oscl\_file\_async\_read.h File Reference

```
#include "oscl_base.h"
#include "osclconfig_io.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_scheduler_ao.h"
#include "oscl_file_io.h"
#include "oscl_string_containers.h"
#include "oscl_semaphore.h"
```

### Data Structures

- class [OsclPtrC](#)
- class [OsclPtr](#)
- class [OsclBuf](#)
- class [OsclAsyncFileBuffer](#)
- class [OsclAsyncFile](#)

## 8.32 oscl\_file\_cache.h File Reference

The file [oscl\\_file\\_cache.h](#) defines the class [OsclFileCache](#).

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_file_io.h"
```

### Data Structures

- class [OsclFileCacheBuffer](#)
- class [OsclFileCache](#)

#### 8.32.1 Detailed Description

The file [oscl\\_file\\_cache.h](#) defines the class [OsclFileCache](#).

## 8.33 oscl\_file\_dir\_utils.h File Reference

The file `oscl_file_dir_utils.h` defines some unix-style directory ops.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
```

### Data Structures

- struct `oscl_fsstat`
- struct `oscl_stat_buf`

### Typedefs

- typedef struct `oscl_fsstat` `OSCL_FSSTAT`
- typedef struct `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\_stats (OSCL\_FSSTAT \*stats, const char \*path)
- OSCL\_IMPORT\_REF OSCL\_FILEMGMT\_ERR\_TYPE oscl\_stats (OSCL\_FSSTAT \*stats, const oscl\_wchar \*path)

### 8.33.1 Detailed Description

The file [oscl\\_file\\_dir\\_utils.h](#) defines some unix-style directory ops.

## 8.34 oscl\_file\_find.h File Reference

The file [oscl\\_file\\_find.h](#) defines the class [Oscl\\_FileFind](#).

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_mem.h"
#include "oscl_vector.h"
#include "oscl_string_containers.h"
#include "oscl_file_types.h"
```

### Data Structures

- class [Oscl\\_FileFind](#)

#### 8.34.1 Detailed Description

The file [oscl\\_file\\_find.h](#) defines the class [Oscl\\_FileFind](#).

## 8.35 oscl\_file\_handle.h File Reference

The file [oscl\\_file\\_handle.h](#) defines the class [OsclFileHandle](#).

```
#include "osclconfig_io.h"
#include "oscl_base.h"
```

### Data Structures

- class [OsclFileHandle](#)

### TypeDefs

- typedef FILE \* [TOsclFileHandle](#)

#### 8.35.1 Detailed Description

The file [oscl\\_file\\_handle.h](#) defines the class [OsclFileHandle](#).

## 8.36 oscl\_file\_io.h File Reference

The file [oscl\\_file\\_io.h](#) defines the class [Oscl\\_File](#). This is the public API to the basic file I/O operations.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_mem.h"
#include "oscl_vector.h"
#include "oscl_file_server.h"
#include "oscl_file_find.h"
#include "oscl_file_dir_utils.h"
#include "oscl_file_handle.h"
```

### Data Structures

- class [Oscl\\_File](#)
- class [Oscl\\_File::OsclFixedCacheParam](#)
- class [Oscl\\_File::OsclCacheObserver](#)

### Defines

- #define [TOsclFileOffsetInt32](#) int32

#### 8.36.1 Detailed Description

The file [oscl\\_file\\_io.h](#) defines the class [Oscl\\_File](#). This is the public API to the basic file I/O operations.

## 8.37 oscl\_file\_manager.h File Reference

File management class.

```
#include "oscl_base.h"
```

### Data Structures

- class [OsclFileManager](#)

#### 8.37.1 Detailed Description

File management class.

## 8.38 oscl\_file\_native.h File Reference

The file [oscl\\_file\\_native.h](#) defines the class [OsclNativeFile](#). This is the porting layer for basic file I/O operations.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
#include "oscl_aostatus.h"
#include "oscl_file_io.h"
#include "oscl_file_types.h"
```

### Data Structures

- class [OsclNativeFile](#)

#### 8.38.1 Detailed Description

The file [oscl\\_file\\_native.h](#) defines the class [OsclNativeFile](#). This is the porting layer for basic file I/O operations.

## 8.39 oscl\_file\_server.h File Reference

The file [oscl\\_file\\_server.h](#) defines the class [Oscl\\_FileServer](#). This is the porting layer for file server implementations.

```
#include "osclconfig_io.h"
#include "oscl_base.h"
```

### Data Structures

- class [Oscl\\_FileServer](#)

#### 8.39.1 Detailed Description

The file [oscl\\_file\\_server.h](#) defines the class [Oscl\\_FileServer](#). This is the porting layer for file server implementations.

## 8.40 oscl\_file\_stats.h File Reference

File stats class.

```
#include "oscl_base.h"
#include "osclconfig_io.h"
```

### Data Structures

- class [OsclFileStatsItem](#)
- class [OsclFileStats](#)

### Defines

- #define [OSCL\\_FILE\\_STATS\\_LOGGER\\_NODE](#) "OsclFileStats"

### Enumerations

- enum [TOsclFileOp](#) {  
    [EOsclFileOp\\_Open](#), [EOsclFileOp\\_Close](#), [EOsclFileOp\\_Read](#), [EOsclFileOp\\_Write](#),  
    [EOsclFileOp\\_Seek](#), [EOsclFileOp\\_Tell](#), [EOsclFileOp\\_Size](#), [EOsclFileOp\\_Flush](#),  
    [EOsclFileOp\\_EndOfFile](#),   [EOsclFileOp\\_SetSize](#),   [EOsclFileOp\\_NativeOpen](#),   [EOsclFileOp\\_NativeClose](#),  
    [EOsclFileOp\\_NativeRead](#), [EOsclFileOp\\_NativeWrite](#), [EOsclFileOp\\_NativeSeek](#), [EOsclFileOp\\_NativeTell](#),  
    [EOsclFileOp\\_NativeSize](#),       [EOsclFileOp\\_NativeFlush](#),       [EOsclFileOp\\_NativeEndOfFile](#),  
    [EOsclFileOp\\_NativeSetSize](#),  
    [EOsclFileOp\\_Last](#) }

### 8.40.1 Detailed Description

File stats class.

## 8.41 oscl\_file\_types.h File Reference

The file [oscl\\_file\\_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.

### Data Structures

- class [OsclNativeFileParams](#)

### Defines

- #define [OSCL\\_IO\\_FILENAME\\_MAXLEN](#) 512
- #define [OSCL\\_IO\\_EXTENSION\\_MAXLEN](#) 512
- #define [OSCL\\_FILE\\_WCHAR\\_PATH\\_DELIMITER](#) \_STRLIT("/")
- #define [OSCL\\_FILE\\_CHAR\\_PATH\\_DELIMITER](#) \_STRLIT\_CHAR("/")

### 8.41.1 Detailed Description

The file [oscl\\_file\\_types.h](#) defines some constants and types for file I/O implementations. Anything that needs to be shared across implementation modules can go here.

## 8.42 oscl\_heapbase.h File Reference

OSCL Heap Base include file.

```
#include "osclconfig_error.h"
#include "oscl_base.h"
#include "oscl_heapbase.inl"
```

### Data Structures

- class [\\_OsclHeapBase](#)
- class [OsclTrapItem](#)

### Typedefs

- `typedef void(* OsclTrapOperation )(OsclAny *)`

#### 8.42.1 Detailed Description

OSCL Heap Base include file.

## 8.43 oscl\_init.h File Reference

Global oscl initialization.

```
#include "oscl_base.h"  
#include <stdio.h>
```

### Data Structures

- class [OsclSelect](#)
- class [OsclInit](#)

#### 8.43.1 Detailed Description

Global oscl initialization.

## 8.44 oscl\_int64\_utils.h File Reference

```
#include "oscl_base.h"
```

### Data Structures

- class [Oscl\\_Int64\\_Utils](#)  
*The Oscl\_Int64\_Utils class provides a wrapper for commonly used int64/uint64 operations.*
- struct [OsclInteger64Transport](#)

### Typedefs

- typedef struct [OsclInteger64Transport \\_OsclInteger64Transport](#)

#### 8.44.1 Typedef Documentation

##### 8.44.1.1 typedef struct OsclInteger64Transport \_OsclInteger64Transport

[OsclInteger64Transport](#) Structure

Structure to only transport 64-bit integer values uint64 and int64 could be classes so needed for cases where having a class will not work.

## 8.45 oscl\_ip\_socket.h File Reference

```
#include "oscl_socket_types.h"  
#include "oscl_vector.h"  
#include "oscl_mem.h"
```

### Data Structures

- class [OsclIPSocketI](#)

## 8.46 oscl\_linked\_list.h File Reference

The file [oscl\\_linked\\_list.h](#) defines the template class [Oscl\\_Linked\\_List](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
#include "oscl_opaque_type.h"
#include "oscl_assert.inl"
```

### Data Structures

- class [LinkedListElement< LLClass >](#)
- class [Oscl\\_Linked\\_List\\_Base](#)
- class [Oscl\\_Linked\\_List< LLClass, Alloc >](#)
- class [Oscl\\_MTLinked\\_List< LLClass, Alloc, TheLock >](#)

#### 8.46.1 Detailed Description

The file [oscl\\_linked\\_list.h](#) defines the template class [Oscl\\_Linked\\_List](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

## 8.47 oscl\_lock\_base.h File Reference

This file defines an abstract lock class, [OsclLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OsclNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OsclScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.

### Data Structures

- class [OsclLockBase](#)
- class [OsclNullLock](#)
- class [OsclScopedLock< 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.*

### 8.47.1 Detailed Description

This file defines an abstract lock class, [OsclLockBase](#), that is used for APIs potentially requiring multi-thread safety. A null-lock implementation, [OsclNullLock](#), is also provided for single-thread configurations (basically a noop for lock/unlock). Also provides the [OsclScopedLock](#) class which is template class takes care of freeing the lock when the class goes out of scope.

## 8.48 oscl\_map.h File Reference

The file `oscl_map.h` defines the template class `Oscl_Map` which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_tree.h"
#include "oscl_defalloc.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- struct `Oscl_Less< T >`
- struct `Oscl_Select1st< V, U >`
- class `Oscl_Map< Key, T, Alloc, Compare >`
- class `Oscl_Map< Key, T, Alloc, Compare >::value_compare`

### Defines

- #define `OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

#### 8.48.1 Detailed Description

The file `oscl_map.h` defines the template class `Oscl_Map` which has a very similar API as the STL Map class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

## 8.49 oscl\_math.h File Reference

Provides math functions.

```
#include "osclconfig_util.h"
#include "oscl_base.h"
#include "oscl_math.inl"
```

### Functions

- OSCL\_COND\_IMPORT\_REF double [oscl\\_log](#) (double value)
- OSCL\_COND\_IMPORT\_REF double [oscl\\_log10](#) (double value)
- OSCL\_COND\_IMPORT\_REF double [oscl\\_sqrt](#) (double value)
- OSCL\_COND\_IMPORT\_REF double [oscl\\_pow](#) (double x, double y)
- OSCL\_COND\_IMPORT\_REF double [oscl\\_exp](#) (double value)
- OSCL\_COND\_IMPORT\_REF double [oscl\\_sin](#) (double value)
- OSCL\_COND\_IMPORT\_REF double [oscl\\_cos](#) (double value)
- OSCL\_COND\_IMPORT\_REF double [oscl\\_tan](#) (double value)
- OSCL\_COND\_IMPORT\_REF double [oscl\\_asin](#) (double value)
- OSCL\_COND\_IMPORT\_REF double [oscl\\_atan](#) (double value)
- OSCL\_COND\_IMPORT\_REF double [oscl\\_floor](#) (double value)

### 8.49.1 Detailed Description

Provides math functions.

## 8.50 oscl\_media\_data.h File Reference

Defines a container class for media data made up of a collection of memory fragments.

```
#include "oscl_base.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_media_status.h"
```

### Data Structures

- class [MemAllocator< T >](#)
- class [BufferMgr](#)
- class [BufferState](#)
- class [BufferFragment](#)
- class [BuffFragGroup< ChainClass, max\\_frags >](#)
- class [MediaData< ChainClass, max\\_frags, local\\_bufsize >](#)

### Typedefs

- typedef void(\* [BufferFreeFuncPtr](#) )(void \*)
- typedef uint32 [MediaTimestamp](#)

#### 8.50.1 Detailed Description

Defines a container class for media data made up of a collection of memory fragments.

## 8.51 oscl\_media\_status.h File Reference

Defines a status values for the [MediaData](#) containers.

### Data Structures

- class [BuffFragStatusClass](#)
- class [MediaStatusClass](#)

### Variables

- const int32 [APPEND\\_MEDIA\\_AT\\_END](#) = -1

#### 8.51.1 Detailed Description

Defines a status values for the [MediaData](#) containers.

## 8.52 oscl\_mem.h File Reference

This file contains basic memory definitions for common use across platforms.

```
#include "osclconfig_memory.h"
#include "oscl_base.h"
#include "oscl_types.h"
#include "oscl_assert.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_lock_base.h"
#include "osclconfig_compiler_warnings.h"
#include "oscl_mem_inst.h"
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_refcounter.h"
#include "oscl_error.h"
#include "oscl_exception.h"
#include "oscl_mem.inl"
```

### Data Structures

- class [OsclMem](#)
- class [OsclAuditCB](#)
- class [OsclMemAllocator](#)
- class [OsclMemBasicAllocator](#)
- class [OsclMemAllocDestructDealloc< T >](#)
- class [OsclMemBasicAllocDestructDealloc< T >](#)
- class [OsclMemGlobalAuditObject](#)
- class [HeapBase](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #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\_malloc(count)
- #define [oscl\\_malloc\(a\)](#) OSCL\_MALLOC(a)
- #define [OSCL\\_DEFAULT\\_MALLOC\(x\)](#) OSCL\_MALLOC(x)
- #define [OSCL\\_AUDIT\\_MALLOC\(auditCB, count\)](#) \_oscl\_malloc(count)
- #define [OSCL\\_CALLOC\(num, size\)](#) \_oscl\_calloc(num,size)
- #define [oscl\\_calloc\(a, b\)](#) OSCL\_CALLOC(a,b)
- #define [OSCL\\_AUDIT\\_CALLOC\(auditCB, num, size\)](#) \_oscl\_calloc(num,size)

- #define OSCL\_REALLOC(ptr, new\_size) \_oscl\_realloc(ptr,new\_size)
- #define oscl\_realloc(a, b) OSCL\_REALLOC(a,b)
- #define OSCL\_AUDIT\_REALLOC(auditCB, ptr, new\_size) \_oscl\_realloc(ptr,new\_size)
- #define OSCL\_FREE(ptr) \_oscl\_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\_new(sizeof(T)),\_oscl\_free,T\_ptr,T,params)
- #define OSCL\_AUDIT\_NEW(auditCB, T, params) new(\_oscl\_default\_new(sizeof(T))) T params
- #define OSCL\_TRAP\_AUDIT\_NEW(T\_ptr, auditCB, T, params) \_OSCL\_TRAP\_NEW(\_oscl\_default\_new(sizeof(T)),\_oscl\_free,T\_ptr,T,params)
- #define OSCL\_DELETE(ptr)
- #define OSCL\_AUDIT\_ARRAY\_NEW(auditCB, T, count) new(\_oscl\_default\_new(sizeof(T)\*(count))) 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\_default\_new (size\_t nBytes)
- void \* operator new (size\_t aSize)
- void operator delete (void \*aPtr)
- void \* operator new[ ] (size\_t aSize)
- void operator delete[ ] (void \*aPtr)

### 8.52.1 Detailed Description

This file contains basic memory definitions for common use across platforms. This is the main entry point header file for the OSCL memory library. It should be the only one users directly include. Basic memory copy, compare, and move functions are defined here as well as the allocation functions.

## 8.53 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 "osclconfig_memory.h"
#include "oscl_base.h"
#include "oscl_types.h"
#include "oscl_assert.h"
#include "oscl_mem_basic_functions.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"
#include "oscl_mem.h"
```

### Data Structures

- struct [MM\\_Stats\\_t](#)
- struct [MM\\_FailInsertParam](#)
- class [OsclMemStatsNode](#)
- struct [MM\\_Stats\\_CB](#)
- struct [MM\\_AllocQueryInfo](#)
- struct [MM\\_AllocInfo](#)
- struct [MM\\_AllocNode](#)
- struct [MM\\_AuditOverheadStats](#)
- class [MM\\_Audit\\_Imp](#)
- class [OsclMemAudit](#)

### 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 > > MMAuditCharAutoPtr
- typedef OSCLMemAutoPtr< uint8, Oscl\_TAlloc< uint8, \_OsclBasicAllocator > > MMAuditUInt8AutoPtr
- typedef OSCLMemAutoPtr< MM\_AllocNode, Oscl\_TAlloc< MM\_AllocNode, OsclMemBasicAllocator > > MM\_AllocNodeAutoPtr
- typedef OSCLMemAutoPtr< OsclMemStatsNode, Oscl\_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > > MM\_StatsNodeTagTreeType
- typedef OSCLMemAutoPtr< OsclMemStatsNode, Oscl\_TAlloc< OsclMemStatsNode, OsclMemBasicAllocator > > OsclMemStatsNodeAutoPtr
- typedef Oscl\_TAlloc< MM\_StatsNodeTagTreeType, OsclMemBasicAllocator > TagTree\_Allocator
- typedef Oscl\_TagTree< MM\_StatsNodeTagTreeType, TagTree\_Allocator > OsclTagTreeType

### 8.53.1 Detailed Description

This file contains the definition and partial implementation of MM\_Audit class.

## 8.54 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_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"
#include "oscl_mem_inst.h"
```

### Data Structures

- struct [MM\\_AllocBlockHdr](#)
- struct [MM\\_AllocBlockFence](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [MM\\_AUDIT\\_ALLOC\\_NODE\\_SUPPORT](#) 1
- #define [MM\\_AUDIT\\_FENCE\\_SUPPORT](#) 0
- #define [MM\\_AUDIT\\_INCLUDE\\_ALL\\_HEAP\\_VALIDATION](#) 1
- #define [MM\\_AUDIT\\_FILL\\_SUPPORT](#) 0
- #define [MM\\_AUDIT\\_FAILURE\\_SIMULATION\\_SUPPORT](#) 1
- #define [FENCE\\_PATTERN](#) 0xAA
- #define [MIN\\_FENCE\\_SIZE](#) 4
- #define [MEM\\_ALIGN\\_SIZE](#) 8
- #define [COMPUTE\\_MEM\\_ALIGN\\_SIZE](#)(x, y, z) (y+(((x+y)%z) ? (z - (x+y)%z) : 0))
- #define [DEFAULT\\_PREFILL\\_PATTERN](#) 0x96
- #define [DEFAULT\\_POSTFILL\\_PATTERN](#) 0x5A

### 8.54.1 Detailed Description

This file contains the internal definitions for the mem audit library.

## 8.55 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 "oscl_mem.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- class [OSCLMemAutoPtr< T, \\_Allocator >](#)

*The oscl\_auto\_ptr class is a template class that defines a pointer like object intended to be assigned an address obtained (directly or or indirectly) by new. When the oscl\_auto\_ptr expires, its destructor uses delete to free the memory.*

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)
- #define [OSCL\\_DISABLE\\_WARNING\\_RETURN\\_TYPE\\_NOT\\_UDT](#)

### 8.55.1 Detailed Description

This file defines the oscl\_mem\_auto\_ptr template class. This class is used to avoid any potential memory leaks that may arise while returning from methods in case of error.

## 8.56 oscl\_mem\_basic\_functions.h File Reference

This file contains prototypes for the basic memory functions.

```
#include "oscl_base_macros.h"
#include "oscl_mem_basic_functions.inl"
#include "osclconfig_memory.h"
#include "osclconfig_compiler_warnings.h"
```

### Functions

- OSCL\_COND\_IMPORT\_REF void \* [\\_oscl\\_malloc](#) (int32 count)
- OSCL\_COND\_IMPORT\_REF void \* [\\_oscl\\_calloc](#) (int32 nelems, int32 size)
- OSCL\_COND\_IMPORT\_REF void \* [\\_oscl\\_realloc](#) (void \*src, int32 count)
- OSCL\_COND\_IMPORT\_REF void [\\_oscl\\_free](#) (void \*src)
- OSCL\_COND\_IMPORT\_REF void \* [oscl\\_memcpy](#) (void \*dest, const void \*src, uint32 count)
- OSCL\_COND\_IMPORT\_REF void \* [oscl\\_memmove](#) (void \*dest, const void \*src, uint32 count)
- OSCL\_COND\_IMPORT\_REF void \* [oscl\\_memmove32](#) (void \*dest, const void \*src, uint32 count)
- OSCL\_COND\_IMPORT\_REF void \* [oscl\\_memset](#) (void \*dest, uint8 val, uint32 count)
- OSCL\_COND\_IMPORT\_REF int [oscl\\_memcmp](#) (const void \*buf1, const void \*buf2, uint32 count)

### 8.56.1 Detailed Description

This file contains prototypes for the basic memory functions.

## 8.57 oscl\_mem\_inst.h File Reference

The file defines default memory instrumentation level.

```
#include "osclconfig_memory.h"
```

### Defines

- #define PVMEM\_INST\_LEVEL 0

#### 8.57.1 Detailed Description

The file defines default memory instrumentation level.

## 8.58 oscl\_mem\_mempool.h File Reference

This file contains the definition of memory pool allocators.

```
#include "oscl_mem.h"  
#include "oscl_defalloc.h"  
#include "oscl_vector.h"
```

### Data Structures

- class [OsclMemPoolFixedChunkAllocatorObserver](#)
- class [OsclMemPoolFixedChunkAllocator](#)
- class [OsclMemPoolResizableAllocatorObserver](#)
- class [OsclMemPoolResizableAllocatorMemoryObserver](#)
- class [OsclMemPoolResizableAllocator](#)
- struct [OsclMemPoolResizableAllocator::MemPoolBufferInfo](#)
- struct [OsclMemPoolResizableAllocator::MemPoolBlockInfo](#)

#### 8.58.1 Detailed Description

This file contains the definition of memory pool allocators.

## 8.59 oscl\_mutex.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"  
#include "oscl_types.h"  
#include "oscl_base.h"  
#include "oscl_thread.h"  
#include "oscl_lock_base.h"
```

### Data Structures

- class [OsclMutex](#)
- class [OsclThreadLock](#)

### Typedefs

- typedef [OsclMutex OsclNoYieldMutex](#)

#### 8.59.1 Detailed Description

This file provides implementation of mutex.

#### 8.59.2 Typedef Documentation

##### 8.59.2.1 typedef OsclMutex OsclNoYieldMutex

Class OsclNoYieldMutex can be used in use cases where there will be no CPU-yielding operation done while the Mutex is locked.

CPU-yielding operations include [OsclMutex::Lock](#), [OsclSemaphore::Wait](#), [OsclThread::Sleep](#), and [OsclBrewThreadUtil::BThreadYield](#).

The behavior of OsclNoYieldMutex depends on whether the threading model is pre-emptive or not. When threading is pre-emptive, it is identical to [OsclMutex](#). When threading is non-pre-emptive, it is a NO-OP.

An example of this type of use case is for simple data protection.

## 8.60 oscl\_namestring.h File Reference

Name string class include file.

```
#include "oscl_base.h"
```

### Data Structures

- class [OsclNameString<\\_\\_len>](#)

#### 8.60.1 Detailed Description

Name string class include file.

## 8.61 oscl\_opaque\_type.h File Reference

The file [oscl\\_opaque\\_type.h](#) defines pure virtual classes for working with opaque types.

```
#include "oscl_base.h"
```

### Data Structures

- class [Oscl\\_Opaque\\_Type\\_Alloc](#)
- class [Oscl\\_Opaque\\_Type\\_Compare](#)
- class [Oscl\\_Opaque\\_Type\\_Alloc\\_LL](#)

#### 8.61.1 Detailed Description

The file [oscl\\_opaque\\_type.h](#) defines pure virtual classes for working with opaque types.

## 8.62 oscl\_priqueue.h File Reference

Implements a priority queue data structure similar to STL.

```
#include "oscl_base.h"
#include "oscl_vector.h"
```

### Data Structures

- class [OsclPriorityQueueBase](#)
- class [OsclCompareLess< T >](#)
- class [OsclPriorityQueue< Qelem, Alloc, Container, Compare >](#)

#### 8.62.1 Detailed Description

Implements a priority queue data structure similar to STL. Implements a priority queue data structure similar to the STL class. The properties of the class include O(Log\_2(N)) insertion and deletion complexity and O(1) complexity to access the top priority item.

## 8.63 oscl\_procstatus.h File Reference

### Data Structures

- class [OsclProcStatus](#)

## 8.64 oscl\_queue.h File Reference

The file [oscl\\_queue.h](#) defines the template class [Oscl\\_Queue](#). It is similar to the STL::queue class, with some differences:

- less complete
- based on array rather than a deque
- some interfaces modeled on oscl\_vector, for ease of transition Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_assert.h"
#include "oscl_opaque_type.h"
```

### Data Structures

- class [Oscl\\_Queue\\_Base](#)
- class [Oscl\\_Queue< T, Alloc >](#)

#### 8.64.1 Detailed Description

The file [oscl\\_queue.h](#) defines the template class [Oscl\\_Queue](#). It is similar to the STL::queue class, with some differences:

- less complete
- based on array rather than a deque
- some interfaces modeled on oscl\_vector, for ease of transition Memory allocation is abstracted through the use of an allocator template parameter.

## 8.65 oscl\_rand.h File Reference

Provides pseudo-random number generation.

```
#include "osclconfig_util.h"
#include "oscl_base.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_rand.inl"
```

### Data Structures

- class [OsclRand](#)

#### 8.65.1 Detailed Description

Provides pseudo-random number generation.

## 8.66 oscl\_refcounter.h File Reference

A general purpose reference counter to object lifetimes.

```
#include "oscl_assert.h"  
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclRefCounter](#)
- class [OsclRefCounterDA](#)
- class [OsclRefCounterSA< DeallocType >](#)
- class [OsclRefCounterMTDA< LockType >](#)
- class [OsclRefCounterMTSA< DeallocType, LockType >](#)
- class [Oscl\\_DefAllocWithRefCounter< DefAlloc >](#)

#### 8.66.1 Detailed Description

A general purpose reference counter to object lifetimes.

## 8.67 oscl\_refcounter\_memfrag.h File Reference

This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount.

```
#include "oscl_base.h"
#include "oscl_refcounter.h"
#include "oscl_assert.h"
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclRefCounterMemFrag](#)

#### 8.67.1 Detailed Description

This file provides the definition of reference counted memory fragment, which provides access to a buffer and helps manage its lifetime through the refcount.

## 8.68 oscl\_registry\_access\_client.h File Reference

Client-side implementation Registry Access implementation.

```
#include "oscl_registry_types.h"
#include "oscl_types.h"
#include "oscl_string_containers.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclRegistryAccessClient](#)

#### 8.68.1 Detailed Description

Client-side implementation Registry Access implementation.

## 8.69 oscl\_registry\_client.h File Reference

Client-side implementation of OsclRegistry.

```
#include "oscl_registry_types.h"  
#include "oscl_mem.h"  
#include "oscl_string.h"
```

### Data Structures

- class [OsclRegistryClient](#)

#### 8.69.1 Detailed Description

Client-side implementation of OsclRegistry.

## 8.70 oscl\_registry\_client\_impl.h File Reference

Client-side implementation of OsclRegistryInterface.

```
#include "oscl_base.h"
#include "osclconfig_proc.h"
#include "oscl_vector.h"
#include "oscl_string.h"
#include "oscl_registry_types.h"
#include "oscl_registry_serv_impl_tls.h"
```

### Data Structures

- class [OsclRegistryClientImpl](#)
- class [OsclRegistryAccessClientImpl](#)
- class [OsclRegistryClientTlsImpl](#)
- class [OsclRegistryAccessClientTlsImpl](#)

#### 8.70.1 Detailed Description

Client-side implementation of OsclRegistryInterface.

## 8.71 oscl\_registry\_serv\_impl.h File Reference

Server-side implementation of OsclRegistry interfaces.

```
#include "oscl_base.h"
#include "osclconfig_proc.h"
#include "oscl_registry_types.h"
#include "oscl_string.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
```

### Data Structures

- class [OsclComponentRegistryElement](#)
- class [OsclComponentRegistryData](#)
- class [OsclComponentRegistry](#)

#### 8.71.1 Detailed Description

Server-side implementation of OsclRegistry interfaces.

## 8.72 oscl\_registry\_serv\_impl\_global.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_base.h"
```

## 8.73 oscl\_registry\_serv\_impl\_tls.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_registry_serv_impl.h"
#include "oscl_base.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 [OsclRegistryServTlsImpl](#)

## 8.74 oscl\_registry\_types.h File Reference

Common types used in Oscl registry interfaces.

```
#include "oscl_types.h"  
#include "oscl_string_containers.h"
```

### Data Structures

- class [OsclRegistryAccessElement](#)

### Typedefs

- typedef [OsclAny](#) \* [OsclComponentFactory](#)

#### 8.74.1 Detailed Description

Common types used in Oscl registry interfaces.

## 8.75 oscl\_scheduler.h File Reference

```
#include "oscl_scheduler_types.h"
#include "osclconfig_proc.h"
#include "oscl_base.h"
#include "oscl_aostatus.inl"
#include "oscl_heapbase.h"
#include "oscl_scheduler_types.h"
#include "oscl_namestring.h"
#include "oscl_aostatus.h"
#include "oscl_assert.h"
#include "oscl_double_list.inl"
#include "oscl_types.h"
#include "oscl_thread.h"
#include "oscl_lock_base.h"
#include "oscl_priqueue.h"
#include "oscl_base_alloc.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
#include "oscl_string_containers.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_semaphore.h"
#include "oscl_scheduler_readyq.h"
#include "oscl_defalloc.h"
```

### Data Structures

- class [OsclScheduler](#)
- class [OsclSchedulerObserver](#)
- class [OsclExecSchedulerCommonBase](#)
- class [OsclExecScheduler](#)
- class [PVSchedulerStopper](#)

### Defines

- #define [PVSCHEDNAMELEN](#) 30

## 8.76 oscl\_scheduler\_ao.h File Reference

Oscl Scheduler user execution object classes.

```
#include "oscl_scheduler_types.h"
#include "oscl_scheduler_aobase.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclActiveObject](#)
- class [OsclTimerObject](#)

#### 8.76.1 Detailed Description

Oscl Scheduler user execution object classes.

## 8.77 oscl\_scheduler\_aobase.h File Reference

Oscl Scheduler internal active object classes.

```
#include "oscl_scheduler_types.h"
#include "oscl_namestring.h"
#include "oscl_scheduler_threadcontext.h"
#include "oscl_scheduler_readyq.h"
#include "oscl_string_containers.h"
```

### Data Structures

- class [PVActiveBase](#)

### Defines

- #define [OSCL\\_ZEROIZE](#)(ptr, size) oscl\_memset(ptr, 0, size)
- #define [PVEEXECNAMELEN](#) 30

### 8.77.1 Detailed Description

Oscl Scheduler internal active object classes.

## 8.78 oscl\_scheduler\_readyq.h File Reference

ready q types for oscl scheduler

```
#include "oscl_scheduler_types.h"
#include "oscl_scheduler_tuneables.h"
#include "oscl_priqueue.h"
#include "oscl_base_alloc.h"
#include "oscl_semaphore.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
#include "oscl_string_containers.h"
```

### Data Structures

- class [OsclReadyAlloc](#)
- class [OsclReadyCompare](#)
- class [OsclTimerCompare](#)
- class [OsclReadyQ](#)
- class [OsclTimerQ](#)
- class [TReadyQueLink](#)

### Typedefs

- typedef [PVActiveBase](#) \* TOsclReady

#### 8.78.1 Detailed Description

ready q types for oscl scheduler

## 8.79 oscl\_scheduler\_threadcontext.h File Reference

Thread context functions needed by oscl scheduler.

```
#include "oscl_aostatus.h"  
#include "oscl_double_list.h"  
#include "oscl_mutex.h"
```

### Data Structures

- class [PVThreadContext](#)

### Enumerations

- enum [TPVThreadContext](#) { [EPVThreadContext\\_InThread](#), [EPVThreadContext\\_OsclThread](#), [EPVThreadContext\\_NonOsclThread](#), [EPVThreadContext\\_Undetermined](#) }

### 8.79.1 Detailed Description

Thread context functions needed by oscl scheduler.

## 8.80 oscl\_scheduler\_tuneables.h File Reference

Tunable settings for Oscl Scheduler.

```
#include "osclconfig_proc.h"
```

### Defines

- #define PV\_SCHED\_ENABLE\_LOOP\_STATS 0
- #define PV\_SCHED\_ENABLE\_PERF\_LOGGING 1
- #define PV\_SCHED\_ENABLE\_THREAD\_CONTEXT\_CHECKS 1
- #define PV\_SCHED\_LOG\_Q 0
- #define PV\_SCHED\_CHECK\_Q 0
- #define PV\_SCHED\_FAIR\_SCHEDULING 1
- #define OSCL\_PERF\_SUMMARY\_LOGGING 0

### 8.80.1 Detailed Description

Tunable settings for Oscl Scheduler.

## 8.81 oscl\_scheduler\_types.h File Reference

Scheduler common types include file.

```
#include "osclconfig_proc.h"
#include "oscl_aostatus.h"
#include "oscl_heapbase.h"
```

### Data Structures

- class [OsclExecSchedulerBase](#)

#### 8.81.1 Detailed Description

Scheduler common types include file.

## 8.82 oscl\_semaphore.h File Reference

This file provides implementation of mutex.

```
#include "osclconfig_proc.h"
#include "oscl_thread.h"
```

### Data Structures

- class [OsclSemaphore](#)

#### 8.82.1 Detailed Description

This file provides implementation of mutex.

## 8.83 oscl\_shared\_ptr.h File Reference

This file defines a template class [OsclSharedPtr](#) which is a "smart pointer" to the parameterized type.

```
#include "oscl_base.h"
#include "oscl_refcounter.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- class [OsclSharedPtr< TheClass >](#)  
*A parameterized smart pointer class.*

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_RETURN\\_TYPE\\_NOT\\_UDT](#)

#### 8.83.1 Detailed Description

This file defines a template class [OsclSharedPtr](#) which is a "smart pointer" to the parameterized type.

## 8.84 oscl\_singleton.h File Reference

This file defines the OsclSingleton class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time.

```
#include "oscl_base.h"  
#include "oscl_defalloc.h"
```

### 8.84.1 Detailed Description

This file defines the OsclSingleton class. This class provides a container which used to give access to a set of process-level singleton objects. Each object is indexed by an integer ID, listed below. There can only be one instance of each object per process at a given time. OsclSingleton is initialized in OsclBase::Init.

## 8.85 oscl\_snprintf.h File Reference

Provides a portable implementation of snprintf.

```
#include "oscl_base.h"
#include "osclconfig_util.h"
```

### Functions

- OSCL\_IMPORT\_REF int32 [oscl\\_snprintf](#) (char \*str, uint32 count, const char \*fmt,...)
- OSCL\_IMPORT\_REF int32 [oscl\\_snprintf](#) ([oscl\\_wchar](#) \*str, uint32 count, const [oscl\\_wchar](#) \*fmt,...)
- OSCL\_IMPORT\_REF int32 [oscl\\_vsnprintf](#) (char \*str, uint32 count, const char \*fmt, va\_list args)
- OSCL\_IMPORT\_REF int32 [oscl\\_vsnprintf](#) ([oscl\\_wchar](#) \*str, uint32 count, const [oscl\\_wchar](#) \*fmt, va\_list args)

### 8.85.1 Detailed Description

Provides a portable implementation of snprintf.

## 8.86 oscl\_socket.h File Reference

The file [oscl\\_socket.h](#) defines the OSCL Socket APIs.

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_heapbase.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclSocketServ](#)
- class [OsclUDPSocket](#)
- class [OsclTCPSocket](#)

#### 8.86.1 Detailed Description

The file [oscl\\_socket.h](#) defines the OSCL Socket APIs.

## 8.87 oscl\_socket\_accept.h File Reference

```
#include "oscl_socket_imp.h"
#include "oscl_socket_serv_imp.h"
#include "osclconfig_io.h"
#include "oscl_socket_tuneables.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_socket_request.h"
#include "pvlogger.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclAcceptMethod](#)
- class [OsclAcceptRequest](#)

## 8.88 oscl\_socket\_bind.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclBindMethod](#)
- class [OsclBindRequest](#)

## 8.89 oscl\_socket\_connect.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclConnectMethod](#)
- class [OsclConnectRequest](#)

## 8.90 oscl\_socket\_imp.h File Reference

```
#include "oscl_socket_tuneables.h"
```

## 8.91 oscl\_socket\_imp\_base.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_request.h"
#include "oscl_defalloc.h"
#include "oscl_mutex.h"
#include "oscl_socket_stats.h"
#include "oscl_base.h"
```

### Data Structures

- class [OsclSocketIBase](#)

## 8.92 oscl\_socket\_imp\_pv.h File Reference

```
#include "oscl_socket_imp_base.h"
```

### Data Structures

- class [OsclSocketI](#)

### Defines

- #define [PVSOCK\\_ERR\\_BAD\\_PARAM](#) (-1)
- #define [PVSOCK\\_ERR SOCK\\_NOT\\_OPEN](#) (-2)
- #define [PVSOCK\\_ERR SOCK\\_NO\\_SERV](#) (-3)
- #define [PVSOCK\\_ERR SERV\\_NOT\\_CONNECTED](#) (-4)
- #define [PVSOCK\\_ERR SOCK\\_NOT\\_CONNECTED](#) (-5)
- #define [PVSOCK\\_ERR NOT\\_IMPLEMENTED](#) (-6)
- #define [PVSOCK\\_ERR NOT\\_SUPPORTED](#) (-7)

### 8.92.1 Define Documentation

#### 8.92.1.1 #define PVSOCK\_ERR\_BAD\_PARAM (-1)

Socket implementation for PV socket server some error codes for request completion these are negative so they won't conflict with errors from the OS socket layer.

#### 8.92.1.2 #define PVSOCK\_ERR\_NOT\_IMPLEMENTED (-6)

#### 8.92.1.3 #define PVSOCK\_ERR\_NOT\_SUPPORTED (-7)

#### 8.92.1.4 #define PVSOCK\_ERR\_SERV\_NOT\_CONNECTED (-4)

#### 8.92.1.5 #define PVSOCK\_ERR SOCK\_NO\_SERV (-3)

#### 8.92.1.6 #define PVSOCK\_ERR SOCK\_NOT\_CONNECTED (-5)

#### 8.92.1.7 #define PVSOCK\_ERR SOCK\_NOT\_OPEN (-2)

## 8.93 oscl\_socket\_listen.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclListenMethod](#)
- class [OsclListenRequest](#)

### Defines

- #define [OSCL\\_SOCKET\\_LISTEN\\_H\\_INCLUDEDd](#)

#### 8.93.1 Define Documentation

##### 8.93.1.1 #define OSCL\_SOCKET\_LISTEN\_H\_INCLUDEDd

## 8.94 oscl\_socket\_method.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_types.h"
#include "oscl_scheduler_ao.h"
#include "oscl_socket_request.h"
#include "pvlogger.h"
#include "oscl_socket_tuneables.h"
#include "oscl_ip_socket.h"
#include "oscl_socket_imp.h"
```

### Data Structures

- class [OsclSocketMethod](#)
- class [OsclSocketRequestAO](#)

### Defines

- #define [MSEC\\_TO\\_MICROSEC](#) 1000

#### 8.94.1 Define Documentation

##### 8.94.1.1 #define MSEC\_TO\_MICROSEC 1000

## 8.95 oscl\_socket\_recv.h File Reference

```
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclRecvMethod](#)
- class [OsclRecvRequest](#)

## 8.96 oscl\_socket\_recv\_from.h File Reference

```
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclRecvFromMethod](#)
- class [OsclRecvFromRequest](#)

## 8.97 oscl\_socket\_request.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_tuneables.h"
```

### Data Structures

- class [PVSockBufSend](#)
- class [PVSockBufRecv](#)
- class [SocketRequestParam](#)
- class [SendParam](#)
- class [SendToParam](#)
- class [RecvParam](#)
- class [RecvFromParam](#)
- class [BindParam](#)
- class [ListenParam](#)
- class [ConnectParam](#)
- class [AcceptParam](#)
- class [ShutdownParam](#)

## 8.98 oscl\_socket\_send.h File Reference

```
#include "oscl_socket_types.h"  
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclSendMethod](#)
- class [OsclSendRequest](#)

## 8.99 oscl\_socket\_send\_to.h File Reference

```
#include "oscl_socket_types.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclSendToMethod](#)
- class [OsclSendToRequest](#)

## 8.100 oscl\_socket\_serv\_imp.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_socket_tuneables.h"
```

## 8.101 oscl\_socket\_serv\_imp\_base.h File Reference

```
#include "oscl_base.h"
#include "oscl_socket_stats.h"
```

### Data Structures

- class [OsclSocketServIBase](#)

## 8.102 oscl\_socket\_serv\_imp\_pv.h File Reference

```
#include "oscl_socket_serv_imp_base.h"
#include "oscl_base.h"
#include "oscl_socket_stats.h"
#include "oscl_socket_tuneables.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_scheduler_ao.h"
```

### Data Structures

- class [OsclSocketServI](#)

### Defines

- #define [OSCL\\_READSET\\_FLAG](#) 0x04
- #define [OSCL\\_WRITESET\\_FLAG](#) 0x02
- #define [OSCL\\_EXCEPTSET\\_FLAG](#) 0x01

#### 8.102.1 Define Documentation

##### 8.102.1.1 #define OSCL\_EXCEPTSET\_FLAG 0x01

##### 8.102.1.2 #define OSCL\_READSET\_FLAG 0x04

A bitmask for socket select operations

##### 8.102.1.3 #define OSCL\_WRITESET\_FLAG 0x02

## 8.103 oscl\_socket\_serv\_imp\_reqlist.h File Reference

```
#include "oscl_socket_tuneables.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OsclSocketServRequestQElem](#)
- class [OsclSocketServRequestList](#)

## 8.104 oscl\_socket\_shutdown.h File Reference

```
#include "oscl_socket_types.h"  
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclShutdownMethod](#)
- class [OsclShutdownRequest](#)

## 8.105 oscl\_socket\_stats.h File Reference

```
#include "oscl_base.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_mutex.h"
#include "oscl_socket_tuneables.h"
```

### Enumerations

- enum **TOsclSocketStatEvent** {
 EOscSocket\_RequestAO\_Success, EOscSocket\_RequestAO\_Canceled, EOscSocket\_RequestAO\_Error, EOscSocket\_RequestAO\_Timeout,
 EOscSocket\_ServRequestIssued, EOscSocket\_ServPoll, EOscSocket\_OS, EOscSocket\_Readable,
 EOscSocket\_Writable, EOscSocket\_Except, EOscSocket\_DataRecv, EOscSocket\_DataSent,
 EOscSocket\_ServRequestComplete, EOscSocket\_ServRequestCancelIssued, EOscSocketServ\_LoopsockOk, EOscSocketServ\_LoopsockError
 }
- enum **TOsclSocketServStatEvent** {
 EOscSocketServ\_SelectNoActivity = 0, EOscSocketServ\_SelectActivity, EOscSocketServ\_SelectRescheduleAsap, EOscSocketServ\_SelectReschedulePoll,
 EOscSocketServ\_LastEvent }

#### 8.105.1 Enumeration Type Documentation

##### 8.105.1.1 enum TOsclSocketServStatEvent

Enumerator:

*EOscSocketServ\_SelectNoActivity*  
*EOscSocketServ\_SelectActivity*  
*EOscSocketServ\_SelectRescheduleAsap*  
*EOscSocketServ\_SelectReschedulePoll*  
*EOscSocketServ\_LastEvent*

##### 8.105.1.2 enum TOsclSocketStatEvent

Socket diagnostics.

Enumerator:

*EOscSocket\_RequestAO\_Success*  
*EOscSocket\_RequestAO\_Canceled*  
*EOscSocket\_RequestAO\_Error*  
*EOscSocket\_RequestAO\_Timeout*

*EOsclSocket\_ServRequestIssued*  
*EOsclSocket\_ServPoll*  
*EOsclSocket\_OS*  
*EOsclSocket\_Readable*  
*EOsclSocket\_Writable*  
*EOsclSocket\_Except*  
*EOsclSocket\_DataRecv*  
*EOsclSocket\_DataSent*  
*EOsclSocket\_ServRequestComplete*  
*EOsclSocket\_ServRequestCancelIssued*  
*EOsclSocketServ\_LoopsockOk*  
*EOsclSocketServ\_LoopsockError*

## 8.106 oscl\_socket\_tuneables.h File Reference

```
#include "osclconfig_io.h"
#include "osclconfig_proc.h"
```

### Defines

- #define PV\_OSCL\_SOCKET\_STATS\_LOGGING 0
- #define PV\_SOCKET\_SERVER 1

#### 8.106.1 Define Documentation

##### 8.106.1.1 #define PV\_OSCL\_SOCKET\_STATS\_LOGGING 0

This file contains default definitions of all the the tuning parameters for the Oscl sockets.

If parameters are defined in [osclconfig\\_io.h](#), those definitions will take precedence over the ones in this file.

PV\_SOCKET\_REQUEST\_AO\_PRIORITY sets the priority of the socket request completion AOs.

Set this to 0 or 1 to enable/disable socket stats logging with "OsclSocketStats" node. This feature is fairly costly so should be off in production code.

##### 8.106.1.2 #define PV\_SOCKET\_SERVER 1

Enable/disable the PV socket server here.

## 8.107 oscl\_socket\_types.h File Reference

```
#include "osclconfig_io.h"
#include "oscl_types.h"
#include "oscl_scheduler_types.h"
#include "oscl_namestring.h"
#include "oscl_stdstring.h"
```

### Data Structures

- class OsclSocketTOS
- class OsclNetworkAddress
- class OsclIpMReq
- 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, EPVSocketNotImplemented }
- enum TPVSocketShutdown { EPVSocketSendShutdown, EPVSocketRecvShutdown, EPVSocketBothShutdown }
- enum TPVSocketOptionName { EPVIMulticastTTL, EPVIPAddMembership, EPVIPTOS, EPV-SockReuseAddr }
- enum TPVSocketOptionLevel { EPVIPProtoIP, EPVIPProtoTCP, EPVSocket }

#### 8.107.1 Define Documentation

##### 8.107.1.1 #define PVNETWORKADDRESS\_LEN 50

#### 8.107.2 Enumeration Type Documentation

##### 8.107.2.1 enum TPVSocketEvent

Return codes for asynchronous APIs

###### Enumerator:

*EPVSocketSuccess*

*EPVSocketPending*  
*EPVSocketTimeout*  
*EPVSocketFailure*  
*EPVSocketCancel*  
*EPVSocketNotImplemented*

#### 8.107.2.2 enum TPVSocketFxn

Enumerator:

*EPVSocketSend*  
*EPVSocketSendTo*  
*EPVSocketRecv*  
*EPVSocketRecvFrom*  
*EPVSocketConnect*  
*EPVSocketAccept*  
*EPVSocketShutdown*  
*EPVSocketBind*  
*EPVSocketListen*  
*EPVSocket\_Last*

#### 8.107.2.3 enum TPVSocketOptionLevel

Enumerator:

*EPVIPProtoIP*  
*EPVIPProtoTCP*  
*EPVSocket*

#### 8.107.2.4 enum TPVSocketOptionName

Enumerator:

*EPVIMulticastTTL*  
*EPVIPAddMembership*  
*EPVIPTOS*  
*EPVSockReuseAddr*

#### 8.107.2.5 enum TPVSocketShutdown

Enumerator:

*EPVSocketSendShutdown*  
*EPVSocketRecvShutdown*  
*EPVSocketBothShutdown*

## 8.108 oscl\_stdstring.h File Reference

This file provides standard string operations such as strlen, strcpy, etc. ANSI defines undefined behavior when the destination pointer is null for operations such as strcpy, strcat, etc. But, we chose to define one. In such cases, we return the destination as null.

```
#include "oscl_base.h"
```

### Functions

- OSCL\_IMPORT\_REF uint32 `oscl_strlen` (const char \*str)
- OSCL\_IMPORT\_REF uint32 `oscl_strlen` (const `oscl_wchar` \*str)
- OSCL\_IMPORT\_REF char \* `oscl_strncpy` (char \*dest, const char \*src, uint32 count)
- OSCL\_IMPORT\_REF `oscl_wchar` \* `oscl_strncpy` (`oscl_wchar` \*dest, const `oscl_wchar` \*src, uint32 count)
- OSCL\_IMPORT\_REF int32 `oscl_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)

### 8.108.1 Detailed Description

This file provides standard string operations such as strlen, strncpy, etc. ANSI defines undefined behavior when the destination pointer is null for operations such as strncpy, strncat, etc. But, we chose to define one. In such cases, we return the destination as null.

## 8.109 oscl\_str\_ptr\_len.h File Reference

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

```
#include "oscl_base.h"
#include "oscl_stdstring.h"
```

### Data Structures

- struct [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.*
- struct [StrCSumPtrLen](#)  
*same as [StrPtrLen](#), but includes checksum field and method to speed up querying*

### Typedefs

- typedef struct [StrPtrLen](#) [StrPtrLen](#)  
*This data structure encapsulates a set of functions used to perform.*
- typedef struct [WStrPtrLen](#) [WStrPtrLen](#)  
*This data structure encapsulates a set of functions used to perform.*
- typedef [StrCSumPtrLen](#) [StrCSumPtrLen](#)  
*same as [StrPtrLen](#), but includes checksum field and method to speed up querying*
- typedef [WStrPtrLen](#) [OSCL\\_TStrPtrLen](#)

### Variables

- const uint8 [OSCL\\_ASCII\\_CASE\\_MAGIC\\_BIT](#) = 0x20

#### 8.109.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

## 8.110 oscl\_string.h File Reference

Provides a standardized set of string containers that can be used in place of character arrays.

```
#include "oscl_base.h"
#include "oscl_mem.h"
```

### Data Structures

- class [OSCL\\_String](#)
- class [OSCL\\_wString](#)

### Enumerations

- enum [TOSCL\\_StringOp](#) { [EOSCL\\_StringOp\\_CompressASCII](#), [EOSCL\\_StringOp\\_UTF16ToUTF8](#) }
- enum [TOSCL\\_wStringOp](#) { [EOSCL\\_wStringOp\\_ExpandASCII](#), [EOSCL\\_wStringOp\\_UTF8ToUTF16](#) }

#### 8.110.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

## 8.111 oscl\_string\_containers.h File Reference

Provides a standardized set of string containers that can be used in place of character arrays.

```
#include "oscl_string.h"
#include "oscl_base.h"
#include "oscl_mem.h"
#include "oscl_defalloc.h"
#include "oscl_refcounter.h"
#include "oscl_error.h"
#include "oscl_stdstring.h"
```

### Data Structures

- class [OSCL\\_HeapString< Alloc >](#)
- class [OSCL\\_wHeapString< Alloc >](#)
- class [OSCL\\_HeapStringA](#)
- class [OSCL\\_wHeapStringA](#)
- class [OSCL\\_StackString< MaxBufSize >](#)
- class [OSCL\\_wStackString< MaxBufSize >](#)
- class [OSCL\\_FastString](#)
- class [OSCL\\_wFastString](#)

#### 8.111.1 Detailed Description

Provides a standardized set of string containers that can be used in place of character arrays.

## 8.112 oscl\_string\_rep.h File Reference

Contains some internal implementation for string containers.

```
#include "oscl_defalloc.h"
```

### Data Structures

- class [CHheapRep](#)
- class [CStackRep](#)
- class [CFastRep](#)

#### 8.112.1 Detailed Description

Contains some internal implementation for string containers.

## 8.113 oscl\_string\_uri.h File Reference

Utilities to unescape URIs.

```
#include "oscl_base.h"  
#include "oscl_string.h"
```

### Functions

- OSCL\_IMPORT\_REF bool [oscl\\_str\\_unescape\\_uri](#) (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes, uint32 &out\_buf\_len)  
*unescape any of the special escape sequence in the uri string*
- OSCL\_IMPORT\_REF bool [oscl\\_str\\_unescape\\_uri](#) (const [OSCL\\_String](#) &oscl\_str\_in, [OSCL\\_String](#) &oscl\_str\_out, uint32 &out\_buf\_len)  
*unescape any of the special escape sequence in the uri string*

### 8.113.1 Detailed Description

Utilities to unescape URIs.

## 8.114 oscl\_string\_utf8.h File Reference

Utilities to validate and truncate UTF-8 encoded strings.

```
#include "oscl_base.h"
```

### Functions

- OSCL\_IMPORT\_REF bool `oscl_str_is_valid_utf8` (const uint8 \*str\_buf, uint32 &num\_valid\_characters, uint32 max\_bytes=0, uint32 max\_char\_2\_valid=0, uint32 \*num\_byte\_4\_char=NULL)

*Check if the input string contains any illegal UTF-8 character. The function scans the string and validate that each character is a valid utf-8. It stops at the first NULL character, invalid character or the max\_byte value. The string is valid if and only if every character is a valid utf-8 character and the scanning stopped on a character boundary.*

- OSCL\_IMPORT\_REF int32 `oscl_str_truncate_utf8` (uint8 \*str\_buf, uint32 max\_char, uint32 max\_bytes=0)

*Truncates the UTF-8 string upto the required size.*

### 8.114.1 Detailed Description

Utilities to validate and truncate UTF-8 encoded strings.

## 8.115 oscl\_string\_utils.h File Reference

Utilities to parse and convert strings.

```
#include "oscl_base.h"
```

### Defines

- #define `oscl_isdigit(c) ((c) >= '0' && (c) <= '9')`

### Functions

- OSCL\_IMPORT\_REF const char \* `skip_whitespace` (const char \*ptr)
- OSCL\_IMPORT\_REF char \* `skip_whitespace` (char \*ptr)
- OSCL\_IMPORT\_REF const char \* `skip_whitespace` (const char \*start, const char \*end)
- OSCL\_IMPORT\_REF const char \* `skip_to_whitespace` (const char \*start, const char \*end)
- OSCL\_IMPORT\_REF const char \* `skip_to_line_term` (const char \*start\_ptr, const char \*end\_ptr)
- OSCL\_IMPORT\_REF const char \* `skip_whitespace_and_line_term` (const char \*start, const char \*end)
- OSCL\_IMPORT\_REF int `extract_string` (const char \*in\_ptr, char \*outstring, int maxsize)
- OSCL\_IMPORT\_REF int `extract_string` (const char \*start, const char \*end, char \*outstring, int maxsize)
- OSCL\_IMPORT\_REF bool `PV_atoi` (const char \*buf, const char new\_format, uint32 &value)
- OSCL\_IMPORT\_REF bool `PV_atoi` (const char \*buf, const char new\_format, int length, uint32 &value)
- OSCL\_IMPORT\_REF bool `PV_atoi` (const char \*buf, const char new\_format, int length, `uint64` &value)
- OSCL\_IMPORT\_REF bool `PV_atof` (const char \*buf, `OsclFloat` &value)
- OSCL\_IMPORT\_REF bool `PV_atof` (const char \*buf, int length, `OsclFloat` &value)
- OSCL\_IMPORT\_REF int `oscl_abs` (int aVal)

### 8.115.1 Detailed Description

Utilities to parse and convert strings.

## 8.116 oscl\_string\_xml.h File Reference

Utilities to escape special characters in XML strings.

```
#include "oscl_base.h"
```

### Functions

- OSCL\_IMPORT\_REF bool [oscl\\_str\\_need\\_escape\\_xml](#) (const char \*str\_buf, uint32 &num\_escape\_bytes, uint32 max\_bytes=0)

*Check if the input string contains any special ASCII character like &, <, >, ', ". The function scans the string and check if each character is a special character. It stops at the first NULL character (if max\_bytes = 0), or the max\_byte value.*

- OSCL\_IMPORT\_REF int32 [oscl\\_str\\_escape\\_xml](#) (const char \*str\_buf\_in, char \*str\_buf\_out, uint32 max\_out\_buf\_bytes, uint32 max\_bytes=0, uint32 \*num\_bytes\_written=NULL)

*Escape any of the following special characters in the string Special ASCII characters: &, <, >, ', ".*

### 8.116.1 Detailed Description

Utilities to escape special characters in XML strings.

## 8.117 oscl\_tagtree.h File Reference

The file [oscl\\_tagtree.h](#) ...

```
#include "oscl_base.h"
#include "oscl_map.h"
#include "oscl_tree.h"
#include "osclconfig_compiler_warnings.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_assert.h"
#include "oscl_opaque_type.h"
#include "oscl_defalloc.h"
```

### Data Structures

- struct [Oscl\\_Tag\\_Base](#)
- struct [Oscl\\_Tag< Alloc >](#)
- class [Oscl\\_TagTree< T, Alloc >](#)
- struct [Oscl\\_TagTree< T, Alloc >::Node](#)
- struct [Oscl\\_TagTree< T, Alloc >::iterator](#)
- struct [Oscl\\_TagTree< T, Alloc >::const\\_iterator](#)

### Defines

- #define [OSCL\\_DISABLE\\_WARNING\\_TRUNCATE\\_DEBUG\\_MESSAGE](#)

### 8.117.1 Detailed Description

The file [oscl\\_tagtree.h](#) ...

## 8.118 oscl\_tcp\_socket.h File Reference

```
#include "oscl_ip_socket.h"
#include "oscl_defalloc.h"
#include "oscl_vector.h"
#include "oscl_mem.h"
#include "oscl_socket_listen.h"
#include "oscl_socket_types.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
```

### Data Structures

- class [OsclTCPSocketI](#)

## 8.119 oscl\_thread.h File Reference

```
#include "osclconfig_proc.h"
#include "oscl_procstatus.h"
#include "oscl_base.h"
```

### Data Structures

- class [OsclThread](#)

### Typedefs

- typedef [TOsclThreadFuncRet\(OSCL\\_THREAD DECL \\* TOsclThreadFuncPtr \)\(TOsclThreadFuncArg\)](#)

### Enumerations

- enum [OsclThread\\_State](#) { [Start\\_on\\_creation](#), [Suspend\\_on\\_creation](#) }
- enum [OsclThreadPriority](#) {
 [ThreadPriorityLowest](#), [ThreadPriorityLow](#), [ThreadPriorityBelowNormal](#), [ThreadPriorityNormal](#),
 [ThreadPriorityAboveNormal](#), [ThreadPriorityHighest](#), [ThreadPriorityTimeCritical](#) }
- enum [TOsclThreadTerminate](#) { [EOsclThreadTerminate\\_Join](#), [EOsclThreadTerminate\\_Kill](#),
[EOsclThreadTerminate\\_NOP](#) }

### 8.119.1 Detailed Description

.This file provides THREAD implementation that can be ported  
to three OS LINUX, SYMBIAN, WIN32

### 8.119.2 Typedef Documentation

#### 8.119.2.1 [typedef TOsclThreadFuncRet\(OSCL\\_THREAD DECL \\* TOsclThreadFuncPtr \)\(TOsclThreadFuncArg\)](#)

### 8.119.3 Enumeration Type Documentation

#### 8.119.3.1 enum [OsclThread\\_State](#)

**Enumerator:**

*Start\_on\_creation*

*Suspend\_on\_creation*

### 8.119.3.2 enum OsclThreadPriority

Enumerator:

*ThreadPriorityLowest*  
*ThreadPriorityLow*  
*ThreadPriorityBelowNormal*  
*ThreadPriorityNormal*  
*ThreadPriorityAboveNormal*  
*ThreadPriorityHighest*  
*ThreadPriorityTimeCritical*

### 8.119.3.3 enum TOsclThreadTerminate

Enumerator:

*EOsclThreadTerminate\_Join*  
*EOsclThreadTerminate\_Kill*  
*EOsclThreadTerminate\_NOP*

## 8.120 oscl\_tickcount.h File Reference

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

```
#include "oscl_base.h"
#include "oscl_tickcount.inl"
#include "osclconfig.h"
#include "oscl_singleton.h"
#include "osclconfig_time.h"
```

### Data Structures

- class [OsclTickCount](#)

### Defines

- #define [OSCLTICKCOUNT\\_MAX\\_TICKS](#) 0xffffffff

### 8.120.1 Detailed Description

Defines a data structure for string containment/manipulations where the storage for the string is maintained externally.

## 8.121 oscl\_time.h File Reference

The file `oscl_time.h` defines two classes `NTPTime` and `TimeValue` for getting, manipulating, and formatting time values. The `TimeValue` class is based on the native system time format while `NTPTime` is used for the standard Network Time Protocol format.

```
#include "oscl_base.h"
#include "osclconfig_time.h"
#include "oscl_int64_utils.h"
#include "oscl_int64_utils.h"
#include "oscl_assert.h"
#include "oscl_stdstring.h"
#include "oscl_mem_basic_functions.h"
```

### Data Structures

- class `NTPTime`

*The `NTPTime` class represents a time value as the number of seconds since 0h (UTC) Jan. 1, 1900.*

- class `TimeValue`

*The `TimeValue` class represents a time value in a format native to the system.*

### Typedefs

- `typedef char CtimeStrBuf [CTIME_BUFFER_SIZE]`
- `typedef char PV8601timeStrBuf [PV8601TIME_BUFFER_SIZE]`
- `typedef char ISO8601timeStrBuf [ISO8601TIME_BUFFER_SIZE]`

### Enumerations

- enum `TimeUnits` { `SECONDS` = 0, `MILLISECONDS` = 1, `MICROSECONDS` = 2 }

*The `TimeUnits` enum can be used when constructing a `TimeValue` class.*

### Functions

- `OSCL_IMPORT_REF void PV8601ToRFC822 (PV8601timeStrBuf pv8601_buffer, CtimeStrBuf ctime_buffer)`
- `OSCL_IMPORT_REF void ISO8601ToRFC822 (ISO8601timeStrBuf iso8601_buffer, CtimeStrBuf ctime_buffer)`
- `OSCL_IMPORT_REF void RFC822ToPV8601 (CtimeStrBuf ctime_buffer, PV8601timeStrBuf)`
- `OSCL_COND_IMPORT_REF TimeValue operator- (const TimeValue &a, const TimeValue &b)`
- `OSCL_COND_IMPORT_REF TimeValue operator+ (const TimeValue &a, const int32 bSeconds)`
- `OSCL_COND_IMPORT_REF TimeValue operator+ (const int32 aSeconds, const TimeValue &b)`
- `OSCL_COND_IMPORT_REF TimeValue operator- (const TimeValue &a, const int32 bSeconds)`
- `OSCL_COND_IMPORT_REF TimeValue operator- (const int32 aSeconds, const TimeValue &b)`

## Variables

- const int `CTIME_BUFFER_SIZE` = 26
- const int `PV8601TIME_BUFFER_SIZE` = 21
- const int `ISO8601TIME_BUFFER_SIZE` = 21
- const long `USEC_PER_SEC` = 1000000
- const long `MSEC_PER_SEC` = 1000
- const uint32 `unix_ntp_offset` = 2208988800U

### 8.121.1 Detailed Description

The file `oscl_time.h` defines two classes `NTPTime` and `TimeValue` for getting, manipulating, and formatting time values. The `TimeValue` class is based on the native system time format while `NTPTime` is used for the standard Network Time Protocol format.

## 8.122 oscl\_timer.h File Reference

```
#include "oscl_base.h"
#include "osclconfig_util.h"
#include "oscl_vector.h"
#include "oscl_tickcount.h"
#include "oscl_rand.h"
#include "oscl_scheduler_ao.h"
```

### Data Structures

- class [OsclTimerObserver](#)
- class [CallbackTimerObserver](#)
- class [CallbackTimer< Alloc >](#)
- class [OsclTimer< Alloc >](#)
- struct [OsclTimer< Alloc >::\\_TimerEntry](#)

## 8.123 oscl\_tls.h File Reference

```
#include "oscl_base.h"
#include "oscl_defalloc.h"
```

### Data Structures

- class [TLSStorageOps](#)
- class [OsclTLSRegistry](#)
- class [OsclTLS< T, ID, Registry >](#)

### Defines

- #define [OSCL\\_TLS\\_BASE\\_SLOTS](#) OSCL\_TLS\_ID\_BASE\_LAST +1
- #define [OSCL\\_TLS\\_MAX\\_SLOTS](#) ( OSCL\_TLS\_BASE\_SLOTS + OSCL\_TLS\_EXTERNAL\_SLOTS )

### Typedefs

- typedef [OsclAny](#) TOsclTlsKey

### Variables

- const uint32 [OSCL\\_TLS\\_ID\\_MAGICNUM](#) = 0
- const uint32 [OSCL\\_TLS\\_ID\\_ERRORHOOK](#) = 1
- const uint32 [OSCL\\_TLS\\_ID\\_PVLOGGER](#) = 2
- const uint32 [OSCL\\_TLS\\_ID\\_TEST](#) = 3
- const uint32 [OSCL\\_TLS\\_ID\\_PVSCHEDULER](#) = 4
- const uint32 [OSCL\\_TLS\\_ID\\_PVERRORTRAP](#) = 5
- const uint32 [OSCL\\_TLS\\_ID\\_SDPMEDIAPARSER](#) = 6
- const uint32 [OSCL\\_TLS\\_ID\\_PAYLOADPARSER](#) = 7
- const uint32 [OSCL\\_TLS\\_ID\\_PVMFRECOGNIZER](#) = 8
- const uint32 [OSCL\\_TLS\\_ID\\_WMDRM](#) = 9
- const uint32 [OSCL\\_TLS\\_ID\\_OSCLREGISTRY](#) = 10
- const uint32 [OSCL\\_TLS\\_ID\\_SQLITE3](#) = 11
- const uint32 [OSCL\\_TLS\\_ID\\_BASE\\_LAST](#) = 11

## 8.124 oscl\_tree.h File Reference

The file `oscl_tree.h` defines the template class `Oscl_Rb_Tree` which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the `Oscl_Map` class. Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_defalloc.h"
#include "oscl_base.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- struct `Oscl_Pair< T1, T2 >`
- struct `Oscl_Rb_Tree_Node_Base`
- struct `Oscl_Rb_Tree_Node< Value >`
- struct `Oscl_Rb_Tree_Iterator< Value >`
- struct `Oscl_Rb_Tree_Const_Iterator< Value >`
- class `Oscl_Rb_Tree_Base`
- class `Oscl_Rb_Tree< Key, Value, KeyOfValue, Compare, Alloc >`

### Defines

- #define `OSCL_DISABLE_WARNING_TRUNCATE_DEBUG_MESSAGE`

#### 8.124.1 Detailed Description

The file `oscl_tree.h` defines the template class `Oscl_Rb_Tree` which has a very similar API as the STL Tree class. It is an implementation of a Red-Black Tree for use by the `Oscl_Map` class. Memory allocation is abstracted through the use of an allocator template parameter.

## 8.125 oscl\_types.h File Reference

This file contains basic type definitions for common use across platforms.

```
#include "osclconfig.h"
```

### Data Structures

- struct [OsclMemoryFragment](#)

### Typedefs

- typedef int [c\\_bool](#)

*The c\_bool type is mapped to an integer to provide a bool type for C interfaces.*

- typedef void [OsclAny](#)

*The OsclAny is meant to be used the context of a generic pointer (i.e., no specific type).*

- typedef char [mbchar](#)

*mbchar is multi-byte char (e.g., UTF-8) with null termination.*

- typedef unsigned int [uint](#)

*The uint type is a convenient abbreviation for unsigned int.*

- typedef uint8 [octet](#)

*The octet type is meant to be used for referring to a byte or collection bytes without suggesting anything about the underlying meaning of the bytes.*

- typedef float [OsclFloat](#)

*The Float type defined as OsclFloat.*

- typedef size\_t [OsclSizeT](#)

- typedef OSCL\_NATIVE\_INT64\_TYPE [int64](#)

- typedef OSCL\_NATIVE\_UINT64\_TYPE [uint64](#)

- typedef OSCL\_NATIVE\_WCHAR\_TYPE [oscl\\_wchar](#)

- typedef [oscl\\_wchar OSCL\\_TCHAR](#)

*define OSCL\_TCHAR*

### 8.125.1 Detailed Description

This file contains basic type definitions for common use across platforms.

## 8.126 oscl\_udp\_socket.h File Reference

```
#include "oscl_ip_socket.h"
#include "oscl_defalloc.h"
#include "oscl_socket_recv_from.h"
#include "oscl_socket_serv_imp.h"
#include "oscl_socket_imp.h"
#include "oscl_socket_method.h"
#include "oscl_socket_types.h"
#include "oscl_socket_bind.h"
```

### Data Structures

- class [OsclUDPSocketI](#)

## 8.127 oscl\_utf8conv.h File Reference

Utilities to convert unicode to utf8 and vice versa.

```
#include "oscl_base.h"
```

### Defines

- #define MAX\_NUMBER\_OF\_BYTE\_PER\_UTF8 3

### Functions

- OSCL\_IMPORT\_REF int32 **oscl\_UTF8ToUnicode** (const char \*input, int32 inLength, **oscl\_wchar** \*output, int32 outLength)  
*Convert UTF8 byte sequence to Unicode UTF-16 string.*
- OSCL\_IMPORT\_REF int32 **oscl\_UnicodeToUTF8** (const **oscl\_wchar** \*input, int32 inLength, char \*output, int32 outLength)  
*Convert UTF-16 Unicode string to UTF8 byte sequence.*

### 8.127.1 Detailed Description

Utilities to convert unicode to utf8 and vice versa.

## 8.128 oscl\_uuid.h File Reference

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32.

```
#include "oscl_base_macros.h"
#include "oscl_mem_basic_functions.h"
```

### Data Structures

- struct [OsclUuid](#)

### Defines

- #define [EMPTY\\_UUID](#) PVUuid(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
- #define [BYTES\\_IN\\_UUID\\_ARRAY](#) 8

### Typedefs

- typedef uint32 [OsclUid32](#)

#### 8.128.1 Detailed Description

This file defines the OSCL UUID structure used for unique identifiers as well as the short (32-bit) identifiers OsclUid32.

#### 8.128.2 Define Documentation

**8.128.2.1 #define BYTES\_IN\_UUID\_ARRAY 8**

**8.128.2.2 #define EMPTY\_UUID PVUuid(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)**

#### 8.128.3 Typedef Documentation

**8.128.3.1 typedef uint32 OsclUid32**

## 8.129 oscl\_uuid\_utils.h File Reference

```
#include "oscl_string_utils.h"
#include "oscl_stdstring.h"
```

### Variables

- const char **PV\_CHAR\_CLOSE\_BRACKET** = ')
- const char **PV\_CHAR\_COMMA** = ','

#### 8.129.1 Detailed Description

#### 8.129.2 Variable Documentation

**8.129.2.1 const char PV\_CHAR\_CLOSE\_BRACKET = ')**

**8.129.2.2 const char PV\_CHAR\_COMMA = ','**

## 8.130 oscl\_vector.h File Reference

The file [oscl\\_vector.h](#) defines the template class [Oscl\\_Vector](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

```
#include "oscl_base.h"
#include "oscl_mem_basic_functions.h"
#include "oscl_assert.h"
#include "oscl_opaque_type.h"
#include "oscl_defalloc.h"
```

### Data Structures

- class [Oscl\\_Vector\\_Base](#)
- class [Oscl\\_Vector< T, Alloc >](#)

#### 8.130.1 Detailed Description

The file [oscl\\_vector.h](#) defines the template class [Oscl\\_Vector](#) which has a very similar API as the STL Vector class (it basically provides a subset of the STL functionality). Memory allocation is abstracted through the use of an allocator template parameter.

## 8.131 osclconfig.h File Reference

This file contains configuration information for the linux platform.

```
#include <dirent.h>
#include <dlfcn.h>
#include <fcntl.h>
#include <sys/types.h>
#include <sys/stat.h>
#include "osclconfig_limits_typedefs.h"
#include "osclconfig_unix_android.h"
#include <stdlib.h>
#include <stdarg.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ctype.h>
#include <math.h>
#include "osclconfig_ix86.h"
#include "osclconfig_check.h"
```

### Defines

- #define OSCL\_HAS\_ANDROID\_SUPPORT 1
- #define OSCL\_HAS\_ANDROID\_FILE\_IO\_SUPPORT 1
- #define OSCL\_EXPORT\_REF \_\_attribute\_\_ ((visibility("default")))
- #define OSCL\_IMPORT\_REF \_\_attribute\_\_ ((visibility("default"))))
- #define OSCL\_RELEASE\_BUILD 0
- #define OSCL\_UNSIGNED\_CONST(x) x##u
- #define OSCL\_NATIVE\_UINT64\_TYPE u\_int64\_t
- #define OSCL\_TEMPLATED\_DESTRUCTOR\_CALL(type, simple\_type) ~type ()
- #define \_\_TFS\_\_ <>
- #define OSCL\_HAS\_PRAGMA\_PACK 0
- #define OSCL\_HAS\_PACKED\_STRUCT 1
- #define OSCL\_PACKED\_VAR(x) x \_\_attribute\_\_((packed))
- #define OSCL\_PACKED\_STRUCT\_BEGIN
- #define OSCL\_PACKED\_STRUCT\_END \_\_attribute\_\_((packed))
- #define OSCL\_ASSERT\_ALWAYS 0

### 8.131.1 Detailed Description

This file contains configuration information for the linux platform.

### 8.131.2 Define Documentation

- 8.131.2.1 `#define __TFS__ <>`
- 8.131.2.2 `#define OSCL_ASSERT_ALWAYS 0`
- 8.131.2.3 `#define OSCL_EXPORT_REF __attribute__ ((visibility("default")))`
- 8.131.2.4 `#define OSCL_HAS_ANDROID_FILE_IO_SUPPORT 1`
- 8.131.2.5 `#define OSCL_HAS_ANDROID_SUPPORT 1`
- 8.131.2.6 `#define OSCL_HAS_PACKED_STRUCT 1`
- 8.131.2.7 `#define OSCL_HAS_PRAGMA_PACK 0`
- 8.131.2.8 `#define OSCL_IMPORT_REF __attribute__ ((visibility("default")))`
- 8.131.2.9 `#define OSCL_NATIVE_UINT64_TYPE u_int64_t`
- 8.131.2.10 `#define OSCL_PACKED_STRUCT_BEGIN`
- 8.131.2.11 `#define OSCL_PACKED_STRUCT_END __attribute__((packed))`
- 8.131.2.12 `#define OSCL_PACKED_VAR(x) x __attribute__((packed))`
- 8.131.2.13 `#define OSCL_RELEASE_BUILD 0`
- 8.131.2.14 `#define OSCL_TEMPLATED_DESTRUCTOR_CALL(type, simple_type) ~type()`
- 8.131.2.15 `#define OSCL_UNSIGNED_CONST(x) x##u`

## 8.132 osclconfig\_ansi\_memory.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include <memory.h>
```

### Defines

- #define OSCL\_HAS\_ANSI\_MEMORY\_FUNCS 1

### Typedefs

- typedef size\_t oscl\_memsize\_t

#### 8.132.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header. This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

#### 8.132.2 Define Documentation

8.132.2.1 #define OSCL\_HAS\_ANSI\_MEMORY\_FUNCS 1

#### 8.132.3 Typedef Documentation

8.132.3.1 typedef size\_t oscl\_memsize\_t

## 8.133 osclconfig\_check.h File Reference

### Typedefs

- `typedef int8 __int8__check__`
- `typedef uint8 __uint8__check__`
- `typedef int16 __int16__check__`
- `typedef uint16 __uint16__check__`
- `typedef int32 __int32__check__`
- `typedef uint32 __uint32__check__`

## 8.134 osclconfig\_compiler\_warnings.h File Reference

This file contains the ability to turn off/on compiler warnings.

### Defines

- #define OSCL\_FUNCTION\_PTR(x) (&x)

#### 8.134.1 Detailed Description

This file contains the ability to turn off/on compiler warnings.

#### 8.134.2 Define Documentation

##### 8.134.2.1 #define OSCL\_FUNCTION\_PTR(x) (&x)

## 8.135 osclconfig\_error.h File Reference

This file contains the common typedefs and header files needed to compile osclerror.

```
#include "osclconfig.h"  
#include <dirent.h>  
#include <dlfcn.h>  
#include <fcntl.h>  
#include <sys/types.h>  
#include <sys/stat.h>  
#include "osclconfig_limits_typedefs.h"  
#include "osclconfig_unix_android.h"  
#include "osclconfig_ix86.h"  
#include "osclconfig_check.h"  
#include <setjmp.h>  
#include <errno.h>  
#include "osclconfig_error_check.h"
```

### Defines

- #define OSCL\_HAS\_EXCEPTIONS 1
- #define OSCL\_HAS\_ERRNO\_H 1
- #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0
- #define OSCL\_HAS\_SETJMP\_H 1

### 8.135.1 Detailed Description

This file contains the common typedefs and header files needed to compile osclerror.

### 8.135.2 Define Documentation

**8.135.2.1 #define OSCL\_HAS\_ERRNO\_H 1**

**8.135.2.2 #define OSCL\_HAS\_EXCEPTIONS 1**

**8.135.2.3 #define OSCL\_HAS\_SETJMP\_H 1**

**8.135.2.4 #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0**

## 8.136 osclconfig\_error\_check.h File Reference

## 8.137 osclconfig\_global\_new\_delete.h File Reference

### Functions

- void \* [operator new](#) (size\_t)
- void [operator delete](#) (void \*)

## 8.138 osclconfig\_global\_placement\_new.h File Reference

### Functions

- void \* [operator new](#) (size\_t, void \*ptr)

#### 8.138.1 Function Documentation

##### 8.138.1.1 void\* operator new (size\_t, void \*ptr) [inline]

## 8.139 osclconfig\_io.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include "osclconfig.h"
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <fcntl.h>
#include <signal.h>
#include <netdb.h>
#include <sys/mman.h>
#include <sys/types.h>
#include <errno.h>
#include <sys/vfs.h>
#include <dirent.h>
#include "osclconfig_io_check.h"
```

### Defines

- #define OSCL\_HAS\_GLOB 0
- #define OSCL\_HAS\_ANSI\_FILE\_IO\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_64BIT\_FILE\_IO\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_FILE\_IO\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION 0
- #define OSCL\_HAS\_NATIVE\_FILE\_CACHE\_ENABLE 1
- #define OSCL\_FILE\_BUFFER\_MAX\_SIZE 32768
- #define OSCL\_HAS\_PV\_FILE\_CACHE 0
- #define OSCL\_HAS\_LARGE\_FILE\_SUPPORT 1
- #define OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER 0
- #define OSCL\_HAS\_SYMBIAN\_DNS\_SERVER 0
- #define OSCL\_HAS\_BERKELEY\_SOCKETS 1
- #define OSCL\_HAS\_SOCKET\_SUPPORT 1
- #define OsclValidInetAddr(addr) (inet\_addr(addr)!=INADDR\_NONE)
- #define OsclMakeSockAddr(sockaddr, port, addrstr, ok)
- #define OsclUnMakeSockAddr(sockaddr, addrstr) addrstr=inet\_ntoa(sockaddr.sin\_addr);
- #define OsclMakeInAddr(in\_addr, addrstr, ok)
- #define OsclUnMakeInAddr(in\_addr, addrstr) addrstr=inet\_ntoa(in\_addr);
- #define OsclSetRecvBufferSize(s, val, ok, err)
- #define OsclBind(s, addr, ok, err)
- #define OsclSetSockOpt(s, optLevel, optName, optVal, optLen, ok, err)

- #define `OsclJoin`(s, addr, ok, err)
- #define `OsclListen`(s, size, ok, err)
- #define `OsclAccept`(s, accept\_s, ok, err, wouldblock)
- #define `OsclSetNonBlocking`(s, ok, err)
- #define `OsclShutdown`(s, how, ok, err)
- #define `OsclSocket`(s, fam, type, prot, ok, err)
- #define `OsclSendTo`(s, buf, len, addr, ok, err, nbytes, wouldblock)
- #define `OsclSend`(s, buf, len, ok, err, nbytes, wouldblock)
- #define `OsclCloseSocket`(s, ok, err)
- #define `OsclConnect`(s, addr, ok, err, wouldblock)
- #define `OsclGetPeerName`(s, name, namelen, ok, err)
- #define `OsclGetAsyncSockErr`(s, ok, err)
- #define `OsclPipe`(x) pipe(x)
- #define `OsclReadFD`(fd, buf, cnt) read(fd,buf,cnt)
- #define `OsclWriteFD`(fd, buf, cnt) write(fd,buf,cnt)
- #define `OsclConnectComplete`(s, wset, eset, success, fail, ok, err)
- #define `OsclRecv`(s, buf, len, ok, err, nbytes, wouldblock)
- #define `OsclRecvFrom`(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)
- #define `OsclSocketSelect`(nfds, rd, wr, ex, timeout, ok, err, nhandles)
- #define `OsclSocketStartup`(ok)
- #define `OsclSocketCleanup`(ok)
- #define `OsclGethostbyname`(name, hostent, ok, err)
- #define `OsclGetDottedAddr`(hostent, dottedaddr, ok)
- #define `OsclGetDottedAddrVector`(hostent, dottedaddr, dottedaddrvect, ok)
- #define `OSCL_SD_RECEIVE` SHUT\_RD
- #define `OSCL_SD_SEND` SHUT\_WR
- #define `OSCL_SD_BOTH` SHUT\_RDWR
- #define `OSCL_AF_INET` AF\_INET
- #define `OSCL SOCK_STREAM` SOCK\_STREAM
- #define `OSCL SOCK_DGRAM` SOCK\_DGRAM
- #define `OSCL IPPROTO_IP` IPPROTO\_IP
- #define `OSCL IPPROTO_TCP` IPPROTO\_TCP
- #define `OSCL IPPROTO_UDP` IPPROTO\_UDP
- #define `OSCL SOL_SOCKET` SOL\_SOCKET
- #define `OSCL SOL_IP` IPPROTO\_IP
- #define `OSCL SOL_TCP` IPPROTO\_TCP
- #define `OSCL SOL_UDP` IPPROTO\_UDP
- #define `OSCL_SOCKOPT_IP_MULTICAST_TTL` IP\_MULTICAST\_TTL
- #define `OSCL_SOCKOPT_IP_ADDMEMBERSHIP` IP\_ADD\_MEMBERSHIP
- #define `OSCL_SOCKOPT_IP_TOS` IP\_TOS
- #define `OSCL_SOCKOPT_SOL_REUSEADDR` SO\_REUSEADDR
- #define `MAX_TOSCLFILEOFFSET_VALUE` 0x7FFFFFFFFFFFFFFFLL;

## Typedefs

- typedef int `TOsclSocket`
- typedef struct sockaddr\_in `TOsclSockAddr`
- typedef socklen\_t `TOsclSockAddrLen`
- typedef struct ip\_mreq `TIpMReq`
- typedef struct hostent `TOsclHostent`
- typedef off64\_t `TOsclFileOffset`

### 8.139.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header. This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.



### 8.139.2 Define Documentation

```
8.139.2.1 #define MAX_TOSCLFILEOFFSET_VALUE 0x7FFFFFFFFFFFFFLL;  
8.139.2.2 #define OSCL_AF_INET AF_INET  
8.139.2.3 #define OSCL_FILE_BUFFER_MAX_SIZE 32768  
8.139.2.4 #define OSCL_HAS_ANSI_64BIT_FILE_IO_SUPPORT 0  
8.139.2.5 #define OSCL_HAS_ANSI_FILE_IO_SUPPORT 1  
8.139.2.6 #define OSCL_HAS_BERKELEY_SOCKETS 1  
8.139.2.7 #define OSCL_HAS_GLOB 0  
8.139.2.8 #define OSCL_HAS_LARGE_FILE_SUPPORT 1  
8.139.2.9 #define OSCL_HAS_MSWIN_FILE_IO_SUPPORT 0  
8.139.2.10 #define OSCL_HAS_NATIVE_FILE_CACHE_ENABLE 1  
8.139.2.11 #define OSCL_HAS_PV_FILE_CACHE 0  
8.139.2.12 #define OSCL_HAS_SOCKET_SUPPORT 1  
8.139.2.13 #define OSCL_HAS_SYMBIAN_COMPATIBLE_IO_FUNCTION 0  
8.139.2.14 #define OSCL_HAS_SYMBIAN_DNS_SERVER 0  
8.139.2.15 #define OSCL_HAS_SYMBIAN_SOCKET_SERVER 0  
8.139.2.16 #define OSCL IPPROTO_IP IPPROTO_IP  
8.139.2.17 #define OSCL IPPROTO_TCP IPPROTO_TCP  
8.139.2.18 #define OSCL IPPROTO_UDP IPPROTO_UDP  
8.139.2.19 #define OSCL_SD_BOTH SHUT_RDWR  
8.139.2.20 #define OSCL_SD_RECEIVE SHUT_RD  
8.139.2.21 #define OSCL_SD_SEND SHUT_WR  
8.139.2.22 #define OSCL_SOCK_DGRAM SOCK_DGRAM  
8.139.2.23 #define OSCL_SOCK_STREAM SOCK_STREAM  
8.139.2.24 #define OSCL_SOCKOPT_IP_ADDMEMBERSHIP IP_ADD_MEMBERSHIP  
8.139.2.25 #define OSCL_SOCKOPT_IP_MULTICAST_TTL IP_MULTICAST_TTL  
8.139.2.26 #define OSCL_SOCKOPT_IP_TOS IP_TOS  
8.139.2.27 #define OSCL_SOCKOPT_SOL_REUSEADDR SO_REUSEADDR          OSCL API

---

8.139.2.28 #define OSCL_SOL_IP IPPROTO_IP  
8.139.2.29 #define OSCL_SOL_SOCKET SOL_SOCKET  
8.139.2.30 #define OSCL_SOL_TCP IPPROTO_TCP
```

```
accept_s=accept(s,NULL,NULL); \
ok=(accept_s!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

**8.139.2.33 #define OsclBind(s, addr, ok, err)****Value:**

```
TOsclSockAddr* tmpadr = &addr; \
sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr); \
ok=(bind(s,sadr,sizeof(addr))!=(-1)); \
if (!ok)err=errno
```

**8.139.2.34 #define OsclCloseSocket(s, ok, err)****Value:**

```
ok=(close(s)!=(-1)); \
if (!ok)err=errno
```

**8.139.2.35 #define OsclConnect(s, addr, ok, err, wouldblock)****Value:**

```
TOsclSockAddr* tmpadr = &addr; \
sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr); \
ok=(connect(s,sadr,sizeof(addr))!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EINPROGRESS);}
```

**8.139.2.36 #define OsclConnectComplete(s, wset, eset, success, fail, ok, err)****Value:**

```
success=fail=false; \
if (FD_ISSET(s,&eset)) \
{fail=true;OsclGetAsyncSockErr(s,ok,err);} \
else if (FD_ISSET(s,&wset)) \
{OsclGetAsyncSockErr(s,ok,err);if (ok && err==0)success=true;else fail=true;}
```

**8.139.2.37 #define OsclGetAsyncSockErr(s, ok, err)****Value:**

```
int opterr;socklen_t optlen(sizeof(opterr)); \
ok=(getsockopt(s,SOL_SOCKET,SO_ERROR,(void *)&opterr,&optlen)!=(-1)); \
if(ok)err=opterr;else err=errno;
```

**8.139.2.38 #define OsclGetDottedAddr(hostent, dottedaddr, ok)****Value:**

```
long *_hostaddr=(long*)hostent->h_addr_list[0];\
    struct in_addr _inaddr; \
    _inaddr.s_addr=_hostaddr; \
    dottedaddr=inet_ntoa(_inaddr); \
    ok=(dottedaddr!=NULL);
```

**8.139.2.39 #define OsclGetDottedAddrVector(hostent, dottedaddr, dottedaddrvect, ok)****Value:**

```
if(dottedaddrvect) \
{ \
    long **_addrlist=(long**)hostent->h_addr_list; \
    for(int i = 0; _addrlist[i] != NULL; i++) { \
        struct in_addr _inaddr; \
        _inaddr.s_addr=*_addrlist[i]; \
        OsclNetworkAddress addr(inet_ntoa(_inaddr), 0); \
        dottedaddrvect->push_back(addr); \
    } \
    if (!dottedaddrvect->empty()) \
        {dottedaddr->port = dottedaddrvect->front().port; dottedaddr->ipAddr.Set( \
            dottedaddrvect->front().ipAddr.Str());} \
    ok=(!dottedaddrvect->empty() && (((*dottedaddrvect)[0]).ipAddr.Str() != NULL) \
        ); \
    } \
    else \
{ \
    char *add; \
    OsclGetDottedAddr(hostent, add, ok); \
    if(ok) dottedaddr->ipAddr.Set(add); \
}
```

**8.139.2.40 #define OsclGethostbyname(name, hostent, ok, err)****Value:**

```
hostent=gethostbyname((const char*)name); \
ok=(hostent!=NULL); \
if (!ok)err=errno;
```

**8.139.2.41 #define OsclGetPeerName(s, name, namelen, ok, err)****Value:**

```
ok=(getpeername(s, (sockaddr*)&name, (socklen_t*)&namelen) != (-1)); \
if (!ok)err=errno
```

**8.139.2.42 #define OsclJoin(s, addr, ok, err)****Value:**

```
\\
    struct ip_mreq mreq; \
    void* p = &addr; \
ok=(bind(s,(sockaddr*)p,sizeof(addr))!=(-1)); \
mreq.imr_multiaddr.s_addr = addr.sin_addr.s_addr ; \
mreq.imr_interface.s_addr = htonl(INADDR_ANY); \
ok=(setsockopt(s, IPPROTO_IP, IP_ADD_MEMBERSHIP, &mreq, sizeof(struct ip_ \
mreq))!=(-1)); \
    if (!ok)err=errno; \
}
```

**8.139.2.43 #define OsclListen(s, size, ok, err)****Value:**

```
ok=(listen(iSocket,qSize)!=(-1)); \
    if (!ok)err=errno
```

**8.139.2.44 #define OsclMakeInAddr(in\_addr, addrstr, ok)****Value:**

```
int32 result = inet_aton((const char*)addrstr, &in_addr); \
ok=(result!=0);
```

**8.139.2.45 #define OsclMakeSockAddr(sockaddr, port, addrstr, ok)****Value:**

```
sockaddr.sin_family=OSCL_AF_INET; \
    sockaddr.sin_port=htons(port); \
    int32 result=inet_aton((const char*)addrstr,&sockaddr.sin_addr); \
ok=(result!=0);
```

**8.139.2.46 #define OsclPipe(x) pipe(x)****8.139.2.47 #define OsclReadFD(fd, buf, cnt) read(fd,buf,cnt)****8.139.2.48 #define OsclRecv(s, buf, len, ok, err, nbytes, wouldblock)****Value:**

```
nbytes=recv(s,(void *)(buf),(size_t)(len),0); \
ok=(nbytes!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN);}
```

**8.139.2.49 #define OsclRecvFrom(s, buf, len, paddr, paddrlen, ok, err, nbytes, wouldblock)****Value:**

```
\\
void* p=paddr; \
nbytes=recvfrom(s,(void*)(buf),(size_t)(len),0,(struct sockaddr*)p,paddrlen); \
ok=(nbytes!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN);}\
```

**8.139.2.50 #define OsclSend(s, buf, len, ok, err, nbytes, wouldblock)****Value:**

```
nbytes=send(s, (const void*) (buf), (size_t) (len), 0); \
ok=(nbytes!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

**8.139.2.51 #define OsclSendTo(s, buf, len, addr, ok, err, nbytes, wouldblock)****Value:**

```
TOsclSockAddr* tmpadr = &addr; \
sockaddr* sadr = OSCL_STATIC_CAST(sockaddr*, tmpadr); \
nbytes=sendto(s, (const void*) (buf), (size_t) (len), 0, sadr, (socklen_t) sizeof(addr)); \
ok=(nbytes!=(-1)); \
if (!ok){err=errno;wouldblock=(err==EAGAIN||err==EWOULDBLOCK);}
```

**8.139.2.52 #define OsclSetNonBlocking(s, ok, err)****Value:**

```
ok=(fcntl(s,F_SETFL,O_NONBLOCK)!=(-1)); \
if (!ok)err=errno
```

**8.139.2.53 #define OsclSetRecvBufferSize(s, val, ok, err)****Value:**

```
ok=(setsockopt(s,SOL_SOCKET,SO_RCVBUF,(char*)&val, sizeof(int)) !=-1); \
if (!ok)err=errno
```

**8.139.2.54 #define OsclSetSockOpt(s, optLevel, optName, optVal, optLen, ok, err)****Value:**

```
ok=(setsockopt(s,optLevel,optName,OSCL_STATIC_CAST(const char*,optVal),optLen) !=(-1)); \
if (!ok)err=errno
```

**8.139.2.55 #define OsclShutdown(s, how, ok, err)****Value:**

```
ok=(shutdown(iSocket,how)!=(-1)); \
if (!ok)err=errno
```

**8.139.2.56 #define OsclSocket(s, fam, type, prot, ok, err)****Value:**

```
s=socket(fam,type,prot); \
ok=(s!=(-1)); \
if (!ok)err=errno
```

**8.139.2.57 #define OsclSocketCleanup(ok)****Value:**

```
signal(SIGPIPE,SIG_DFL); \
ok=true
```

**8.139.2.58 #define OsclSocketSelect(nfds, rd, wr, ex, timeout, ok, err, nhandles)****Value:**

```
nhandles=select(nfds,&rd,&wr,&ex,&timeout); \
ok=(nhandles!=(-1)); \
if (!ok)err=errno
```

**8.139.2.59 #define OsclSocketStartup(ok)****Value:**

```
signal(SIGPIPE,SIG_IGN); \
ok=true
```

**8.139.2.60 #define OsclUnMakeInAddr(in\_addr, addrstr) addrstr=inet\_ntoa(in\_addr);****8.139.2.61 #define OsclUnMakeSockAddr(sockaddr, addrstr) addrstr=inet\_ntoa(sockaddr.sin\_addr);****8.139.2.62 #define OsclValidInetAddr(addr) (inet\_addr(addr)!=INADDR\_NONE)****8.139.2.63 #define OsclWriteFD(fd, buf, cnt) write(fd,buf,cnt)****8.139.3 Typedef Documentation****8.139.3.1 typedef struct ip\_mreq TIpMReq****8.139.3.2 typedef off64\_t TOsclFileOffset****8.139.3.3 typedef struct hostent TOsclHostent****8.139.3.4 typedef struct sockaddr\_in TOsclSockAddr****8.139.3.5 typedef socklen\_t TOsclSockAddrLen****8.139.3.6 typedef int TOsclSocket**

## 8.140 osclconfig\_io\_check.h File Reference

### Typedefs

- `typedef TOsclFileOffset __verify__TOsclFileOffset__defined__`

#### 8.140.1 Typedef Documentation

##### 8.140.1.1 `typedef TOsclFileOffset __verify__TOsclFileOffset__defined__`

OSCL\_HAS\_ANSI\_FILE\_IO\_SUPPORT macro should be set to 1 if the target platform supports the ANSI C file I/O functions (`fopen`, `fread`, etc). Otherwise it should be set to 0. OSCL\_HAS\_ANSI\_64BIT\_FILE\_IO\_SUPPORT macro should be set to 1 if the target platform supports the 64-bit ANSI C file I/O functions (`fopen`, `fread`, etc). Otherwise it should be set to 0. OSCL\_HAS\_MSWIN\_FILE\_IO\_SUPPORT macro should be set to 1 if the target platform supports the ANSI C file I/O functions (`fopen`, `fread`, etc). Otherwise it should be set to 0. OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION macro should be set to 1 if the target platform supports the Symbian file I/O functions (`RFile`, `RFs`). Otherwise it should be set to 0. On Symbian platforms only: OSCL\_HAS\_NATIVE\_DUPLICATE\_FILE\_HANDLE macro should be set to 1 if the target platform supports the Symbian file I/O function `RFile::Duplicate`. Otherwise it should be set to 0. OSCL\_HAS\_NATIVE\_FILE\_CACHE\_ENABLE macro should be set to 1 if the target platform includes native file cache capability. Otherwise it should be set to 0. OSCL\_HAS\_PV\_FILE\_CACHE macro should be set to 1 if the target platform includes PV file cache capability. Otherwise it should be set to 0. OSCL\_HAS\_LARGE\_FILE\_SUPPORT macro should be set to 1 if the target platform supports more than 32bit file I/O capability. Otherwise it should be set to 0. type `TOsclFileOffset` should be defined as the type used for file size and offsets on the target platform. Example: `typedef size_t TOsclFileOffset;`

## 8.141 osclconfig\_ix86.h File Reference

This file contains configuration information for the ix86 processor family.

### Defines

- #define OSCL\_INTEGERS\_WORD\_ALIGNED 1
- #define OSCL\_BYTE\_ORDER\_BIG\_ENDIAN 0
- #define OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN 1

### 8.141.1 Detailed Description

This file contains configuration information for the ix86 processor family.

### 8.141.2 Define Documentation

8.141.2.1 #define OSCL\_BYTE\_ORDER\_BIG\_ENDIAN 0

8.141.2.2 #define OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN 1

8.141.2.3 #define OSCL\_INTEGERS\_WORD\_ALIGNED 1

## 8.142 osclconfig\_lib.h File Reference

This file contains configuration information for the ANSI build.

```
#include "osclconfig_lib_check.h"
```

### Defines

- #define OSCL\_HAS\_RUNTIME\_LIB\_LOADING\_SUPPORT 1
- #define PV\_RUNTIME\_LIB\_FILENAME\_EXTENSION "so"
- #define PV\_DYNAMIC\_LOADING\_CONFIG\_FILE\_PATH "./"

### 8.142.1 Detailed Description

This file contains configuration information for the ANSI build.

### 8.142.2 Define Documentation

8.142.2.1 #define OSCL\_HAS\_RUNTIME\_LIB\_LOADING\_SUPPORT 1

8.142.2.2 #define PV\_DYNAMIC\_LOADING\_CONFIG\_FILE\_PATH "./"

8.142.2.3 #define PV\_RUNTIME\_LIB\_FILENAME\_EXTENSION "so"

## **8.143 osclconfig\_lib\_check.h File Reference**

## 8.144 osclconfig\_limits\_typedefs.h File Reference

This file contains common typedefs based on the ANSI C limits.h header.

```
#include <limits.h>
```

### Defines

- #define OSCL\_CHAR\_IS\_UNSIGNED 1
- #define OSCL\_CHAR\_IS\_SIGNED 0

#### 8.144.1 Detailed Description

This file contains common typedefs based on the ANSI C limits.h header. This header file should work for any ANSI C compiler to determine the proper native C types to use for OSCL integer types.

#### 8.144.2 Define Documentation

**8.144.2.1 #define OSCL\_CHAR\_IS\_SIGNED 0**

**8.144.2.2 #define OSCL\_CHAR\_IS\_UNSIGNED 1**

## 8.145 osclconfig\_memory.h File Reference

```
#include "osclconfig.h"
#include "osclconfig_ansi_memory.h"
#include "osclconfig_memory_check.h"
```

### Defines

- #define OSCL\_BYPASS\_MEMMGT 1
- #define OSCL\_HAS\_GLOBAL\_NEW\_DELETE 1
- #define PVMEM\_INST\_LEVEL 1
- #define OSCL\_HAS\_HEAP\_BASE\_SUPPORT 1
- #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0

#### 8.145.1 Define Documentation

8.145.1.1 #define OSCL\_BYPASS\_MEMMGT 1

8.145.1.2 #define OSCL\_HAS\_GLOBAL\_NEW\_DELETE 1

8.145.1.3 #define OSCL\_HAS\_HEAP\_BASE\_SUPPORT 1

8.145.1.4 #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0

8.145.1.5 #define PVMEM\_INST\_LEVEL 1

## 8.146 osclconfig\_memory\_check.h File Reference

## 8.147 osclconfig\_no\_os.h File Reference

### Defines

- #define OSCL\_HAS\_UNIX\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_HAS\_SAVAJE\_SUPPORT 0
- #define OSCL\_HAS\_PV\_C\_OS\_SUPPORT 0
- #define OSCL\_HAS\_ANDROID\_SUPPORT 0
- #define OSCL\_HAS\_IPHONE\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_ERRORTRAP 0
- #define OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS 0
- #define OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_FUNCS 0
- #define OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS 0
- #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 0
- #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- #define OSCL\_HAS\_SYMBIAN\_MATH 0
- #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 0
- #define OSCL\_HAS\_PTHREAD\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION 0
- #define OSCL\_HAS\_SAVAJE\_IO\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER 0
- #define OSCL\_HAS\_SYMBIAN\_DNS\_SERVER 0
- #define OSCL\_HAS\_BERKELEY\_SOCKETS 0

## 8.148 osclconfig\_proc.h File Reference

This file contains configuration information for the linux platform.

```
#include "osclconfig.h"  
#include "osclconfig_proc_unix_android.h"  
#include <pthread.h>  
#include <errno.h>  
#include <signal.h>  
#include "osclconfig_proc_check.h"
```

### 8.148.1 Detailed Description

This file contains configuration information for the linux platform.

## 8.149 osclconfig\_proc\_check.h File Reference

### Typedefs

- `typedef TOsclThreadId __verify__TOsclThreadId_defined__`
- `typedef TOsclThreadFuncRet __verify__TOsclThreadFuncRet_defined__`
- `typedef TOsclThreadFuncArg __verify__TOsclThreadFuncArg_defined__`
- `typedef TOsclThreadObject __verify__TOsclThreadObject_defined__`
- `typedef TOsclMutexObject __verify__TOsclMutexObject_defined__`
- `typedef TOsclSemaphoreObject __verify__TOsclSemaphoreObject_defined__`
- `typedef TOsclConditionObject __verify__TOsclConditionObject_defined__`

### 8.149.1 Typedef Documentation

#### 8.149.1.1 `typedef TOsclConditionObject __verify__TOsclConditionObject_defined__`

type `TOsclConditionObject` should be defined as the type used as a condition variable on the target platform.  
Example: `typedef pthread_cond_t TOsclConditionObject;`

Note: Condition variables are only used with certain semaphore implementations. If the semaphore implementation does not require a condition variable, then this type can be defined as 'int' as follows: `typedef int TOsclConditionObject; //not used`

#### 8.149.1.2 `typedef TOsclMutexObject __verify__TOsclMutexObject_defined__`

type `TOsclMutexObject` should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef pthread_mutex_t TOsclMutexObject;`

#### 8.149.1.3 `typedef TOsclSemaphoreObject __verify__TOsclSemaphoreObject_defined__`

type `TOsclSemaphoreObject` should be defined as the type used as a mutex object or handle on the target platform. Example: `typedef sem_t TOsclSemaphoreObject;`

#### 8.149.1.4 `typedef TOsclThreadFuncArg __verify__TOsclThreadFuncArg_defined__`

type `TOsclThreadFuncArg` should be defined as the type used as a thread function argument on the target platform. Example: `typedef LPVOID TOsclThreadFuncArg;`

#### 8.149.1.5 `typedef TOsclThreadFuncRet __verify__TOsclThreadFuncRet_defined__`

type `TOsclThreadFuncRet` should be defined as the type used as a thread function return value on the target platform. Example: `typedef DWORD TOsclThreadFuncRet;`

#### 8.149.1.6 `typedef TOsclThreadId __verify__TOsclThreadId_defined__`

`OSCL_HAS_THREAD_SUPPORT` macro should be set to 1 if the target platform supports threads. Otherwise it should be set to 0. `OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT` macro should be set to 1 if the target platform supports non-pre-emptive threads. Otherwise it should be set to 0. `OSCL_HAS_SYMBIAN_SCHEDULER` macro should be set to 1 if the target platform supports Symbian active object

scheduler. Otherwise it should be set to 0. OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT macro should be set to 1 if the target platform supports POSIX-compliant semaphores (semaphore.h) with advanced realtime features including sem\_timedwait. Otherwise it should be set to 0. OSCL\_HAS\_PTHREAD\_SUPPORT macro should be set to 1 if the target platform supports POSIX-compliant pthreads (pthread.h). Otherwise it should be set to 0. type TOsclThreadId should be defined as the type used as a thread ID on the target platform. Example: `typedef DWORD TOsclThreadId;`

#### **8.149.1.7 `typedef TOsclThreadObject __verify__TOsclThreadObject__defined__`**

OSCL\_THREAD\_DECL macro should be defined to the necessary function declaration modifiers for thread routines, or a null macro if no modifiers are needed. Example: define OSCL\_THREAD\_DECL WINAPI Example of a declaration of a thread routine called MyThreadMain using the Oscl definitions:

`static TOsclThreadFuncRet OSCL_THREAD_DECL MyThreadMain(TOsclThreadFuncArg arg);` type TOsclThreadObject should be defined as the type used as a thread object or handle on the target platform. Example: `typedef pthread_t TOsclThreadObject;`

## 8.150 osclconfig\_proc\_unix\_android.h File Reference

```
#include <pthread.h>
#include <errno.h>
#include <signal.h>
```

### Defines

- #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- #define OSCL\_HAS\_THREAD\_SUPPORT 1
- #define OSCL\_HAS\_NON\_PREEMPTIVE\_THREAD\_SUPPORT 0
- #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 0
- #define OSCL\_HAS\_PTHREAD\_SUPPORT 1
- #define OSCL\_THREAD\_DECL

### Typedefs

- typedef pthread\_t TOsclThreadId
- typedef void \* TOsclThreadFuncArg
- typedef void \* TOsclThreadFuncRet
- typedef pthread\_t TOsclThreadObject
- typedef pthread\_mutex\_t TOsclMutexObject
- typedef int TOsclSemaphoreObject
- typedef pthread\_cond\_t TOsclConditionObject

### 8.150.1 Define Documentation

8.150.1.1 `#define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0`

8.150.1.2 `#define OSCL_HAS_PTHREAD_SUPPORT 1`

8.150.1.3 `#define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 0`

8.150.1.4 `#define OSCL_HAS_SYMBIAN_SCHEDULER 0`

8.150.1.5 `#define OSCL_HAS_THREAD_SUPPORT 1`

8.150.1.6 `#define OSCL_THREAD_DECL`

### 8.150.2 Typedef Documentation

8.150.2.1 `typedef pthread_cond_t TOsclConditionObject`

8.150.2.2 `typedef pthread_mutex_t TOsclMutexObject`

8.150.2.3 `typedef int TOsclSemaphoreObject`

8.150.2.4 `typedef void* TOsclThreadFuncArg`

8.150.2.5 `typedef void* TOsclThreadFuncRet`

8.150.2.6 `typedef pthread_t TOsclThreadId`

8.150.2.7 `typedef pthread_t TOsclThreadObject`

## 8.151 osclconfig\_proc\_unix\_common.h File Reference

```
#include <time.h>
#include <semaphore.h>
#include <pthread.h>
#include <errno.h>
```

### Defines

- #define OSCL\_HAS\_SYMBIAN\_SCHEDULER 0
- #define OSCL\_HAS\_THREAD\_SUPPORT 1
- #define OSCL\_HAS\_NON\_PREEMPTIVE\_THREAD\_SUPPORT 0
- #define OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT 1
- #define OSCL\_HAS\_PTHREAD\_SUPPORT 1
- #define OSCL\_THREAD\_DECL

### Typedefs

- typedef pthread\_t TOsclThreadId
- typedef void \* TOsclThreadFuncArg
- typedef void \* TOsclThreadFuncRet
- typedef pthread\_t TOsclThreadObject
- typedef pthread\_mutex\_t TOsclMutexObject
- typedef sem\_t TOsclSemaphoreObject
- typedef pthread\_cond\_t TOsclConditionObject

### 8.151.1 Define Documentation

8.151.1.1 `#define OSCL_HAS_NON_PREEMPTIVE_THREAD_SUPPORT 0`

8.151.1.2 `#define OSCL_HAS_PTHREAD_SUPPORT 1`

8.151.1.3 `#define OSCL_HAS_SEM_TIMEDWAIT_SUPPORT 1`

8.151.1.4 `#define OSCL_HAS_SYMBIAN_SCHEDULER 0`

8.151.1.5 `#define OSCL_HAS_THREAD_SUPPORT 1`

8.151.1.6 `#define OSCL_THREAD_DECL`

### 8.151.2 Typedef Documentation

8.151.2.1 `typedef pthread_cond_t TOsclConditionObject`

8.151.2.2 `typedef pthread_mutex_t TOsclMutexObject`

8.151.2.3 `typedef sem_t TOsclSemaphoreObject`

8.151.2.4 `typedef void* TOsclThreadFuncArg`

8.151.2.5 `typedef void* TOsclThreadFuncRet`

8.151.2.6 `typedef pthread_t TOsclThreadId`

8.151.2.7 `typedef pthread_t TOsclThreadObject`

## 8.152 osclconfig\_time.h File Reference

```
#include "osclconfig.h"
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include "osclconfig_time_check.h"
```

### Defines

- #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 1

### TypeDefs

- typedef struct timeval OsclBasicTimeStruct
- typedef tm OsclBasicDateTimeStruct

#### 8.152.1 Define Documentation

##### 8.152.1.1 #define OSCL\_HAS\_UNIX\_TIME\_FUNCS 1

#### 8.152.2 Typedef Documentation

##### 8.152.2.1 typedef tm OsclBasicDateTimeStruct

##### 8.152.2.2 typedef struct timeval OsclBasicTimeStruct

## 8.153 osclconfig\_time\_check.h File Reference

### Typedefs

- `typedef OsclBasicTimeStruct __Validate__BasicTimeStruct__`
- `typedef OsclBasicDateTimeStruct __Validate__BasicTimeDateStruct__`

#### 8.153.1 Typedef Documentation

##### 8.153.1.1 `typedef OsclBasicDateTimeStruct __Validate__BasicTimeDateStruct__`

`OsclBasicDateTimeStruct` type should be defined to the platform-specific date + time type.

##### 8.153.1.2 `typedef OsclBasicTimeStruct __Validate__BasicTimeStruct__`

`OSCL_HAS_UNIX_TIME_FUNCS` macro should be set to 1 if the target platform supports unix time of day functions. Otherwise it should be set to 0. `OsclBasicTimeStruct` type should be defined to the platform-specific time of day type.

## 8.154 osclconfig\_unix\_android.h File Reference

```
#include <stdlib.h>
#include <stdarg.h>
#include <sys/types.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ctype.h>
#include <math.h>
```

### Defines

- #define OSCL\_DISABLE\_INLINES 0
- #define OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_MATH\_SUPPORT 1
- #define OSCL\_HAS\_GLOBAL\_VARIABLE\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_STRING\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_WIDE\_STRING\_SUPPORT 0
- #define OSCL\_HAS\_ANSI\_STDIO\_SUPPORT 1
- #define OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN 1
- #define OSCL\_HAS\_UNIX\_SUPPORT 1
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_HAS\_IPHONE\_SUPPORT 0
- #define OSCL\_NATIVE\_INT64\_TYPE int64\_t
- #define OSCL\_NATIVE\_UINT64\_TYPE uint64\_t
- #define INT64(x) x##LL
- #define UINT64(x) x##ULL
- #define INT64\_HILO(high, low) (((high##LL)<<32)|low)
- #define UINT64\_HILO(high, low) (((high##ULL)<<32)|low)
- #define OSCL\_HAS\_UNICODE\_SUPPORT 1
- #define OSCL\_NATIVE\_WCHAR\_TYPE wchar\_t
- #define \_STRLIT(x) x
- #define \_STRLIT\_CHAR(x) x
- #define \_STRLIT\_WCHAR(x) L ## x
- #define OSCL\_HAS\_TLS\_SUPPORT 1
- #define OSCL\_TLS\_IS\_KEYED 1
- #define OSCL\_TLS\_KEY\_CREATE\_FUNC(key) (pthread\_key\_create(&key,NULL)==0)
- #define OSCL\_TLS\_KEY\_DELETE\_FUNC(key) pthread\_key\_delete(key)
- #define OSCL\_TLS\_STORE\_FUNC(key, ptr) (pthread\_setspecific(key,(const void\*)ptr)==0)
- #define OSCL\_TLS\_GET\_FUNC(key) pthread\_getspecific(key)
- #define OSCL\_HAS\_BASIC\_LOCK 1

## Typedefs

- `typedef pthread_key_t TOsclTlsKey`
  
- `typedef pthread_mutex_t TOsclBasicLockObject`



### 8.154.1 Define Documentation

8.154.1.1 `#define _STRLIT(x) x`

8.154.1.2 `#define _STRLIT_CHAR(x) x`

8.154.1.3 `#define _STRLIT_WCHAR(x) L ## x`

8.154.1.4 `#define INT64(x) x##LL`

8.154.1.5 `#define INT64_HILO(high, low) (((high##LL))<<32)|low)`

8.154.1.6 `#define OSCL_DISABLE_INLINES 0`

8.154.1.7 `#define OSCL_HAS_ANSI_MATH_SUPPORT 1`

8.154.1.8 `#define OSCL_HAS_ANSI_STDIO_SUPPORT 1`

8.154.1.9 `#define OSCL_HAS_ANSI_STDLIB_SUPPORT 1`

8.154.1.10 `#define OSCL_HAS_ANSI_STRING_SUPPORT 1`

8.154.1.11 `#define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 0`

8.154.1.12 `#define OSCL_HAS_BASIC_LOCK 1`

8.154.1.13 `#define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1`

8.154.1.14 `#define OSCL_HAS_IPHONE_SUPPORT 0`

8.154.1.15 `#define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0`

8.154.1.16 `#define OSCL_HAS_MSWIN_SUPPORT 0`

8.154.1.17 `#define OSCL_HAS_SYMBIAN_SUPPORT 0`

8.154.1.18 `#define OSCL_HAS_TLS_SUPPORT 1`

8.154.1.19 `#define OSCL_HAS_UNICODE_SUPPORT 1`

8.154.1.20 `#define OSCL_HAS_UNIX_SUPPORT 1`

8.154.1.21 `#define OSCL_MEMFRAG_PTR_BEFORE_LEN 1`

8.154.1.22 `#define OSCL_NATIVE_INT64_TYPE int64_t`

8.154.1.23 `#define OSCL_NATIVE_UINT64_TYPE uint64_t`

8.154.1.24 `#define OSCL_NATIVE_WCHAR_TYPE wchar_t`

8.154.1.25 `#define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)`

8.154.1.26 `#define OSCL_TLS_IS_KEYED 1`

8.154.1.27 `#define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)` OSCL API

8.154.1.28 `#define OSCL_TLS_KEY_DELETE_FUNC(key) pthread_key_delete(key)`

8.154.1.29 `#define OSCL_TLS_STORE_FUNC(key, ptr) (pthread_setspecific(key,(const void*)ptr)==0)`

## 8.155 osclconfig\_unix\_common.h File Reference

```
#include <stdlib.h>
#include <stdarg.h>
#include <sys/types.h>
#include <stdio.h>
#include <wchar.h>
#include <string.h>
#include <unistd.h>
#include <pthread.h>
#include <ctype.h>
#include <math.h>
```

### Defines

- #define OSCL\_DISABLE\_INLINES 0
- #define OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_MATH\_SUPPORT 1
- #define OSCL\_HAS\_GLOBAL\_VARIABLE\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_STRING\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_WIDE\_STRING\_SUPPORT 1
- #define OSCL\_HAS\_ANSI\_STDIO\_SUPPORT 1
- #define OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN 1
- #define OSCL\_HAS\_UNIX\_SUPPORT 1
- #define OSCL\_HAS\_MSWIN\_SUPPORT 0
- #define OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT 0
- #define OSCL\_HAS\_SYMBIAN\_SUPPORT 0
- #define OSCL\_NATIVE\_INT64\_TYPE int64\_t
- #define OSCL\_NATIVE\_UINT64\_TYPE uint64\_t
- #define INT64(x) x##LL
- #define UINT64(x) x##ULL
- #define INT64\_HILO(high, low) (((high##LL))<<32)|low)
- #define UINT64\_HILO(high, low) (((high##ULL))<<32)|low)
- #define OSCL\_HAS\_UNICODE\_SUPPORT 1
- #define OSCL\_NATIVE\_WCHAR\_TYPE wchar\_t
- #define \_STRLIT(x) L ## x
- #define \_STRLIT\_CHAR(x) x
- #define \_STRLIT\_WCHAR(x) L ## x
- #define OSCL\_HAS\_TLS\_SUPPORT 1
- #define OSCL\_TLS\_IS\_KEYED 1
- #define OSCL\_TLS\_KEY\_CREATE\_FUNC(key) (pthread\_key\_create(&key,NULL)==0)
- #define OSCL\_TLS\_KEY\_DELETE\_FUNC(key) pthread\_key\_delete(key)
- #define OSCL\_TLS\_STORE\_FUNC(key, ptr) (pthread\_setspecific(key,(const void\*)ptr)==0)
- #define OSCL\_TLS\_GET\_FUNC(key) pthread\_getspecific(key)
- #define OSCL\_HAS\_BASIC\_LOCK 1

## Typedefs

- `typedef pthread_key_t TOsclTlsKey`
  
- `typedef pthread_mutex_t TOsclBasicLockObject`



### 8.155.1 Define Documentation

- 8.155.1.1 `#define _STRLIT(x) L ## x`
- 8.155.1.2 `#define _STRLIT_CHAR(x) x`
- 8.155.1.3 `#define _STRLIT_WCHAR(x) L ## x`
- 8.155.1.4 `#define INT64(x) x##LL`
- 8.155.1.5 `#define INT64_HILO(high, low) (((high##LL))<<32)|low)`
- 8.155.1.6 `#define OSCL_DISABLE_INLINES 0`
- 8.155.1.7 `#define OSCL_HAS_ANSI_MATH_SUPPORT 1`
- 8.155.1.8 `#define OSCL_HAS_ANSI_STDIO_SUPPORT 1`
- 8.155.1.9 `#define OSCL_HAS_ANSI_STDLIB_SUPPORT 1`
- 8.155.1.10 `#define OSCL_HAS_ANSI_STRING_SUPPORT 1`
- 8.155.1.11 `#define OSCL_HAS_ANSI_WIDE_STRING_SUPPORT 1`
- 8.155.1.12 `#define OSCL_HAS_BASIC_LOCK 1`
- 8.155.1.13 `#define OSCL_HAS_GLOBAL_VARIABLE_SUPPORT 1`
- 8.155.1.14 `#define OSCL_HAS_MSWIN_PARTIAL_SUPPORT 0`
- 8.155.1.15 `#define OSCL_HAS_MSWIN_SUPPORT 0`
- 8.155.1.16 `#define OSCL_HAS_SYMBIAN_SUPPORT 0`
- 8.155.1.17 `#define OSCL_HAS_TLS_SUPPORT 1`
- 8.155.1.18 `#define OSCL_HAS_UNICODE_SUPPORT 1`
- 8.155.1.19 `#define OSCL_HAS_UNIX_SUPPORT 1`
- 8.155.1.20 `#define OSCL_MEMFRAG_PTR_BEFORE_LEN 1`
- 8.155.1.21 `#define OSCL_NATIVE_INT64_TYPE int64_t`
- 8.155.1.22 `#define OSCL_NATIVE_UINT64_TYPE uint64_t`
- 8.155.1.23 `#define OSCL_NATIVE_WCHAR_TYPE wchar_t`
- 8.155.1.24 `#define OSCL_TLS_GET_FUNC(key) pthread_getspecific(key)`
- 8.155.1.25 `#define OSCL_TLS_IS_KEYED 1`
- 8.155.1.26 `#define OSCL_TLS_KEY_CREATE_FUNC(key) (pthread_key_create(&key,NULL)==0)`
- 8.155.1.27 `#define OSCL_TLS_KEY_DELETE_FUNC(key) pthread_key_delete(key)`
- 8.155.1.28 `#define OSCL_TLS_STORE_FUNC(key, ptr) (pthread_setspecific(key,(const void*)ptr)==0)`
- 8.155.1.29 `#define UINT64(x) x##ULL`

## 8.156 osclconfig\_util.h File Reference

```
#include "osclconfig.h"
#include <stdio.h>
#include <time.h>
#include <sys/time.h>
#include <unistd.h>
#include "osclconfig_util_check.h"
```

### Defines

- #define OSCL\_CLOCK\_HAS\_DRIFT\_CORRECTION 0
- #define OSCL\_HAS\_SYMBIAN\_TIMERS 0
- #define OSCL\_HAS\_SYMBIAN\_MATH 0
- #define OSCL RAND\_MAX RAND\_MAX
- #define OSCL\_HAS\_SNPRINTF\_LONGLONG\_SUPPORT 1
- #define SLEEP\_ONE\_SEC sleep(1)

#### 8.156.1 Define Documentation

8.156.1.1 #define OSCL\_CLOCK\_HAS\_DRIFT\_CORRECTION 0

8.156.1.2 #define OSCL\_HAS\_SNPRINTF\_LONGLONG\_SUPPORT 1

8.156.1.3 #define OSCL\_HAS\_SYMBIAN\_MATH 0

8.156.1.4 #define OSCL\_HAS\_SYMBIAN\_TIMERS 0

8.156.1.5 #define OSCL RAND\_MAX RAND\_MAX

8.156.1.6 #define SLEEP\_ONE\_SEC sleep(1)

## 8.157 osclconfig\_util\_check.h File Reference

## 8.158 pvlogger.h File Reference

This file contains basic logger interfaces for common use across platforms.

```
#include "oscl_base.h"
#include "oscl_vector.h"
#include "oscl_defalloc.h"
#include "oscl_shared_ptr.h"
#include "oscl_refcounter.h"
#include "osclconfig_compiler_warnings.h"
#include "osclconfig.h"
#include "osclconfig_memory.h"
```

### Data Structures

- class [PVLogger](#)

### Defines

- #define PVLOGMSG\_INST\_REL 0
- #define PVLOGMSG\_INST\_PROF 1
- #define PVLOGMSG\_INST\_HLDBG 2
- #define PVLOGMSG\_INST\_MLDBG 3
- #define PVLOGMSG\_INST\_LLDBG 4
- #define PVLOGGER\_INST\_LEVEL 5
- #define \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- #define \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- #define PVLOGGER\_INST\_LEVEL\_SUPPORT 1
- #define PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_-  
PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_-  
PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_-  
PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_-  
PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- #define PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_-  
PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_-  
PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_-  
PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_-  
PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- #define PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_HLDBG(LOGGER, LEVEL, MESSAGE) \_-  
PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)

- #define **PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_HLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_HLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_HLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_V\_INST\_MLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_LLDBG**(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG(IL, LOGGER, LEVEL, MESSAGE)** PVLOGGER\_LOGMSG\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGMSG\_V(IL, LOGGER, LEVEL, MESSAGE)** PVLOGGER\_LOGMSG\_V\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN(IL, LOGGER, LEVEL, MESSAGE)** PVLOGGER\_LOGBIN\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOGBIN\_V(IL, LOGGER, LEVEL, MESSAGE)** PVLOGGER\_LOGBIN\_V\_## IL (LOGGER, LEVEL, MESSAGE)
- #define **PVLOGGER\_LOG\_USE\_ONLY(x)** x
- #define **PVLOGGER\_ENABLE** 1

## Variables

- const int32 **PVLOGGER\_LEVEL\_UNINITIALIZED** = -1
- const **PVLogger::log\_level\_type PVLOGMSG\_EMERG** = 0
- const **PVLogger::log\_level\_type PVLOGMSG\_ALERT** = 1
- const **PVLogger::log\_level\_type PVLOGMSG\_CRIT** = 2
- const **PVLogger::log\_level\_type PVLOGMSG\_ERR** = 3
- const **PVLogger::log\_level\_type PVLOGMSG\_WARNING** = 4
- const **PVLogger::log\_level\_type PVLOGMSG\_NOTICE** = 5
- const **PVLogger::log\_level\_type PVLOGMSG\_INFO** = 6
- const **PVLogger::log\_level\_type PVLOGMSG\_STACK\_TRACE** = 7
- const **PVLogger::log\_level\_type PVLOGMSG\_DEBUG** = 8
- const **PVLogger::log\_level\_type PVLOGMSG\_FATAL\_ERROR** = **PVLOGMSG\_EMERG**
- const **PVLogger::log\_level\_type PVLOGMSG\_NONFATAL\_ERROR** = **PVLOGMSG\_ERR**
- const **PVLogger::log\_level\_type PVLOGMSG\_STATISTIC** = **PVLOGMSG\_INFO**
- const **PVLogger::log\_level\_type PVLOGMSG\_VERBOSE** = **PVLOGMSG\_DEBUG**

### 8.158.1 Detailed Description

This file contains basic logger interfaces for common use across platforms. This is the main entry point header file for the logger library. It should be the only one users directly include.

### 8.158.2 Define Documentation

#### 8.158.2.1 #define \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
    if (LOGGER) \
{\
    if (LOGGER->IsActive(LEVEL)) \
{\
    LOGGER->LogMsgBuffers MESSAGE; \
}\
}\
}
```

#### 8.158.2.2 #define \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
    if (LOGGER) \
{\
    if (LOGGER->IsActive(LEVEL)) \
{\
    LOGGER->LogMsgBuffersV MESSAGE; \
}\
}\
}
```

#### 8.158.2.3 #define \_PVLOGGER\_LOGMSG(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
    if (LOGGER) \
{\
    if (LOGGER->IsActive(LEVEL)) \
{\
    LOGGER->LogMsgString MESSAGE; \
}\
}\
}
```

#### 8.158.2.4 #define \_PVLOGGER\_LOGMSG\_V(LOGGER, LEVEL, MESSAGE)

**Value:**

```
{\
    if (LOGGER) \

```

```

{ \
    if (LOGGER->IsActive(LEVEL)) \
    { \
        LOGGER->LogMsgStringV MESSAGE; \
    } \
}

```

#### 8.158.2.5 #define PVLOGGER\_ENABLE 1

In case logging is compiled out, there is no need to compile the logger runtime code either.

#### 8.158.2.6 #define PVLOGGER\_INST\_LEVEL 5

#### 8.158.2.7 #define PVLOGGER\_INST\_LEVEL\_SUPPORT 1

#### 8.158.2.8 #define PVLOGGER\_LOG\_USE\_ONLY(x) x

Used to compile in/out lines of code that are used only for [PVLogger](#) macros.

This code will be removed at compile time when [PVLogger](#) is disabled, i.e. Release mode. So do not put in any code that is necessary for correct functionality of the module

#### 8.158.2.9 #define PVLOGGER\_LOGBIN(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGBIN\_## IL (LOGGER, LEVEL, MESSAGE)

This is a binary API to log messages

##### Parameters

**IL** Instrumentation level.

**LOGGER** Pointer to the logger object, that acts as the logging control/interface point

**LEVEL** Log level of the message

**MESSAGE** Log Message which includes the message id, and message buffers that need to be logged.

Example Usage: PVLOGGER\_LOGBIN (PVLOGMSG\_INST\_LLDBG, logger\_1, PVLOGMSG\_WARNING, (10, 3, msgBuf1Size, msgBuf1, msgBuf2Size, msgBuf2, msgBuf3Size, msgBuf3));

-This message contains THREE (ptr\_len, ptr) pairs. Log level of this msg is PVLOGMSG\_WARNING, message id is 10.

- 8.158.2.10 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_HLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.11 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_LLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.12 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_MLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.13 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.14 #define PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN(LOGGER, LEVEL, MESSAGE)
- 8.158.2.15 #define PVLOGGER\_LOGBIN\_V(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGBIN\_V\_ ## IL (LOGGER, LEVEL, MESSAGE)
- 8.158.2.16 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_HLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.17 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_LLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.18 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_PROF(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.19 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_REL(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.20 #define PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_V\_INST\_MLDBG(LOGGER, LEVEL, MESSAGE) \_PVLOGGER\_LOGBIN\_V(LOGGER, LEVEL, MESSAGE)
- 8.158.2.21 #define PVLOGGER\_LOGMSG(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER\_LOGMSG\_ ## IL (LOGGER, LEVEL, MESSAGE)

This is the text based API to log messages

### Parameters

**IL** Instrumentation level.

**LOGGER** Pointer to the logger object, that acts as the logging control/interface point

**LEVEL** Log level of the message

**MESSAGE** Log Message which includes the message id, and any kind of formatting information

Example Usage: PVLOGGER\_LOGMSG(PVLOGMSG\_INST\_LLDBG, logger\_1, PVLOGMSG\_WARNING, (13, "Test Messsage to Node 1\n")); -This message of log level PVLOGMSG\_WARNING, and has a message id of 13

- 8.158.2.22 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 8.158.2.23 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 8.158.2.24 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 8.158.2.25 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 8.158.2.26 `#define PVLOGGER_LOGMSG_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG(LOGGER, LEVEL, MESSAGE)`
- 8.158.2.27 `#define PVLOGGER_LOGMSG_V(IL, LOGGER, LEVEL, MESSAGE) PVLOGGER_LOGMSG_V_## IL (LOGGER, LEVEL, MESSAGE)`
- 8.158.2.28 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_HLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 8.158.2.29 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_LLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 8.158.2.30 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_MLDBG(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 8.158.2.31 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_PROF(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 8.158.2.32 `#define PVLOGGER_LOGMSG_V_PVLOGMSG_INST_REL(LOGGER, LEVEL, MESSAGE) _PVLOGGER_LOGMSG_V(LOGGER, LEVEL, MESSAGE)`
- 8.158.2.33 `#define PVLOGMSG_INST_HLDBG 2`

#### High Level Debug Layer

This layer should contain messages that have very minimal impact on performance, but are at lower level (i.e., provide more information) than would be appropriate in a shipping product. The messages are probably used to gather information and validate proper functionality at a high level as might be appropriate for IOT, stress testing, or QA testing.

#### 8.158.2.34 `#define PVLOGMSG_INST_LLDBG 4`

#### Low Level Debug Layer

This layer should contain messages for early functional testing. The messages are typically at a very low-level and allow testing the functionality of individual modules and components. Messages at this layer will typically have a performance impact (sometimes significant) due to the fact that they are at such a low level.

**8.158.2.35 #define PVLOGMSG\_INST\_MLDBG 3**

Mid Level Debug Layer

This layer should contain messages that are useful in the middle stages of the development cycle where major components are being integrated. The components themselves should already be well-tested so the emphasis is on interfaces between these components and integration testing. Messages at this layer may have some performance impact.

**8.158.2.36 #define PVLOGMSG\_INST\_PROF 1**

Profile Layer

The profile layer is used for messages and information related to measuring and reporting performance-related information.

**8.158.2.37 #define PVLOGMSG\_INST\_REL 0**

Release Layer

The release layer should only be used for messages that should remain in the final release. In certain cases all messaging may be disabled depending on customer requirements. However, when allowed the release layer should contain information that will be useful diagnosing problems in a released product (perhaps after entering a diagnostic mode), but with absolutely minimal performance impact when disabled at runtime.

**8.158.3 Variable Documentation****8.158.3.1 const int32 PVLOGGER\_LEVEL\_UNINITIALIZED = -1****8.158.3.2 const PVLogger::log\_level\_type PVLOGMSG\_ALERT = 1**

action must be taken immediately

**8.158.3.3 const PVLogger::log\_level\_type PVLOGMSG\_CRIT = 2**

critical conditions

**8.158.3.4 const PVLogger::log\_level\_type PVLOGMSG\_DEBUG = 8**

debug-level messages

**8.158.3.5 const PVLogger::log\_level\_type PVLOGMSG\_EMERG = 0**

system is unusable

**8.158.3.6 const PVLogger::log\_level\_type PVLOGMSG\_ERR = 3**

error conditions

**8.158.3.7 const PVLogger::log\_level\_type PVLOGMSG\_FATAL\_ERROR = PVLOGMSG\_EMERG**

**8.158.3.8 const PVLogger::log\_level\_type PVLOGMSG\_INFO = 6**

informational

**8.158.3.9 const PVLogger::log\_level\_type PVLOGMSG\_NONFATAL\_ERROR = PVLOGMSG\_ERR**

**8.158.3.10 const PVLogger::log\_level\_type PVLOGMSG\_NOTICE = 5**

normal but significant condition

**8.158.3.11 const PVLogger::log\_level\_type PVLOGMSG\_STACK\_TRACE = 7**

function enter and exit

**8.158.3.12 const PVLogger::log\_level\_type PVLOGMSG\_STATISTIC = PVLOGMSG\_INFO**

**8.158.3.13 const PVLogger::log\_level\_type PVLOGMSG\_VERBOSE = PVLOGMSG\_DEBUG**

**8.158.3.14 const PVLogger::log\_level\_type PVLOGMSG\_WARNING = 4**

warning conditions

## 8.159 pvlogger\_accessories.h File Reference

```
#include "oscl_base.h"
#include "pvlogger.h"
#include "oscl_vector.h"
#include "oscl_defalloc.h"
#include "oscl_shared_ptr.h"
#include "oscl_base_alloc.h"
```

### Data Structures

- class [PVLoggerLayout](#)
- class [PVLoggerFilter](#)
- class [AllPassFilter](#)
- class [PVLoggerAppender](#)

### Variables

- const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_ACCEPT = 1](#)
- const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_REJECT = 2](#)
- const [PVLoggerFilter::filter\\_status\\_type PVLOGGER\\_FILTER\\_NEUTRAL = 3](#)

### 8.159.1 Variable Documentation

#### 8.159.1.1 const PVLoggerFilter::filter\_status\_type PVLOGGER\_FILTER\_ACCEPT = 1

Referenced by AllPassFilter::FilterOpaqueMessage(), and AllPassFilter::FilterString().

#### 8.159.1.2 const PVLoggerFilter::filter\_status\_type PVLOGGER\_FILTER\_NEUTRAL = 3

#### 8.159.1.3 const PVLoggerFilter::filter\_status\_type PVLOGGER\_FILTER\_REJECT = 2

## 8.160 pvlogger\_c.h File Reference

This file contains basic logger interfaces for common use across platforms. C-callable version.

```
#include "osclconfig.h"
```

### Defines

- #define PVLOGGER\_C\_INST\_LEVEL 5
- #define PVLOGMSG\_C\_INST\_REL 0
- #define PVLOGMSG\_C\_INST\_PROF 1
- #define PVLOGMSG\_C\_INST\_HLDBG 2
- #define PVLOGMSG\_C\_INST\_MLDBG 3
- #define PVLOGMSG\_C\_INST\_LLDBG 4
- #define PVLOGMSG\_C\_EMERG 0
- #define PVLOGMSG\_C\_ALERT 1
- #define PVLOGMSG\_C\_CRIT 2
- #define PVLOGMSG\_C\_ERR 3
- #define PVLOGMSG\_C\_WARNING 4
- #define PVLOGMSG\_C\_NOTICE 5
- #define PVLOGMSG\_C\_INFO 6
- #define PVLOGMSG\_C\_STACK\_TRACE 7
- #define PVLOGMSG\_C\_STACK\_DEBUG 8

### Functions

- OSCL\_IMPORT\_REF void \* [pvLogger\\_GetLoggerObject](#) (const char \*tag)
- OSCL\_IMPORT\_REF int [pvLogger\\_IsActive](#) (void \*logger, int log\_level)
- OSCL\_IMPORT\_REF void [pvLogger\\_LogMsgString](#) (void \*logger, int msgID, const char \*fmt,...)

### 8.160.1 Detailed Description

This file contains basic logger interfaces for common use across platforms. C-callable version. This is the main entry point header file for the logger library. It should be the only one users directly include.

### 8.160.2 Define Documentation

- 8.160.2.1 `#define PVLOGGER_C_INST_LEVEL 5`
- 8.160.2.2 `#define PVLOGMSG_C_ALERT 1`
- 8.160.2.3 `#define PVLOGMSG_C_CRIT 2`
- 8.160.2.4 `#define PVLOGMSG_C_EMERG 0`
- 8.160.2.5 `#define PVLOGMSG_C_ERR 3`
- 8.160.2.6 `#define PVLOGMSG_C_INFO 6`
- 8.160.2.7 `#define PVLOGMSG_C_INST_HLDBG 2`
- 8.160.2.8 `#define PVLOGMSG_C_INST_LLDBG 4`
- 8.160.2.9 `#define PVLOGMSG_C_INST_MLDBG 3`
- 8.160.2.10 `#define PVLOGMSG_C_INST_PROF 1`
- 8.160.2.11 `#define PVLOGMSG_C_INST_REL 0`
- 8.160.2.12 `#define PVLOGMSG_C_NOTICE 5`
- 8.160.2.13 `#define PVLOGMSG_C_STACK_DEBUG 8`
- 8.160.2.14 `#define PVLOGMSG_C_STACK_TRACE 7`
- 8.160.2.15 `#define PVLOGMSG_C_WARNING 4`

### 8.160.3 Function Documentation

- 8.160.3.1 `OSCL_IMPORT_REF void* pvLogger_GetLoggerObject (const char * tag)`
- 8.160.3.2 `OSCL_IMPORT_REF int pvLogger_IsActive (void * logger, int log_level)`
- 8.160.3.3 `OSCL_IMPORT_REF void pvLogger_LogMsgString (void * logger, int msgID, const char * fmt, ...)`

## 8.161 pvlogger\_registry.h File Reference

```
#include "pvlogger.h"
#include "oscl_tagtree.h"
#include "oscl_base.h"
#include "oscl_map.h"
#include "oscl_vector.h"
#include "oscl_stdstring.h"
#include "osclconfig_compiler_warnings.h"
```

### Data Structures

- class [PVLoggerRegistry](#)

# Index

~AllPassFilter  
    AllPassFilter, 142  
~BufFragGroup  
    BufFragGroup, 148  
~BufferMgr  
    BufferMgr, 145  
~CallbackTimer  
    CallbackTimer, 152  
~CallbackTimerObserver  
    CallbackTimerObserver, 154  
~DNSRequestParam  
    DNSRequestParam, 166  
~GetHostByNameParam  
    GetHostByNameParam, 169  
~HeapBase  
    HeapBase, 171  
~MM\_AllocInfo  
    MM\_AllocInfo, 189  
~MM\_AllocNode  
    MM\_AllocNode, 191  
~MediaData  
    MediaData, 179  
~MemAllocator  
    MemAllocator, 183  
~OSCLMemAutoPtr  
    OSCLMemAutoPtr, 462  
~OSCL\_FastString  
    OSCL\_FastString, 215  
~OSCL\_HeapString  
    osclutil, 91  
~OSCL\_HeapStringA  
    OSCL\_HeapStringA, 237  
~OSCL\_StackString  
    osclutil, 91  
~OSCL\_String  
    OSCL\_String, 298  
~OSCL\_wFastString  
    OSCL\_wFastString, 326  
~OSCL\_wHeapString  
    osclutil, 91  
~OSCL\_wHeapStringA  
    OSCL\_wHeapStringA, 331  
~OSCL\_wStackString  
    osclutil, 91  
~OSCL\_wString  
    OSCL\_wString, 336  
~OsclAcceptMethod  
    OsclAcceptMethod, 339  
~OsclActiveObject  
    OsclActiveObject, 343  
~OsclAllocDestructDealloc  
    OsclAllocDestructDealloc, 346  
~OsclAsyncFile  
    OsclAsyncFile, 349  
~OsclAsyncFileBuffer  
    OsclAsyncFileBuffer, 352  
~OsclBinIStream  
    OsclBinIStream, 356  
~OsclBinOStream  
    OsclBinOStream, 363  
~OsclBindMethod  
    OsclBindMethod, 354  
~OsclCacheObserver  
    Oscl\_File::OsclCacheObserver, 374  
~OsclComponentRegistry  
    OsclComponentRegistry, 377  
~OsclComponentRegistryElement  
    OsclComponentRegistryElement, 379  
~OsclConnectMethod  
    OsclConnectMethod, 381  
~OsclDNS  
    osclio, 128  
~OsclDNSI  
    OsclDNSI, 387  
~OsclDNSIBase  
    OsclDNSIBase, 390  
~OsclDNSObserver  
    osclio, 128  
~OsclDestructDealloc  
    OsclDestructDealloc, 384  
~OsclExclusiveArrayPtr  
    OsclExclusiveArrayPtr, 414  
~OsclExclusivePtr  
    OsclExclusivePtr, 417  
~OsclExclusivePtrA  
    OsclExclusivePtrA, 420  
~OsclExecSchedulerCommonBase  
    OsclExecSchedulerCommonBase, 428  
~OsclFileCache  
    OsclFileCache, 435

~OsclGetHostByNameMethod  
     OsclGetHostByNameMethod, 444  
 ~OsclIPSocketI  
     OsclIPSocketI, 450  
 ~OsclJump  
     OsclJump, 452  
 ~OsclListenMethod  
     OsclListenMethod, 453  
 ~OsclLockBase  
     OsclLockBase, 455  
 ~OsclMemPoolFixedChunkAllocator  
     OsclMemPoolFixedChunkAllocator, 471  
 ~OsclMemPoolFixedChunkAllocatorObserver  
     OsclMemPoolFixedChunkAllocatorObserver,  
         474  
 ~OsclMemPoolResizableAllocator  
     OsclMemPoolResizableAllocator, 476  
 ~OsclMemPoolResizableAllocatorMemoryObserver  
     OsclMemPoolResizableAllocatorMemoryOb-  
         server, 484  
 ~OsclMemPoolResizableAllocatorObserver  
     OsclMemPoolResizableAllocatorObserver,  
         485  
 ~OsclMemStatsNode  
     OsclMemStatsNode, 486  
 ~OsclMutex  
     OsclMutex, 488  
 ~OsclNativeFile  
     OsclNativeFile, 493  
 ~OsclNullLock  
     OsclNullLock, 497  
 ~OsclPriorityQueue  
     OsclPriorityQueue, 501  
 ~OsclPriorityQueueBase  
     OsclPriorityQueueBase, 505  
 ~OsclRecvFromMethod  
     OsclRecvFromMethod, 518  
 ~OsclRecvMethod  
     OsclRecvMethod, 522  
 ~OsclRefCounter  
     OsclRefCounter, 525  
 ~OsclRefCounterDA  
     OsclRefCounterDA, 527  
 ~OsclRefCounterMTDA  
     OsclRefCounterMTDA, 531  
 ~OsclRefCounterMTSA  
     OsclRefCounterMTSA, 533  
 ~OsclRefCounterMemFrag  
     OsclRefCounterMemFrag, 529  
 ~OsclRefCounterSA  
     OsclRefCounterSA, 535  
 ~OsclRegistryAccessClient  
     OsclRegistryAccessClient, 537  
 ~OsclRegistryClient  
     OsclRegistryClient, 542  
 ~OsclRegistryServTlsImpl  
     OsclRegistryServTlsImpl, 548  
 ~OsclSchedulerObserver  
     OsclSchedulerObserver, 550  
 ~OsclScopedLock  
     OsclScopedLock, 551  
 ~OsclSemaphore  
     OsclSemaphore, 554  
 ~OsclSendMethod  
     OsclSendMethod, 557  
 ~OsclSendToMethod  
     OsclSendToMethod, 560  
 ~OsclSharedPtr  
     osclbase, 50  
 ~OsclShutdownMethod  
     OsclShutdownMethod, 565  
 ~OsclSingletonEx  
     OsclSingletonEx, 567  
 ~OsclSocketI  
     OsclSocketI, 571  
 ~OsclSocketIBase  
     OsclSocketIBase, 576  
 ~OsclSocketMethod  
     OsclSocketMethod, 581  
 ~OsclSocketObserver  
     OsclSocketObserver, 584  
 ~OsclSocketRequestAO  
     OsclSocketRequestAO, 586  
 ~OsclSocketServ  
     osclio, 129  
 ~OsclSocketServIBase  
     OsclSocketServIBase, 593  
 ~OsclTCPSocket  
     osclio, 129  
 ~OsclTCPSocketI  
     OsclTCPSocketI, 603  
 ~OsclTLS  
     OsclTLS, 624  
 ~OsclTLSEx  
     OsclTLSEx, 626  
 ~OsclThread  
     OsclThread, 606  
 ~OsclThreadLock  
     OsclThreadLock, 610  
 ~OsclTimer  
     OsclTimer, 614  
 ~OsclTimerObject  
     OsclTimerObject, 619  
 ~OsclTimerObserver  
     OsclTimerObserver, 622  
 ~OsclUDPSocket  
     osclio, 129  
 ~OsclUDPSocketI

OsclUDPSocketI, 637  
~Oscl\_Alloc  
    Oscl\_Alloc, 208  
~Oscl\_Dealloc  
    Oscl\_Dealloc, 209  
~Oscl\_File  
    Oscl\_File, 219  
~Oscl\_FileFind  
    Oscl\_FileFind, 227  
~Oscl\_FileServer  
    Oscl\_FileServer, 230  
~Oscl\_Linked\_List  
    Oscl\_Linked\_List, 243  
~Oscl\_Linked\_List\_Base  
    Oscl\_Linked\_List\_Base, 249  
~Oscl\_MTLinked\_List  
    Oscl\_MTLinked\_List, 260  
~Oscl\_Opaque\_Type\_Alloc  
    Oscl\_Opaque\_Type\_Alloc, 264  
~Oscl\_Opaque\_Type\_Alloc\_LL  
    Oscl\_Opaque\_Type\_Alloc\_LL, 266  
~Oscl\_Opaque\_Type\_Compare  
    Oscl\_Opaque\_Type\_Compare, 268  
~Oscl\_Queue  
    Oscl\_Queue, 272  
~Oscl\_Queue\_Base  
    Oscl\_Queue\_Base, 274  
~Oscl\_Rb\_Tree  
    Oscl\_Rb\_Tree, 279  
~Oscl\_TAlloc  
    Oscl\_TAlloc, 312  
~Oscl\_Tag  
    Oscl\_Tag, 302  
~Oscl\_TagTree  
    Oscl\_TagTree, 307  
~Oscl\_Vector  
    Oscl\_Vector, 315  
~Oscl\_Vector\_Base  
    Oscl\_Vector\_Base, 321  
~PVActiveBase  
    PVActiveBase, 642  
~PVLogger  
    PVLogger, 646  
~PVLoggerAppender  
    PVLoggerAppender, 652  
~PVLoggerFilter  
    PVLoggerFilter, 654  
~PVLoggerLayout  
    PVLoggerLayout, 655  
~PVLoggerRegistry  
    PVLoggerRegistry, 657  
~PVSchedulerStopper  
    PVSchedulerStopper, 660  
~PVThreadContext  
    PVThreadContext, 663  
~SendToParam  
    SendToParam, 670  
~\_OsclBasicAllocator  
    \_OsclBasicAllocator, 136  
~\_OsclHeapBase  
    \_OsclHeapBase, 138  
\_OSCL\_Abort  
    osclbase, 38  
\_OSCL\_CLEANUP\_BASE\_CLASS  
    osclmemory, 55  
\_OSCL\_TRAP\_NEW  
    osclmemory, 55  
\_OsclBasicAllocator, 135  
    ~\_OsclBasicAllocator, 136  
    allocate, 136  
    deallocate, 136  
\_OsclHeapBase, 137  
    ~\_OsclHeapBase, 138  
    \_OsclHeapBase, 138  
    \_OsclHeapBase, 138  
    PVCleanupStack, 138  
\_OsclInteger64Transport  
    oscl\_int64\_utils.h, 739  
\_Ownership  
    OSCLMemAutoPtr, 464  
\_PVLOGGER\_LOGBIN  
    pvlogger.h, 881  
\_PVLOGGER\_LOGBIN\_V  
    pvlogger.h, 881  
\_PVLOGGER\_LOGMSG  
    pvlogger.h, 881  
\_PVLOGGER\_LOGMSG\_V  
    pvlogger.h, 881  
\_PV\_TRAP  
    osclerror, 96  
\_PV\_TRAP\_NO\_TLS  
    osclerror, 96, 97  
\_Ptr  
    OsclExclusiveArrayPtr, 415  
    OsclExclusivePtr, 418  
    OsclExclusivePtrA, 421  
    OsclSingletonEx, 568  
    OsclTLS, 625  
    OsclTLSEx, 627  
\_STRLIT  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876  
\_STRLIT\_CHAR  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876  
\_STRLIT\_WCHAR  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876

\_\_TFS\_\_  
     osclconfig.h, 834  
 \_\_Validate\_BasicTimeDateStruct\_\_  
     osclconfig\_time\_check.h, 868  
 \_\_Validate\_BasicTimeStruct\_\_  
     osclconfig\_time\_check.h, 868  
 \_\_int16\_check\_\_  
     osclconfig, 25  
 \_\_int32\_check\_\_  
     osclconfig, 25  
 \_\_int8\_check\_\_  
     osclconfig, 25  
 \_\_uint16\_check\_\_  
     osclconfig, 25  
 \_\_uint32\_check\_\_  
     osclconfig, 25  
 \_\_uint8\_check\_\_  
     osclconfig, 25  
 \_\_verify\_TOscIConditionObject\_defined\_\_  
     osclconfig\_proc\_check.h, 861  
 \_\_verify\_TOscIFileOffset\_defined\_\_  
     osclconfig\_io\_check.h, 852  
 \_\_verify\_TOscIMutexObject\_defined\_\_  
     osclconfig\_proc\_check.h, 861  
 \_\_verify\_TOscISemaphoreObject\_defined\_\_  
     osclconfig\_proc\_check.h, 861  
 \_\_verify\_TOscIThreadFuncArg\_defined\_\_  
     osclconfig\_proc\_check.h, 861  
 \_\_verify\_TOscIThreadFuncRet\_defined\_\_  
     osclconfig\_proc\_check.h, 861  
 \_\_verify\_TOscIThreadId\_defined\_\_  
     osclconfig\_proc\_check.h, 861  
 \_\_verify\_TOscIThreadObject\_defined\_\_  
     osclconfig\_proc\_check.h, 862  
 \_fixedCaches  
     OsclFileCache, 435  
 \_movableCache  
     OsclFileCache, 435  
 \_oscl\_calloc  
     osclmemory, 64  
 \_oscl\_default\_new  
     osclmemory, 64  
 \_oscl\_free  
     osclmemory, 64  
 \_oscl\_malloc  
     osclmemory, 64  
 \_oscl\_realloc  
     osclmemory, 64  
 a  
     internalLeave, 172  
 Abort  
     OsclIDNSMethod, 393  
     OsclIDNSRequestAO, 397  
                 OsclSocketMethod, 581  
                 OsclSocketRequestAO, 586  
 AbortAll  
     OsclIDNSMethod, 393  
     OsclSocketMethod, 581  
 Accept  
     OsclAcceptMethod, 339  
     OsclAcceptRequest, 341  
     osclo, 110  
     OsclSocketI, 571  
     OsclSocketIBase, 576  
     OsclTCPSocketI, 603  
 AcceptParam, 139  
     AcceptParam, 139  
     iBlankSocket, 139  
 AcceptRequest  
     OsclAcceptMethod, 339  
 Activate  
     PVActiveBase, 642  
 Add  
     OsclSocketServRequestList, 595  
     OsclTimerQ, 623  
 add\_element  
     Oscl\_Linked\_List, 244  
     Oscl\_Linked\_List\_Base, 249  
     Oscl\_MTLLinked\_List, 261  
 add\_ref  
     CHHeapRep, 158  
 add\_to\_front  
     Oscl\_Linked\_List, 244  
     Oscl\_Linked\_List\_Base, 249  
     Oscl\_MTLLinked\_List, 261  
 AddAppender  
     PVLogger, 646  
 AddFilter  
     PVLogger, 646  
 AddFixedCache  
     Oscl\_File, 219  
     OsclFileCache, 435  
 AddFragment  
     BuffFragGroup, 149  
 AddLocalFragment  
     MediaData, 179  
 addnewmempoolbuffer  
     OsclMemPoolResizableAllocator, 477  
 addRef  
     Oscl\_DefAllocWithRefCounter, 211  
     OsclMemPoolFixedChunkAllocator, 471  
     OsclMemPoolResizableAllocator, 477  
     OsclRefCounter, 525  
     OsclRefCounterDA, 528  
     OsclRefCounterMTDA, 532  
     OsclRefCounterMTSA, 534  
     OsclRefCounterSA, 536

address  
    Oscl\_TAlloc, 312  
addressListCapacity  
    GetHostNameParam, 168  
AddToExecTimerQ  
    OsclExecSchedulerCommonBase, 428  
AddToScheduler  
    OsclActiveObject, 344  
    OsclTimerObject, 619  
    PVActiveBase, 642  
After  
    OsclTimerObject, 619  
Alloc  
    OsclIPSocketI, 450  
    OsclSocketMethod, 581  
    OsclSocketRequestAO, 586  
ALLOC\_AND\_CONSTRUCT  
    osclbase, 34  
alloc\_and\_construct  
    Oscl\_TAlloc, 312  
alloc\_and\_construct\_fl  
    Oscl\_TAlloc, 312  
ALLOC\_NODE\_FLAG  
    osclmemory, 67  
alloc\_type  
    PVLogger, 646  
    PVLoggerRegistry, 657  
ALLOCATE  
    osclbase, 34  
allocate  
    \_OsclBasicAllocator, 136  
    MemAllocator, 183  
    Oscl\_Alloc, 208  
    Oscl\_DefAlloc, 210  
    Oscl\_Opaque\_Type\_Alloc, 264  
    Oscl\_Opaque\_Type\_Alloc\_LL, 266  
    Oscl\_TAlloc, 312  
    OsclErrorAllocator, 407  
    OsclMemAllocator, 457  
    OsclMemAllocDestructDealloc, 458  
    OSCLMemAutoPtr, 462  
    OsclMemBasicAllocator, 465  
    OsclMemBasicAllocDestructDealloc, 466  
    OsclMemPoolFixedChunkAllocator, 471  
    OsclMemPoolResizableAllocator, 477  
    OsclReadyAlloc, 514  
allocate\_fl  
    Oscl\_Alloc, 208  
    Oscl\_DefAlloc, 210  
    Oscl\_TAlloc, 312  
    OsclReadyAlloc, 514  
allocateblock  
    OsclMemPoolResizableAllocator, 477  
allocator, 140  
allocNum  
    MM\_AllocInfo, 190  
    MM\_AllocQueryInfo, 193  
AllPassFilter, 141  
    ~AllPassFilter, 142  
    AllPassFilter, 142  
    filter\_status\_type, 141  
    FilterOpaqueMessge, 142  
    FilterString, 142  
    log\_level\_type, 141  
    message\_id\_type, 141  
ALREADY\_SUSPENDED\_ERROR  
    OsclProcStatus, 508  
Append  
    OsclPtr, 509  
append  
    CFastRep, 156  
    CHHeapRep, 158  
    CStackRep, 164  
APPEND\_MEDIA\_AT\_END  
    osclutil, 92  
append\_rep  
    CHHeapRep, 158  
    OSCL\_String, 298  
    OSCL\_wString, 336  
AppendBuffers  
    PVLoggerAppender, 652  
AppendNext  
    BufFragGroup, 149  
AppendString  
    PVLoggerAppender, 652  
assign  
    CHHeapRep, 158  
assign\_vector  
    Oscl\_Vector\_Base, 321  
asyncfilereadcancel\_test  
    Oscl\_File, 225  
asyncfilereadwrite\_test  
    Oscl\_File, 225  
Attach  
    OsclBinStream, 369  
audit\_type  
    OsclMemGlobalAuditObject, 468  
available\_localbuf  
    MediaData, 181  
back  
    Oscl\_Queue, 272  
    Oscl\_Vector, 316  
BAD\_THREADID\_ADDR\_ERROR  
    OsclProcStatus, 507  
base\_link\_type  
    Oscl\_Rb\_Tree\_Base, 283  
    Oscl\_Rb\_Tree\_Const\_Iterator, 285

Oscl\_Rb\_Tree\_Iterator, 288  
 Oscl\_Rb\_Tree\_Node\_Base, 291  
**begin**  
 Oscl\_Map, 256  
 Oscl\_Rb\_Tree, 279  
 Oscl\_TagTree, 307  
 Oscl\_Vector, 316  
**BeginScheduling**  
 OsclExecSchedulerCommonBase, 428  
**BFG\_SUCCESS**  
 BufFragStatusClass, 151  
**big\_endian\_to\_host**  
 osclbase, 38  
**Bind**  
 osclbase, 38, 39  
 OsclBindMethod, 354  
 OsclBindRequest, 355  
 osclio, 110  
 OsclIPSocketI, 450  
 OsclSocketI, 571  
 OsclSocketIBase, 576  
**bind**  
 BufferState, 146  
**BindAsync**  
 osclio, 110, 111  
 OsclSocketIBase, 576  
 OsclTCPSocketI, 603  
 OsclUDPSocketI, 637  
**BindParam**, 143  
 BindParam, 143  
 iAddr, 143  
**BindRequest**  
 OsclBindMethod, 354  
**black**  
 Oscl\_Rb\_Tree\_Node\_Base, 291  
**BlockingLoopL**  
 OsclExecSchedulerCommonBase, 428  
**bSetFailure**  
 MM\_AllocInfo, 190  
**Buffer**  
 OsclAsyncFileBuffer, 352  
**buffer**  
 CFastRep, 156  
 CHheapRep, 158  
 CStackRep, 164  
**buffer\_states**  
 BufFragGroup, 150  
**BufferFragment**, 144  
**BufferFreeFuncPtr**  
 osclutil, 74  
**BufferMgr**, 145  
 ~BufferMgr, 145  
 BufferReleased, 145  
**BufferReleased**  
 BufferMgr, 145  
 BufferState, 146  
 bind, 146  
 BufferState, 146  
 decrement\_refcnt, 146  
 get\_buf\_mgr, 146  
 get\_free\_function, 146  
 get\_ptr, 146  
 get\_refcount, 146  
 increment\_refcnt, 146  
 reset, 146  
**BufFragGroup**, 148  
 ~BufFragGroup, 148  
 AddFragment, 149  
 AppendNext, 149  
 buffer\_states, 150  
 BufFragGroup, 148  
 Clear, 149  
 fragments, 150  
 GetLength, 149  
 GetMaxFrags, 149  
 GetNext, 149  
 GetNumFrags, 149  
 length, 150  
 next, 150  
 num\_fragments, 150  
**BufFragStatusClass**, 151  
 BFG\_SUCCESS, 151  
 EMPTY\_FRAGMENT, 151  
 FIXED\_FRAG\_LOC\_FULL, 151  
 INTERNAL\_ERROR, 151  
 INVALID\_ID, 151  
 NOT\_ENOUGH\_SPACE, 151  
 NULL\_INPUT, 151  
 status\_t, 151  
 TOO\_MANY\_FRAGS, 151  
**bufsize**  
 Oscl\_Queue\_Base, 276  
 Oscl\_Vector\_Base, 324  
**BYTES\_IN\_UUID\_ARRAY**  
 oscl\_uuid.h, 830  
**c**  
 OsclPriorityQueue, 503  
**c\_bool**  
 osclbase, 37  
**c\_str**  
 StrPtrLen, 678  
 WStrPtrLen, 692  
**Callback**  
 OsclReadyQ, 516  
**callback\_timer\_type**  
 OsclTimer, 614  
**CallbackTimer**, 152

~CallbackTimer, 152  
CallbackTimer, 152  
Run, 152  
CallbackTimer< Alloc >  
    OsclTimer, 616  
CallbackTimerObserver, 154  
    ~CallbackTimerObserver, 154  
    TimerBaseElapsed, 154  
CallRunExec  
    OsclExecSchedulerCommonBase, 428  
Cancel  
    OsclActiveObject, 344  
    OsclTimer, 614  
    OsclTimerObject, 619  
    PVActiveBase, 642  
CancelAccept  
    osclio, 111  
    OsclSocketIBase, 577  
    OsclTCPSocketI, 603  
CancelBind  
    osclio, 111  
    OsclSocketIBase, 577  
    OsclTCPSocketI, 603  
    OsclUDPSocketI, 637  
CancelConnect  
    osclio, 112  
    OsclSocketIBase, 577  
    OsclTCPSocketI, 603  
CancelFreeChunkAvailableCallback  
    OsclMemPoolFixedChunkAllocator, 471  
    OsclMemPoolResizableAllocator, 477  
CancelFreeMemoryAvailableCallback  
    OsclMemPoolResizableAllocator, 477  
CancelFxns  
    OsclDNSIBase, 390  
    OsclSocketIBase, 577  
CancelGetHostByName  
    OsclDNSIBase, 390  
    osclio, 112  
Cancelled  
    OsclIDNSRequestAO, 397  
CancelListen  
    osclio, 112  
    OsclSocketIBase, 577  
    OsclTCPSocketI, 603  
CancelMethod  
    OsclIDNSMethod, 393  
    OsclSocketMethod, 581  
CancelRecv  
    osclio, 112  
    OsclSocketIBase, 577  
    OsclTCPSocketI, 603  
CancelRecvFrom  
    osclio, 112  
OsclSocketIBase, 577  
OsclUDPSocketI, 637  
CancelSend  
    osclio, 112  
    OsclSocketIBase, 577  
    OsclTCPSocketI, 603  
CancelSendTo  
    osclio, 112  
    OsclSocketIBase, 577  
    OsclUDPSocketI, 637  
CancelShutdown  
    osclio, 113  
    OsclSocketIBase, 577  
    OsclTCPSocketI, 603  
canPersistMoreHostAddresses  
    GetHostByNameParam, 169  
CanTerminate  
    OsclThread, 606  
capacity  
    Oscl\_Queue\_Base, 275  
    Oscl\_Vector\_Base, 321  
    OsclFileCacheBuffer, 437  
CFastRep, 155  
    append, 156  
    buffer, 156  
    CFastRep, 156  
    maxsize, 156  
    overwrite, 156  
    set\_r, 156  
    set\_w, 156  
    size, 156  
    writable, 156  
chartype  
    OSCL\_FastString, 214  
    OSCL\_HeapString, 234  
    OSCL\_HeapStringA, 236  
    OSCL\_StackString, 295  
    OSCL\_String, 298  
    OSCL\_wFastString, 325  
    OSCL\_wHeapString, 329  
    OSCL\_wHeapStringA, 331  
    OSCL\_wStackString, 334  
    OSCL\_wString, 336  
CHheapRep, 157  
    add\_ref, 158  
    append, 158  
    append\_rep, 158  
    assign, 158  
    buffer, 158  
    CHheapRep, 158  
    maxsize, 158  
    refcount, 159  
    remove\_ref, 158  
    set, 158

set\_rep, 158  
 size, 159  
 check\_fence  
     MM\_AllocBlockFence, 186  
 check\_list  
     Oscl\_Linked\_List, 244  
     Oscl\_Linked\_List\_Base, 249  
 checkSum  
     StrCSumPtrLen, 676  
 CheckSumType  
     StrCSumPtrLen, 675  
 children  
     Oscl\_TagTree::Node, 203  
 children\_type  
     Oscl\_TagTree, 307  
     Oscl\_TagTree::Node, 202  
 ChooseCurCache  
     Oscl\_File::OsclCacheObserver, 374  
 CleanInUse  
     OsclAsyncFileBuffer, 352  
 Cleanup  
     OsclErrorTrap, 409  
     OsclInit, 446  
     OsclMem, 456  
     OsclScheduler, 549  
     PVLLogger, 647  
 CleanupExecQ  
     OsclExecSchedulerCommonBase, 428  
 CleanupParam  
     OsclSocketRequestAO, 586  
 Clear  
     BufFragGroup, 149  
     MediaData, 179  
     OsclTimer, 614  
 clear  
     Oscl\_Linked\_List, 244  
     Oscl\_Map, 256  
     Oscl\_Queue, 273  
     Oscl\_Queue\_Base, 275  
     Oscl\_Rb\_Tree, 280  
     Oscl\_TagTree, 308  
     Oscl\_Vector, 316  
 ClearTOS  
     OsclSocketTOS, 599  
 Close  
     Oscl\_File, 219  
     Oscl\_FileFind, 227  
     Oscl\_FileServer, 230  
     OsclAsyncFile, 349  
     OsclDNSI, 387  
     OsclDNSIBase, 390  
     OsclFileCache, 435  
     osclo, 113  
     OsclIPSocketI, 450  
     OsclMutex, 488  
     OsclNativeFile, 493  
     OsclRegistryAccessClient, 537  
     OsclRegistryClient, 542  
     OsclRegistryClientImpl, 544  
     OsclRegistryServTlsImpl, 548  
     OsclSemaphore, 554  
     OsclSocketI, 571  
     OsclSocketIBase, 577  
     OsclSocketServI, 590  
     OsclSocketServIBase, 593  
     OsclSocketServRequestList, 595  
     OsclTCPSocketI, 603  
     OsclUDPSocketI, 637  
 CloseSession  
     OsclComponentRegistry, 377  
 color  
     Oscl\_Rb\_Tree\_Node\_Base, 292  
 color\_type  
     Oscl\_Rb\_Tree\_Node\_Base, 291  
 comp  
     Oscl\_Map::value\_compare, 691  
     OsclPriorityQueue, 503  
 compare  
     OsclCompareLess, 375  
     OsclReadyCompare, 515  
     OsclTimerCompare, 617  
 compare\_data  
     Oscl\_Opaque\_Type\_Alloc\_LL, 266  
 compare\_EQ  
     Oscl\_Opaque\_Type\_Compare, 268  
     OsclPriorityQueue, 501  
 compare\_LT  
     Oscl\_Opaque\_Type\_Compare, 268  
     OsclPriorityQueue, 501  
 CompareId  
     OsclThread, 607  
 COMPUTE\_MEM\_ALIGN\_SIZE  
     osclmemory, 55  
 Connect  
     Oscl\_FileServer, 230  
     OsclConnectMethod, 381  
     OsclConnectRequest, 383  
     osclo, 113, 114  
     OsclRegistryAccessClient, 537  
     OsclRegistryClient, 542  
     OsclRegistryClientImpl, 544  
     OsclRegistryServTlsImpl, 548  
     OsclSocketI, 571  
     OsclSocketIBase, 577  
     OsclSocketServI, 590  
     OsclSocketServIBase, 593  
     OsclTCPSocketI, 604  
 ConnectParam, 160

ConnectParam, 160  
iAddr, 160  
ConnectRequest  
    OsclConnectMethod, 381  
const\_iterator  
    Oscl\_Map, 255  
    Oscl\_Rb\_Tree, 279  
    Oscl\_Rb\_Tree\_Const\_Iterator, 285  
    Oscl\_TagTree::const\_iterator, 162  
    Oscl\_Vector, 315  
const\_pointer  
    Oscl\_Rb\_Tree, 279  
    Oscl\_TAlloc, 312  
const\_reference  
    Oscl\_Map, 255  
    Oscl\_Queue, 272  
    Oscl\_Rb\_Tree, 279  
    Oscl\_TAlloc, 312  
    Oscl\_Vector, 315  
    OsclPriorityQueue, 501  
Construct  
    OsclReadyQ, 516  
    OsclTimerQ, 623  
construct  
    Oscl\_Linked\_List\_Base, 249  
    Oscl\_Opaque\_Type\_Alloc, 264  
    Oscl\_Opaque\_Type\_Alloc\_LL, 266  
    Oscl\_Queue\_Base, 275  
    Oscl\_TAlloc, 312  
    Oscl\_Vector\_Base, 321  
    OsclPriorityQueueBase, 505  
ConstructL  
    OsclIDNSMethod, 393  
    OsclIDNSRequestAO, 397  
    OsclExecSchedulerCommonBase, 428  
    OsclIPSocketI, 450  
    OsclSocketMethod, 581  
    OsclSocketRequestAO, 586  
container\_type  
    OsclPriorityQueue, 501  
Contains  
    Oscl\_File::OsclFixedCacheParam, 443  
    OsclFileCacheBuffer, 436  
count  
    Oscl\_Map, 256  
    Oscl\_Rb\_Tree, 280  
    Oscl\_TagTree, 308  
CPVInterfaceProxy  
    OsclErrorTrapImp, 411  
Create  
    GetHostByNameParam, 169  
    OsclMutex, 488  
    OsclSemaphore, 554  
    OsclThread, 607  
createmempool  
    OsclMemPoolFixedChunkAllocator, 472  
CreatePVLogger  
    PVLoggerRegistry, 658  
CStackRep, 164  
    append, 164  
    buffer, 164  
    CStackRep, 164  
    maxsize, 164  
    set, 164  
    size, 165  
CTIME\_BUFFER\_SIZE  
    osclbase, 51  
CtimeStrBuf  
    osclbase, 37  
Current  
    OsclExecScheduler, 423  
currentPos  
    OsclFileCacheBuffer, 437  
data  
    LinkedListElement, 176  
data1  
    OsclUuid, 640  
data2  
    OsclUuid, 640  
data3  
    OsclUuid, 640  
data4  
    OsclUuid, 640  
deallocate  
    OsclBasicAllocator, 136  
    MemAllocator, 183  
    Oscl\_Dealloc, 209  
    Oscl\_DefAlloc, 210  
    Oscl\_Opaque\_Type\_Alloc, 264  
    Oscl\_Opaque\_Type\_Alloc\_LL, 266  
    Oscl\_TAlloc, 312, 313  
    OsclErrorAllocator, 408  
    OsclMemAllocator, 457  
    OsclMemAllocDestructDealloc, 458  
    OSCLMemAutoPtr, 462  
    OsclMemBasicAllocator, 465  
    OsclMemBasicAllocDestructDealloc, 466  
    OsclMemPoolFixedChunkAllocator, 472  
    OsclMemPoolResizableAllocator, 477  
    OsclReadyAlloc, 514  
deallocateblock  
    OsclMemPoolResizableAllocator, 478  
decrement\_refcnt  
    BufferState, 146  
DEFAULT\_MM\_AUDIT\_MODE  
    osclmemory, 56  
DEFAULT\_POSTFILL\_PATTERN

osclmemory, 56  
 DEFAULT\_PREFILL\_PATTERN  
     osclmemory, 56  
 Delete  
     Oscl\_DefAllocWithRefCounter, 211  
     OsclAsyncFile, 349  
     OsclBuf, 372  
 Depth  
     OsclReadyQ, 516  
 depth  
     Oscl\_TagTree::Node, 202  
 dequeue\_element  
     Oscl\_Linked\_List, 244  
     Oscl\_MTLinkedList, 261  
 Des  
     OsclBuf, 372  
 DesC  
     OsclBuf, 372  
 Destroy  
     DNSRequestParam, 166  
     GetHostNameParam, 169  
     PVActiveBase, 642  
 destroy  
     Oscl\_Linked\_List\_Base, 249  
     Oscl\_Opaque\_Type\_Alloc, 264  
     Oscl\_Opaque\_Type\_Alloc\_LL, 267  
     Oscl\_Queue\_Base, 275  
     Oscl\_TAlloc, 313  
     Oscl\_Vector, 316  
     Oscl\_Vector\_Base, 321  
 destroyallmempoolbuffers  
     OsclMemPoolResizableAllocator, 478  
 destroymempool  
     OsclMemPoolFixedChunkAllocator, 472  
 destruct\_and\_dealloc  
     Oscl\_TAlloc, 313  
     OsclDestructDealloc, 384  
     OsclMemAllocDestructDealloc, 458  
     OsclMemBasicAllocDestructDealloc, 466  
 difference\_type  
     Oscl\_Rb\_Tree, 279  
 DIR\_TYPE  
     Oscl\_FileFind, 226  
 DisableAppenderInheritance  
     PVLogger, 647  
 DiscardAcceptedSocket  
     OsclAcceptMethod, 339  
 DNSRequestParam, 166  
     ~DNSRequestParam, 166  
     Destroy, 166  
     DNSRequestParam, 166  
     iDNSRequest, 167  
     iFxn, 167  
     iRefCount, 167  
 DoCancel  
     OsclActiveObject, 344  
     OsclDNSRequestAO, 397  
     OsclSocketRequestAO, 587  
     OsclTimerObject, 619  
     PVActiveBase, 642  
 E\_BUFFER\_TOO\_SMALL  
     Oscl\_FileFind, 227  
 E\_INVALID\_ARG  
     Oscl\_FileFind, 227  
 E\_INVALID\_STATE  
     Oscl\_FileFind, 226  
 E\_MEMORY\_ERROR  
     Oscl\_FileFind, 227  
 E\_NO\_MATCH  
     Oscl\_FileFind, 227  
 E\_NOT\_IMPLEMENTED  
     Oscl\_FileFind, 227  
 E\_OK  
     Oscl\_FileFind, 226  
 E\_OTHER  
     Oscl\_FileFind, 227  
 E\_PATH\_NOT\_FOUND  
     Oscl\_FileFind, 227  
 E\_PATH\_TOO\_LONG  
     Oscl\_FileFind, 227  
 element\_type  
     Oscl\_FileFind, 226  
 elems  
     Oscl\_Queue\_Base, 276  
     Oscl\_Vector\_Base, 324  
 empty  
     Oscl\_Map, 256  
     Oscl\_Queue\_Base, 275  
     Oscl\_Rb\_Tree, 280  
     Oscl\_TagTree, 308  
     Oscl\_Vector\_Base, 322  
     OsclPriorityQueue, 502  
 EMPTY\_FRAGMENT  
     BufFragStatusClass, 151  
 EMPTY\_UUID  
     oscl\_uuid.h, 830  
 enablenullpointerreturn  
     OsclMemPoolFixedChunkAllocator, 472  
     OsclMemPoolResizableAllocator, 478  
 End  
     OsclFileStats, 441  
 end  
     Oscl\_Map, 256  
     Oscl\_Rb\_Tree, 280  
     Oscl\_TagTree, 308

Oscl\_Vector, 316  
EndOfFile  
    Oscl\_File, 220  
    OsclAsyncFile, 349  
    OsclFileCache, 435  
    OsclNativeFile, 493  
endPos  
    OsclFileCacheBuffer, 437  
EndScheduling  
    OsclExecSchedulerCommonBase, 428  
EnterThreadContext  
    PVThreadContext, 663  
eof  
    OsclBinStream, 370  
EOF\_STATE  
    OsclBinStream, 369  
EOSCL\_StringOp\_CompressASCII  
    osclutil, 75  
EOSCL\_StringOp\_UTF16ToUTF8  
    osclutil, 75  
EOSCL\_wStringOp\_ExpandASCII  
    osclutil, 75  
EOSCL\_wStringOp\_UTF8ToUTF16  
    osclutil, 75  
EOsclFileOp\_Close  
    oscilio, 109  
EOsclFileOp\_EndOfFile  
    oscilio, 109  
EOsclFileOp\_Flush  
    oscilio, 109  
EOsclFileOp\_Last  
    oscilio, 109  
EOsclFileOp\_NativeClose  
    oscilio, 109  
EOsclFileOp\_NativeEndOfFile  
    oscilio, 109  
EOsclFileOp\_NativeFlush  
    oscilio, 109  
EOsclFileOp\_NativeOpen  
    oscilio, 109  
EOsclFileOp\_NativeRead  
    oscilio, 109  
EOsclFileOp\_NativeSeek  
    oscilio, 109  
EOsclFileOp\_NativeSetSize  
    oscilio, 109  
EOsclFileOp\_NativeSize  
    oscilio, 109  
EOsclFileOp\_NativeTell  
    oscilio, 109  
EOsclFileOp\_NativeWrite  
    oscilio, 109  
EOsclFileOp\_Open  
    oscilio, 109  
EOsclFileOp\_Read  
    oscilio, 109  
EOsclFileOp\_Seek  
    oscilio, 109  
EOsclFileOp\_SetSize  
    oscilio, 109  
EOsclFileOp\_Size  
    oscilio, 109  
EOsclFileOp\_Tell  
    oscilio, 109  
EOsclFileOp\_Write  
    oscilio, 109  
eOsclProcError  
    OsclProcStatus, 507  
EOsclSocket\_DataRecv  
    oscl\_socket\_stats.h, 803  
EOsclSocket\_DataSent  
    oscl\_socket\_stats.h, 803  
EOsclSocket Except  
    oscl\_socket\_stats.h, 803  
EOsclSocket\_OS  
    oscl\_socket\_stats.h, 803  
EOsclSocket\_Readable  
    oscl\_socket\_stats.h, 803  
EOsclSocket\_RequestAO\_Canceled  
    oscl\_socket\_stats.h, 802  
EOsclSocket\_RequestAO\_Error  
    oscl\_socket\_stats.h, 802  
EOsclSocket\_RequestAO\_Success  
    oscl\_socket\_stats.h, 802  
EOsclSocket\_RequestAO\_Timeout  
    oscl\_socket\_stats.h, 802  
EOsclSocket\_ServPoll  
    oscl\_socket\_stats.h, 803  
EOsclSocket\_ServRequestCancelIssued  
    oscl\_socket\_stats.h, 803  
EOsclSocket\_ServRequestComplete  
    oscl\_socket\_stats.h, 803  
EOsclSocket\_ServRequestIssued  
    oscl\_socket\_stats.h, 802  
EOsclSocket\_Writable  
    oscl\_socket\_stats.h, 803  
EOsclSocketServ\_LastEvent  
    oscl\_socket\_stats.h, 802  
EOsclSocketServ\_LoopsockError  
    oscl\_socket\_stats.h, 803  
EOsclSocketServ\_LoopsockOk  
    oscl\_socket\_stats.h, 803  
EOsclSocketServ\_SelectActivity  
    oscl\_socket\_stats.h, 802  
EOsclSocketServ\_SelectNoActivity  
    oscl\_socket\_stats.h, 802  
EOsclSocketServ\_SelectRescheduleAsap  
    oscl\_socket\_stats.h, 802

EOsclSocketServ\_SelectReschedulePoll  
     oscl\_socket\_stats.h, 802

EOsclThreadTerminate\_Join  
     oscl\_thread.h, 820

EOsclThreadTerminate\_Kill  
     oscl\_thread.h, 820

EOsclThreadTerminate\_NOP  
     oscl\_thread.h, 820

EPriorityHigh  
     OsclActiveObject, 343

EPriorityHighest  
     OsclActiveObject, 343

EPriorityIdle  
     OsclActiveObject, 343

EPriorityLow  
     OsclActiveObject, 343

EPriorityNominal  
     OsclActiveObject, 343

EPV\_ARM\_GNUC  
     osclbase, 34

EPV\_ARM\_MSEVC  
     osclbase, 34

EPV\_ARM\_RVCT  
     osclbase, 34

EPVCritic\_Ecp  
     OsclSocketTOS, 598

EPVDNSCancel  
     osclio, 109

EPVDNSFailure  
     osclio, 109

EPVDNSGetHostByName  
     osclio, 110

EPVDNSPending  
     osclio, 109

EPVDNSSuccess  
     osclio, 109

EPVDNSTimeout  
     osclio, 109

EPVFlash  
     OsclSocketTOS, 598

EPVHiRel  
     OsclSocketTOS, 598

EPVHiThrpt  
     OsclSocketTOS, 598

EPVImmediate  
     OsclSocketTOS, 598

EPVInetControl  
     OsclSocketTOS, 598

EPVIPAddMembership  
     oscl\_socket\_types.h, 806

EPVIPMulticastTTL  
     oscl\_socket\_types.h, 806

EPVIPPtoIP  
     oscl\_socket\_types.h, 806

EPVIPPProtoTCP  
     oscl\_socket\_types.h, 806

EPVIPTOS  
     oscl\_socket\_types.h, 806

EPVLDelay  
     OsclSocketTOS, 598

EPVNetControl  
     OsclSocketTOS, 598

EPVNoTOS  
     OsclSocketTOS, 598

EPVOverrideFlash  
     OsclSocketTOS, 598

EPVPriority  
     OsclSocketTOS, 598

EPVRoutine  
     OsclSocketTOS, 598

EPVSocket  
     oscl\_socket\_types.h, 806

EPVSocket\_Last  
     oscl\_socket\_types.h, 806

EPVSocketAccept  
     oscl\_socket\_types.h, 806

EPVSocketBind  
     oscl\_socket\_types.h, 806

EPVSocketBothShutdown  
     oscl\_socket\_types.h, 806

EPVSocketCancel  
     oscl\_socket\_types.h, 806

EPVSocketConnect  
     oscl\_socket\_types.h, 806

EPVSocketFailure  
     oscl\_socket\_types.h, 806

EPVSocketListen  
     oscl\_socket\_types.h, 806

EPVSocketNotImplemented  
     oscl\_socket\_types.h, 806

EPVSocketPending  
     oscl\_socket\_types.h, 805

EPVSocketRecv  
     oscl\_socket\_types.h, 806

EPVSocketRecvFrom  
     oscl\_socket\_types.h, 806

EPVSocketRecvShutdown  
     oscl\_socket\_types.h, 806

EPVSocketSend  
     oscl\_socket\_types.h, 806

EPVSocketSendShutdown  
     oscl\_socket\_types.h, 806

EPVSocketSendTo  
     oscl\_socket\_types.h, 806

EPVSocketShutdown  
     oscl\_socket\_types.h, 806

EPVSocketSuccess  
     oscl\_socket\_types.h, 805

EPVSocketTimeout  
    oscl\_socket\_types.h, 806  
EPVSockReuseAddr  
    oscl\_socket\_types.h, 806  
EPVThreadContext\_InThread  
    osclproc, 133  
EPVThreadContext\_NonOsclThread  
    osclproc, 133  
EPVThreadContext\_OsclThread  
    osclproc, 133  
EPVThreadContext\_Undetermined  
    osclproc, 133  
equal\_range  
    Oscl\_Map, 257  
    Oscl\_Rb\_Tree, 280  
erase  
    Oscl\_Map, 257  
    Oscl\_Rb\_Tree, 280, 281  
    Oscl\_TagTree, 308  
    Oscl\_Vector, 317  
    Oscl\_Vector\_Base, 322  
Error  
    OsclExecSchedulerCommonBase, 428  
error\_type  
    Oscl\_FileFind, 226  
ESocketServ\_Connected  
    OsclSocketServIBase, 592  
ESocketServ\_Error  
    OsclSocketServIBase, 593  
ESocketServ\_Idle  
    OsclSocketServIBase, 592  
ESymbianAccessMode\_Rfile  
    Oscl\_File, 219  
ESymbianAccessMode\_RfileBuf  
    Oscl\_File, 219  
EXCEED\_MAX\_COUNT\_VARIABLE\_ERROR  
    OsclProcStatus, 508  
EXCEED\_MAX\_SEM\_COUNT\_ERROR  
    OsclProcStatus, 508  
Exit  
    OsclThread, 607  
ExitThreadContext  
    PVThreadContext, 663  
extract\_string  
    osclutil, 75  
fail  
    OsclBinStream, 370  
FAIL\_STATE  
    OsclBinStream, 369  
Failure  
    OsclDNSRequestAO, 397  
FENCE\_PATTERN  
    osclmemory, 56  
FILE\_TYPE  
    Oscl\_FileFind, 226  
fileName  
    MM\_AllocQueryInfo, 193  
filePosition  
    OsclFileCacheBuffer, 437  
FileSize  
    OsclFileCache, 435  
fill\_fence  
    MM\_AllocBlockFence, 186  
FillFromFile  
    OsclFileCacheBuffer, 436  
filter\_status\_type  
    AllPassFilter, 141  
    PVLogger, 646  
    PVLoggerFilter, 653  
FilterOpaqueMessge  
    AllPassFilter, 142  
    PVLoggerFilter, 654  
FilterString  
    AllPassFilter, 142  
    PVLoggerFilter, 654  
Find  
    OsclComponentRegistryData, 378  
find  
    Oscl\_Map, 257  
    Oscl\_Rb\_Tree, 281  
    Oscl\_TagTree, 309  
find\_heap  
    OsclPriorityQueue, 502  
    OsclPriorityQueueBase, 505  
FindExact  
    OsclComponentRegistry, 377  
FindFirst  
    Oscl\_FileFind, 227, 228  
findfreeblock  
    OsclMemPoolResizableAllocator, 478  
FindHierarchical  
    OsclComponentRegistry, 377  
FindNext  
    Oscl\_FileFind, 228  
FindPVB  
    OsclExecSchedulerCommonBase, 428  
first  
    Oscl\_Pair, 270  
firstFragPtr  
    OsclBinStream, 371  
FIXED\_FRAG\_LOC\_FULL  
    BufFragStatusClass, 151  
Flush  
    Oscl\_File, 220  
    OsclAsyncFile, 349  
    OsclFileCache, 435  
    OsclNativeFile, 493

FormatOpaqueMessage  
    PVLoggerLayout, 655

FormatString  
    PVLoggerLayout, 655

fragments  
    BufFragGroup, 150

fragsLeft  
    OsclBinStream, 371

freeblockavailable  
    OsclMemPoolResizableAllocatorObserver,  
        485

freebytes  
    oscl\_fsstat, 232

freechunkavailable  
    OsclMemPoolFixedChunkAllocatorObserver,  
        474

freememoryavailable  
    OsclMemPoolResizableAllocatorMemoryOb-  
        server, 484

front  
    Oscl\_Queue, 273  
    Oscl\_Vector, 317

get  
    OsclBinIStream, 356  
    OsclExclusiveArrayPtr, 414  
    OsclExclusivePtr, 417  
    OsclExclusivePtrA, 420  
    OSCLMemAutoPtr, 463

get\_buf\_mgr  
    BufferState, 146

get\_count  
    osclbase, 39

get\_cstr  
    OSCL\_FastString, 215  
    OSCL\_HeapStringA, 237  
    OSCL\_String, 298  
    OSCL\_wFastString, 326  
    OSCL\_wHeapStringA, 331  
    OSCL\_wString, 336  
    osclutil, 75, 76

get\_data  
    Oscl\_Opaque\_Type\_Alloc\_LL, 267

get\_element  
    Oscl\_Linked\_List, 245  
    Oscl\_Linked\_List\_Base, 250  
    Oscl\_MTLinked\_List, 261

get\_first  
    Oscl\_Linked\_List, 245  
    Oscl\_Linked\_List\_Base, 250

get\_free\_function  
    BufferState, 146

get\_index  
    Oscl\_Linked\_List, 245

Oscl\_Linked\_List\_Base, 250

Oscl\_MTLinked\_List, 261

get\_int64\_lower32  
    Oscl\_Int64\_Utils, 241

get\_int64\_middle32  
    Oscl\_Int64\_Utils, 241

get\_int64\_upper32  
    Oscl\_Int64\_Utils, 241

get\_ISO8601\_str\_time  
    TimeValue, 683

get\_local\_time  
    TimeValue, 683

get\_lower32  
    NTPTime, 206

get\_maxsize  
    OSCL\_FastString, 215  
    OSCL\_HeapStringA, 237  
    OSCL\_String, 298  
    OSCL\_wFastString, 326  
    OSCL\_wHeapStringA, 331  
    OSCL\_wString, 336  
    osclutil, 76

get\_middle32  
    NTPTime, 206

get\_next  
    Oscl\_Linked\_List, 245  
    Oscl\_Linked\_List\_Base, 250  
    Oscl\_Opaque\_Type\_Alloc\_LL, 267

get\_num\_elements  
    Oscl\_Linked\_List, 246

get\_ptr  
    BufferState, 146

get\_pv8601\_str\_time  
    TimeValue, 683

get\_refcount  
    BufferState, 146

get\_registry  
    TLSStorageOps, 687

get\_rfc822\_gmtime\_str  
    TimeValue, 684

get\_sec  
    TimeValue, 684

get\_size  
    OSCL\_FastString, 215  
    OSCL\_HeapStringA, 237  
    OSCL\_String, 299  
    OSCL\_wFastString, 326  
    OSCL\_wHeapStringA, 331  
    OSCL\_wString, 336  
    osclutil, 77

get\_str  
    OSCL\_FastString, 215  
    OSCL\_HeapStringA, 238  
    OSCL\_String, 299

OSCL\_wFastString, 326  
OSCL\_wHeapStringA, 332  
OSCL\_wString, 336  
osclutil, 77, 78  
get\_str\_ctime  
    TimeValue, 684  
get\_timeval\_ptr  
    TimeValue, 684  
get\_timevalue\_in\_usec  
    TimeValue, 685  
get\_uint64\_lower32  
    Oscl\_Int64\_Utils, 241  
get\_uint64\_middle32  
    Oscl\_Int64\_Utils, 241  
get\_uint64\_upper32  
    Oscl\_Int64\_Utils, 241  
get\_upper32  
    NTPTime, 206  
get\_usec  
    TimeValue, 685  
get\_value  
    NTPTime, 206  
GetAcceptedSocket  
    OsclAcceptMethod, 340  
GetAcceptedSocketL  
    osclio, 114  
    OsclTCPSocketL, 604  
getAllocatedSize  
    OsclMemPoolResizableAllocator, 478  
GetAvailableBufferSize  
    MediaData, 179  
getAvailableSize  
    OsclMemPoolResizableAllocator, 478  
getBufferSize  
    OsclMemPoolResizableAllocator, 478  
GetBufferState  
    osclutil, 78  
getCapacity  
    OsclRefCounterMemFrag, 530  
getCheckSum  
    StrCSumPtrLen, 675  
getCount  
    Oscl\_DefAllocWithRefCounter, 211  
    OsclRefCounter, 525  
    OsclRefCounterDA, 528  
    OsclRefCounterMemFrag, 530  
    OsclRefCounterMTDA, 532  
    OsclRefCounterMTSA, 534  
    OsclRefCounterSA, 536  
GetElementType  
    Oscl\_FileFind, 229  
GetError  
    Oscl\_File, 220  
    OsclNativeFile, 493  
GetErrorTrapImp  
    OsclErrorTrap, 409  
GetFactories  
    OsclRegistryAccessClient, 537  
    OsclRegistryClientImpl, 544  
    OsclRegistryServTlsImpl, 548  
GetFactory  
    OsclRegistryAccessClient, 537  
    OsclRegistryClientImpl, 544  
    OsclRegistryServTlsImpl, 548  
GetFragment  
    osclutil, 78  
getGlobalMemAuditObject  
    OsclMemGlobalAuditObject, 468  
getHead  
    OsclDoubleListBase, 401  
GetHostByName  
    OsclDNSI, 387  
    OsclDNSIBase, 390  
    OsclGetHostByNameMethod, 444  
    osclio, 114  
GetHostByNameParam, 168  
    ~GetHostByNameParam, 169  
    addressListCapacity, 168  
    canPersistMoreHostAddresses, 169  
    Create, 169  
    Destroy, 169  
    iAddr, 169  
    iAddressList, 169  
    iName, 169  
    OsclDNSRequestAO, 398  
    PersistHostAddress, 169  
GetHostByNameResponseContainsAliasInfo  
    OsclDNSI, 388  
    OsclDNSIBase, 390  
GetHostByNameSuccess  
    OsclDNSI, 388  
    OsclDNSIBase, 390  
GetId  
    OsclExecSchedulerCommonBase, 428  
    OsclThread, 607  
getInstance  
    OsclSingletonRegistryEx, 569  
    OsclTLSRegistry, 628  
    OsclTLSRegistryEx, 629  
getLargestContiguousFreeBlockSize  
    OsclMemPoolResizableAllocator, 478  
GetLastError  
    Oscl\_FileFind, 229  
getLeaveCode  
    OsclException, 412  
GetLength  
    BufFragGroup, 149  
GetLocalBufsize

MediaData, 179  
 GetLocalFragment  
     MediaData, 179  
 GetLoggerObject  
     PVLogger, 647  
 GetLogLevel  
     PVLogger, 647  
 GetMaxFrags  
     BufFragGroup, 149  
 GetMediaFragment  
     MediaData, 180  
 GetMediaSize  
     MediaData, 180  
 getMemFrag  
     OsclRefCounterMemFrag, 530  
 getMemFragPtr  
     OsclRefCounterMemFrag, 530  
 getMemFragSize  
     OsclRefCounterMemFrag, 530  
 getMemPoolBufferAllocatedSize  
     OsclMemPoolResizableAllocator, 478  
 getMemPoolBufferSize  
     OsclMemPoolResizableAllocator, 479  
 GetName  
     OsclExecSchedulerCommonBase, 428  
 GetNext  
     BufFragGroup, 149  
 GetNextHost  
     OsclIDNSI, 388  
     OsclDNSIBase, 390  
 GetNextHostSuccess  
     OsclIDNSI, 388  
     OsclDNSIBase, 390  
 GetNumAppenders  
     PVLogger, 648  
 GetNumFrags  
     BufFragGroup, 149  
 GetNumMediaFrags  
     MediaData, 180  
 getOffset  
     OsclDoubleListBase, 401  
 GetParent  
     PVLogger, 648  
 GetPeerName  
     osclio, 115  
     OsclIPSocketI, 450  
     OsclSocketI, 571  
 GetPriority  
     OsclThread, 608  
 GetPVLoggerObject  
     PVLoggerRegistry, 658  
 GetPVLoggerRegistry  
     PVLoggerRegistry, 658  
 GetReadAsyncNumElements  
     OsclNativeFile, 493  
 GetRecvData  
     osclio, 115  
     OsclIPSocketI, 450  
     OsclRecvFromMethod, 518  
     OsclRecvFromRequest, 520  
     OsclRecvMethod, 522  
     OsclRecvRequest, 524  
     OsclTCPSocketI, 604  
     OsclUDPSocketI, 637  
 GetRefCounter  
     osclbase, 39  
 getRefCounter  
     OsclRefCounterMemFrag, 530  
 GetRep  
     osclbase, 39  
 GetScheduler  
     OsclExecSchedulerCommonBase, 428  
 GetSendData  
     osclio, 116  
     OsclIPSocketI, 450  
     OsclSendMethod, 557  
     OsclSendRequest, 559  
     OsclSendToMethod, 560  
     OsclSendToRequest, 562  
     OsclTCPSocketI, 604  
     OsclUDPSocketI, 637  
 GetShutdown  
     OsclSocketIBase, 577  
 GetSocketError  
     OsclDNSRequestAO, 397  
     OsclSocketRequestAO, 587  
 GetTimestamp  
     MediaData, 180  
 GetTOS  
     OsclSocketTOS, 599  
 good  
     OsclBinStream, 370  
 GOOD\_STATE  
     OsclBinStream, 369  
 Handle  
     Oscl\_File, 220  
     OsclFileHandle, 438  
 HandleDNSEvent  
     OsclDNSObserver, 395  
 HandleSocketEvent  
     OsclSocketObserver, 584  
 HasAsyncBind  
     OsclSocketIBase, 577  
 HasAsyncListen  
     OsclSocketIBase, 577  
 HasAsyncRead  
     OsclNativeFile, 493

hash  
     OSCL\_String, 299  
     OSCL\_wString, 336

HasThisOffset  
     OsclAsyncFileBuffer, 352

HaveRoomInCurrentBlock  
     OsclBinStream, 370

Head  
     OsclDoubleList, 400  
     OsclPriorityList, 499

head  
     Oscl\_Linked\_List\_Base, 252

HeapBase, 170  
     ~HeapBase, 171  
     HeapBase, 171  
     operator delete, 171  
     operator new, 171

host\_to\_big\_endian  
     osclbase, 39

host\_to\_little\_endian  
     osclbase, 39

iAddedNum  
     PVActiveBase, 644

iAddr  
     BindParam, 143  
     ConnectParam, 160  
     GetHostByNameParam, 169  
     RecvFromParam, 666  
     SendToParam, 670

iAddress  
     OsclIPSocketI, 451

iAddressList  
     GetHostByNameParam, 169

iAlloc  
     OsclDNSIBase, 391  
     OsclDNSMethod, 394  
     OsclExecSchedulerCommonBase, 432  
     OsclIPSocketI, 451  
     OsclSocketIBase, 579  
     OsclSocketServIBase, 593

iAllocatedSz  
     OsclMemPoolResizableAllocator::MemPoolBufferInfo, 185

iAOPriority  
     TReadyQueLink, 688

iAsyncReadBufferSize  
     OsclNativeFileParams, 495

iBlankSocket  
     AcceptParam, 139

iBlockBuffer  
     OsclMemPoolResizableAllocator::MemPoolBlockInfo, 184

iBlockInfoAlignedSize  
     OsclMemPoolResizableAllocator, 482

iBlockingMode  
     OsclExecSchedulerCommonBase, 432

iBlockPostFence  
     OsclMemPoolResizableAllocator::MemPoolBlockInfo, 184

iBlockPreFence  
     OsclMemPoolResizableAllocator::MemPoolBlockInfo, 184

iBlockSize  
     OsclMemPoolResizableAllocator::MemPoolBlockInfo, 184

iBuffer  
     OsclBuf, 373

iBufferInfoAlignedSize  
     OsclMemPoolResizableAllocator, 482

iBufferPostFence  
     OsclMemPoolResizableAllocator::MemPoolBufferInfo, 185

iBufferPreFence  
     OsclMemPoolResizableAllocator::MemPoolBufferInfo, 185

iBufferSize  
     OsclMemPoolResizableAllocator::MemPoolBufferInfo, 185

iBufRecv  
     RecvFromParam, 666  
     RecvParam, 668

iBufSend  
     SendParam, 669  
     SendToParam, 670

iBusy  
     PVActiveBase, 644

iCancel  
     OsclSocketServRequestQElem, 597

iCBase  
     OsclTrapStackItem, 632

iCheckFreeMemoryAvailable  
     OsclMemPoolResizableAllocator, 482

iCheckNextAvailable  
     OsclMemPoolResizableAllocator, 482

iCheckNextAvailableFreeChunk  
     OsclMemPoolFixedChunkAllocator, 473

iChunkAlignment  
     OsclMemPoolFixedChunkAllocator, 473

iChunkSize  
     OsclMemPoolFixedChunkAllocator, 473

iChunkSizeMemAligned  
     OsclMemPoolFixedChunkAllocator, 473

iComponentId  
     OsclComponentRegistryElement, 379

iComponentIdCounter  
     OsclComponentRegistry, 377

iContainer

OsclFileCacheBuffer, 437  
 OsclSocketMethod, 582  
 OsclSocketRequestAO, 588  
**i**  
 Id  
     OsclAsyncFileBuffer, 352  
     OsclSocketRequestAO, 587  
     PVThreadContext, 663  
 iData  
     OsclComponentRegistry, 377  
 iDebugLogger  
     OsclExecSchedulerCommonBase, 432  
     OsclMemPoolResizableAllocator, 482  
 iDefAlloc  
     OsclExecSchedulerCommonBase, 432  
 iDNSFxn  
     OsclIDNSMethod, 394  
 iDNSI  
     OsclIDNSRequestAO, 398  
 iDNSMethod  
     OsclIDNSRequestAO, 398  
 iDNSObserver  
     OsclIDNSMethod, 394  
 iDNSRequest  
     DNSRequestParam, 167  
 iDNSRequestAO  
     OsclIDNSMethod, 394  
 iDoStop  
     OsclExecSchedulerCommonBase, 432  
 iDoSuspend  
     OsclExecSchedulerCommonBase, 432  
 iEnableNullPtrReturn  
     OsclMemPoolFixedChunkAllocator, 473  
     OsclMemPoolResizableAllocator, 482  
 iEndAddr  
     OsclMemPoolResizableAllocator::MemPoolBufferInfo, 185  
 iErrAlloc  
     OsclSelect, 553  
 iErrorTrapImp  
     OsclExecSchedulerCommonBase, 432  
 iExecTimerQ  
     OsclExecSchedulerCommonBase, 432  
 iExpectedNumBlocksPerBuffer  
     OsclMemPoolResizableAllocator, 482  
 iFactory  
     OsclComponentRegistryElement, 379  
     OsclRegistryAccessElement, 541  
 iFilePosition  
     Oscl\_File::OsclFixedCacheParam, 443  
 iFlags  
     RecvFromParam, 666  
     RecvParam, 668  
     SendParam, 669  
     SendToParam, 670  
 iFreeMemChunkList  
     OsclMemPoolFixedChunkAllocator, 473  
 iFreeMemContextData  
     OsclMemPoolResizableAllocator, 482  
 iFreeMemPoolObserver  
     OsclMemPoolResizableAllocator, 482  
 ifront  
     Oscl\_Queue\_Base, 276  
 iFxn  
     DNSRequestParam, 167  
     SocketRequestParam, 673  
 iHead  
     OsclDoubleListBase, 402  
     OsclDoubleRunner, 404  
 iHeapCheck  
     OsclSelect, 553  
 iHigh  
     OsclInteger64Transport, 447  
 iHow  
     ShutdownParam, 671  
 iId  
     OsclComponentRegistryElement, 379  
     OsclIDNSMethod, 394  
     OsclIPSocketI, 451  
 iIsIn  
     TReadyQueLink, 688  
 iJumpData  
     OsclErrorTrapImp, 411  
 iLeave  
     OsclErrorTrapImp, 411  
 iLen  
     PVSockBufRecv, 661  
     PVSockBufSend, 662  
 iLength  
     OsclBuf, 373  
 iLogger  
     OsclIDNSMethod, 394  
     OsclIDNSRequestAO, 398  
     OsclExecSchedulerCommonBase, 432  
     OsclIPSocketI, 451  
     OsclSocketServIBase, 593  
 iLogPerfIndentStr  
     OsclExecSchedulerCommonBase, 432  
 iLogPerfIndentStrLen  
     OsclExecSchedulerCommonBase, 432  
 iLogPerfTotal  
     OsclExecSchedulerCommonBase, 432  
 iLow  
     OsclInteger64Transport, 447  
 iMaxLen  
     PVSockBufRecv, 661  
 iMaxLength  
     OsclBuf, 373  
 iMaxNewMemPoolBufferSz

OsclMemPoolResizableAllocator, 482  
iMemPool  
  OsclMemPoolFixedChunkAllocator, 473  
iMemPoolAligned  
  OsclMemPoolFixedChunkAllocator, 473  
iMemPoolAllocator  
  OsclMemPoolFixedChunkAllocator, 473  
iMemPoolBufferAllocator  
  OsclMemPoolResizableAllocator, 482  
iMemPoolBufferList  
  OsclMemPoolResizableAllocator, 482  
iMemPoolBufferNumLimit  
  OsclMemPoolResizableAllocator, 482  
iMemPoolBufferSize  
  OsclMemPoolResizableAllocator, 482  
iMemPoolPrevAllocBufferIndex  
  OsclMemPoolResizableAllocator, 482  
iMimeType  
  OsclRegistryAccessElement, 541  
iMultiMaxLen  
  RecvFromParam, 666  
iMutex  
  OsclComponentRegistry, 377  
iName  
  GetHostNameParam, 169  
  OsclExecSchedulerCommonBase, 432  
  PVActiveBase, 644  
iNativeAccessMode  
  OsclNativeFileParams, 495  
iNativeBufferSize  
  OsclNativeFileParams, 495  
iNativeMode  
  OsclExecSchedulerCommonBase, 432  
IncLogPerf  
  OsclExecSchedulerCommonBase, 429  
increment\_refcnt  
  BufferState, 146  
iNext  
  OsclDoubleLink, 399  
  OsclDoubleRunner, 404  
  OsclTrapStackItem, 632  
iNextAvailableContextData  
  OsclMemPoolFixedChunkAllocator, 473  
  OsclMemPoolResizableAllocator, 482  
iNextFreeBlock  
  OsclMemPoolResizableAllocator::MemPoolBlockInfo, 184  
  OsclMemPoolResizableAllocator::MemPoolBufferInfo, 185  
Init  
  OsclErrorTrap, 409  
  OsclInit, 446  
  OsclMem, 456  
  OsclScheduler, 549  
PVLogger, 648  
InitExecQ  
  OsclExecSchedulerCommonBase, 429  
Insert  
  OsclDoubleListBase, 402  
  OsclPriorityList, 499  
insert  
  Oscl\_Map, 257, 258  
  Oscl\_TagTree, 309  
  Oscl\_Vector, 317  
  Oscl\_Vector\_Base, 322  
insert\_element  
  Oscl\_Linked\_List, 246  
  Oscl\_Linked\_List\_Base, 250  
insert\_unique  
  Oscl\_Rb\_Tree, 281  
InsertAfter  
  OsclDoubleLink, 399  
InsertBefore  
  OsclDoubleLink, 399  
InsertHead  
  OsclDoubleList, 400  
  OsclDoubleListBase, 402  
InsertTail  
  OsclDoubleList, 400  
  OsclDoubleListBase, 402  
InstallScheduler  
  OsclExecSchedulerCommonBase, 429  
INT64  
  osclconfig\_unix\_android.h, 872  
  osclconfig\_unix\_common.h, 876  
int64  
  osclbase, 37  
INT64\_HILO  
  osclconfig\_unix\_android.h, 872  
  osclconfig\_unix\_common.h, 876  
interfaceAddr  
  OsclIpMReq, 448  
INTERNAL\_ERROR  
  BufFragStatusClass, 151  
internalLeave, 172  
  a, 172  
  osclerror, 97  
iNumAOAdded  
  OsclExecSchedulerCommonBase, 432  
iNumChunk  
  OsclMemPoolFixedChunkAllocator, 473  
iNumOfRun  
  OsclAsyncFile, 350  
iNumOfRunErr  
  OsclAsyncFile, 350  
iNumOutstanding  
  OsclMemPoolResizableAllocator::MemPoolBufferInfo, 185

iNumSessions  
     OsclComponentRegistry, 377

INVALID\_ACCESS\_ERROR  
     OsclProcStatus, 508

INVALID\_ARGUMENT\_ERROR  
     OsclProcStatus, 508

INVALID\_FUNCTION\_ERROR  
     OsclProcStatus, 508

INVALID\_HANDLE\_ERROR  
     OsclProcStatus, 508

INVALID\_ID  
     BufFragStatusClass, 151

INVALID\_OPERATION\_ERROR  
     OsclProcStatus, 508

INVALID\_PARAM\_ERROR  
     OsclProcStatus, 508

INVALID\_POINTER\_ERROR  
     OsclProcStatus, 508

INVALID\_PRIORITY\_ERROR  
     OsclProcStatus, 508

INVALID\_THREAD\_ERROR  
     OsclProcStatus, 508

INVALID\_THREAD\_ID\_ERROR  
     OsclProcStatus, 507

INVALID\_TYPE  
     Oscl\_FileFind, 226

iObserver  
     OsclIPSocketI, 451  
     OsclMemPoolFixedChunkAllocator, 473  
     OsclMemPoolResizableAllocator, 482

iOffset  
     OsclDoubleListBase, 402  
     OsclDoubleRunner, 404

iOpCount  
     OsclFileStatsItem, 442

iOsclBase  
     OsclSelect, 553

iOsclErrorTrap  
     OsclSelect, 553

iOsclLogger  
     OsclSelect, 553

iOsclMemory  
     OsclSelect, 553

iOsclScheduler  
     OsclSelect, 553

iOutputFile  
     OsclSelect, 553

iPacketLen  
     RecvFromParam, 666

iPacketSource  
     RecvFromParam, 666

ipAddr  
     OsclNetworkAddress, 496

iParam  
     OsclFileStatsItem, 442  
     OsclSocketRequestAO, 588

iParam2  
     OsclFileStatsItem, 442

iParamSize  
     OsclSocketRequestAO, 588

iParentBuffer  
     OsclMemPoolResizableAllocator::MemPoolBlockInfo, 184

iPrev  
     OsclDoubleLink, 399

iPrevAllocBlock  
     OsclMemPoolResizableAllocator::MemPoolBufferInfo, 185

iPrevFreeBlock  
     OsclMemPoolResizableAllocator::MemPoolBlockInfo, 184

iPriority  
     OsclPriorityLink, 498

iPtr  
     PVSockBufRecv, 661  
     PVSockBufSend, 662

iPVReadyQLink  
     PVActiveBase, 644

iQSize  
     ListenParam, 177

iReadyQ  
     OsclExecSchedulerCommonBase, 432

irear  
     Oscl\_Queue\_Base, 276

iRefCount  
     DNSRequestParam, 167  
     OsclMemPoolFixedChunkAllocator, 473  
     OsclMemPoolResizableAllocator, 482

iRequestedAvailableFreeMemSize  
     OsclMemPoolResizableAllocator, 482

iRequestedNextAvailableSize  
     OsclMemPoolResizableAllocator, 482

iResumeSem  
     OsclExecSchedulerCommonBase, 432

is\_writable  
     OSCL\_String, 299  
     OSCL\_wString, 337

is\_zero  
     TimeValue, 685

is\_zulu  
     TimeValue, 685

IsActive  
     PVLogger, 648

IsAdded  
     PVActiveBase, 642

isAllocNodePtr  
     MM\_AllocBlockHdr, 187

IsBusy

OsclActiveObject, 344  
OsclTimerObject, 620  
iSchedulerAlloc  
    OsclSelect, 553  
iSchedulerName  
    OsclSelect, 553  
iSchedulerReserve  
    OsclSelect, 553  
isCIEquivalentTo  
    StrCSumPtrLen, 675  
    StrPtrLen, 678  
    WStrPtrLen, 692  
isCIPrefixOf  
    StrPtrLen, 678  
iSelect  
    OsclSocketServRequestQElem, 597  
IsEmpty  
    OsclDoubleListBase, 402  
iSeqNum  
    TReadyQueLink, 688  
iServerError  
    OsclSocketServIBase, 593  
iServState  
    OsclSocketServIBase, 593  
isFixed  
    OsclFileCacheBuffer, 437  
IsHead  
    OsclDoubleList, 400  
    OsclPriorityList, 499  
IsIn  
    OsclReadyQ, 516  
    OsclTimerQ, 623  
IsInAnyQ  
    PVActiveBase, 642  
IsInstalled  
    OsclExecSchedulerCommonBase, 429  
IsInUse  
    OsclAsyncFileBuffer, 352  
iSize  
    Oscl\_File::OsclFixedCacheParam, 443  
isLetter  
    StrPtrLen, 678  
IsLocalData  
    MediaData, 180  
ISO8601TIME\_BUFFER\_SIZE  
    osclbase, 51  
ISO8601timeStrBuf  
    osclbase, 37  
ISO8601ToRFC822  
    osclbase, 40  
iSocket  
    OsclIPSocketI, 451  
iSocketError  
    OsclDNSRequestAO, 398  
                OsclSocketRequestAO, 588  
iSocketFxn  
    OsclSocketMethod, 582  
iSocketRequest  
    OsclSocketServRequestQElem, 597  
iSocketRequestAO  
    OsclSocketMethod, 582  
iSocketServ  
    OsclDNSIBase, 391  
    OsclIPSocketI, 451  
    OsclSocketIBase, 579  
IsOpen  
    OsclSocketIBase, 577  
IsReady  
    OsclDNSIBase, 390  
IsSameThreadContext  
    PVThreadContext, 663  
IsServConnected  
    OsclSocketServIBase, 593  
IsServerThread  
    OsclSocketServI, 591  
IsStarted  
    OsclExecSchedulerCommonBase, 429  
IsTail  
    OsclDoubleList, 400  
    OsclPriorityList, 499  
iStartAddr  
    OsclMemPoolResizableAllocat-  
        or::MemPoolBufferInfo, 185  
iStartTick  
    OsclFileStatsItem, 442  
iStatus  
    PVActiveBase, 644  
iStopper  
    OsclExecSchedulerCommonBase, 432  
iStopperCrit  
    OsclExecSchedulerCommonBase, 432  
IsUpdated  
    OsclFileCacheBuffer, 436  
iSuspended  
    OsclExecSchedulerCommonBase, 432  
IsValid  
    OsclAsyncFileBuffer, 352  
iTAny  
    OsclTrapStackItem, 632  
iterator  
    Oscl\_Linked\_List\_Base, 252  
    Oscl\_Map, 255  
    Oscl\_Rb\_Tree, 279  
    Oscl\_Rb\_Tree\_Iterator, 288  
    Oscl\_TagTree::iterator, 174  
    Oscl\_Vector, 315  
    OsclPriorityQueue, 501  
iThreadContext

OsclExecSchedulerCommonBase, 432  
 PVActiveBase, 644  
 iTimeCompareThreshold  
     OsclExecSchedulerCommonBase, 432  
 iTimeQueuedTicks  
     TReadyQueLink, 688  
 iTimeToRunTicks  
     TReadyQueLink, 688  
 iTicks  
     OsclFileStatsItem, 442  
 iTrapOperation  
     OsclTrapStackItem, 633  
 iTrapStack  
     OsclErrorTrapImp, 411  
 iVec  
     OsclComponentRegistryData, 378  
 iXferLen  
     SendParam, 669  
     SendToParam, 670  
  
 Join  
     osclio, 116  
     OsclIPSocketI, 450  
     OsclSocketI, 572  
     OsclSocketIBase, 578  
 JoinMulticastGroup  
     osclio, 116  
     OsclUDPSocketI, 637  
 Jump  
     OsclJump, 452  
  
 key\_comp  
     Oscl\_Map, 258  
 key\_compare  
     Oscl\_Map, 255  
 key\_type  
     Oscl\_Map, 255  
     Oscl\_Rb\_Tree, 279  
  
 largeasyncfilereadwrite\_test  
     Oscl\_File, 225  
 Leave  
     OsclError, 405  
 LeaveIfError  
     OsclError, 405  
 LeaveIfNull  
     OsclError, 405  
 Left  
     OsclPtrC, 511  
 left  
     Oscl\_Rb\_Tree\_Node\_Base, 292  
 len  
     OsclMemoryFragment, 469  
     StrPtrLen, 679  
  
 WStrPtrLen, 693  
 Length  
     OsclAsyncFileBuffer, 352  
     OsclBuf, 373  
     OsclPtr, 509  
     OsclPtrC, 511  
 length  
     BuffFragGroup, 150  
     OsclBinStream, 371  
     StrPtrLen, 678  
     WStrPtrLen, 693  
 lineNo  
     MM\_AllocInfo, 190  
     MM\_AllocQueryInfo, 193  
 link\_type  
     Oscl\_Rb\_Tree, 279  
     Oscl\_Rb\_Tree\_Const\_Iterator, 285  
     Oscl\_Rb\_Tree\_Iterator, 288  
     Oscl\_Rb\_Tree\_Node, 290  
 LinkedListElement, 176  
     data, 176  
     LinkedListElement, 176  
     next, 176  
 Listen  
     osclio, 117  
     OsclListenMethod, 453  
     OsclListenRequest, 454  
     OsclSocketI, 572  
     OsclSocketIBase, 578  
     OsclTCPSocketI, 604  
 ListenAsync  
     osclio, 117  
     OsclSocketIBase, 578  
     OsclTCPSocketI, 604  
 ListenParam, 177  
     iQSize, 177  
     ListenParam, 177  
 ListenRequest  
     OsclListenMethod, 453  
 little\_endian\_to\_host  
     osclbase, 40  
 localbuf  
     MediaData, 181  
 Lock  
     OsclLockBase, 455  
     OsclMutex, 489  
     OsclNullLock, 497  
     OsclThreadLock, 610  
 lockAndGetInstance  
     OsclSingletonRegistryEx, 569  
 Log  
     OsclFileStats, 441  
 log\_level\_type  
     AllPassFilter, 141

PVLogger, 646  
PVLoggerFilter, 653  
PVLoggerRegistry, 657  
LogAll  
  OsclFileStats, 441  
Logger  
  OsclSocketI, 572  
LogMsgBuffers  
  PVLogger, 648  
LogMsgBuffersV  
  PVLogger, 649  
LogMsgString  
  PVLogger, 649  
LogMsgStringV  
  PVLogger, 649  
LoopbackSocket  
  OsclSocketServI, 591  
lower\_bound  
  Oscl\_Map, 258  
  Oscl\_Rb\_Tree, 281, 282  
  
MakeAddr  
  OsclSocketI, 572  
MakeMulticastGroupInformation  
  OsclSocketI, 572  
map\_type  
  Oscl\_TagTree, 307  
mapit  
  Oscl\_TagTree::const\_iterator, 163  
  Oscl\_TagTree::iterator, 175  
mapiter  
  Oscl\_TagTree::const\_iterator, 162  
  Oscl\_TagTree::iterator, 174  
Match  
  OsclComponentRegistryElement, 379  
MAX\_THRDS\_REACHED\_ERROR  
  OsclProcStatus, 507  
MAX\_NUMBER\_OF\_BYTE\_PER\_UTF8  
  osclutil, 74  
max\_size  
  Oscl\_Map, 258  
  Oscl\_Rb\_Tree, 282  
MAX\_TOSCLFILEOFFSET\_VALUE  
  osclconfig\_io.h, 846  
maximum  
  Oscl\_Rb\_Tree\_Node\_Base, 292  
MaxLen  
  OsclNameString, 490  
maxsize  
  CFastRep, 156  
  CHeapRep, 158  
  CStackRep, 164  
mbchar  
  osclbase, 37  
  
MediaData, 178  
~MediaData, 179  
AddLocalFragment, 179  
available\_localbuf, 181  
Clear, 179  
GetAvailableBufferSize, 179  
GetLocalBufsize, 179  
GetLocalFragment, 179  
GetMediaFragment, 180  
GetMediaSize, 180  
GetNumMediaFrags, 180  
GetTimestamp, 180  
IsLocalData, 180  
localbuf, 181  
MediaData, 179  
num\_reserved\_fragments, 181  
SetTimestamp, 180  
timestamp, 181  
  
MediaStatusClass, 182  
MediaTimestamp  
  osclutil, 74  
MEM\_ALIGN\_SIZE  
  osclmemory, 56  
MemAllocator, 183  
~MemAllocator, 183  
allocate, 183  
deallocate, 183  
pointer, 183  
  
memoryPoolBufferMgmtOverhead  
  OsclMemPoolResizableAllocator, 479  
message\_id\_type  
  AllPassFilter, 141  
  PVLogger, 646  
  PVLoggerAppender, 652  
  PVLoggerFilter, 653  
  PVLoggerLayout, 655  
MethodDone  
  OsclDNSMethod, 393  
  OsclSocketMethod, 582  
MICROSECONDS  
  osclbase, 38  
MILLISECONDS  
  osclbase, 38  
MIN\_FENCE\_SIZE  
  osclmemory, 56  
minimum  
  Oscl\_Rb\_Tree\_Node\_Base, 292  
MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN  
  osclmemory, 56  
MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN  
  osclmemory, 56  
MM\_AllocBlockFence, 186  
  check\_fence, 186  
  fill\_fence, 186

MM\_AllocBlockFence, 186  
 MM\_AllocBlockFence, 186  
 pad, 186  
 MM\_AllocBlockHdr, 187  
     isAllocNodePtr, 187  
     MM\_AllocBlockHdr, 187  
     MM\_AllocBlockHdr, 187  
     pad, 187  
     pNode, 187  
     pRootNode, 187  
     setAllocNodeFlag, 187  
     size, 187  
 MM\_AllocInfo, 189  
     ~MM\_AllocInfo, 189  
     allocNum, 190  
     bSetFailure, 190  
     lineNo, 190  
     MM\_AllocInfo, 189  
     MM\_AllocInfo, 189  
     operator delete, 189  
     operator new, 189  
     pFileName, 190  
     pMemBlock, 190  
     pStatsNode, 190  
     size, 190  
 MM\_AllocNode, 191  
     ~MM\_AllocNode, 191  
     MM\_AllocNode, 191  
     MM\_AllocNode, 191  
     operator delete, 191  
     operator new, 191  
     pAllocInfo, 192  
     pNext, 192  
     pPrev, 192  
 MM\_AllocNodeAutoPtr  
     osclmemory, 64  
 MM\_AllocQueryInfo, 193  
     allocNum, 193  
     fileName, 193  
     lineNo, 193  
     pMemBlock, 193  
     size, 193  
     tag, 193  
 MM\_AUDIT\_ALLOC\_NODE\_ENABLE\_FLAG  
     osclmemory, 56  
 MM\_AUDIT\_ALLOC\_NODE\_SUPPORT  
     osclmemory, 56  
 MM\_AUDIT\_FAILURE\_SIMULATION\_-  
     SUPPORT  
     osclmemory, 56  
 MM\_AUDIT\_FENCE\_SUPPORT  
     osclmemory, 56  
 MM\_AUDIT\_FILL\_SUPPORT  
     osclmemory, 56  
 MM\_Audit\_Imp, 194  
 MM\_AUDIT\_INCLUDE\_ALL\_HEAP\_-  
     VALIDATION  
     osclmemory, 56  
 MM\_AUDIT\_POSTFILL\_FLAG  
     osclmemory, 56  
 MM\_AUDIT\_PREFILL\_FLAG  
     osclmemory, 56  
 MM\_AUDIT\_SUPPRESS\_FILENAME\_FLAG  
     osclmemory, 56  
 MM\_AUDIT\_VALIDATE\_ALL\_HEAP\_FLAG  
     osclmemory, 56  
 MM\_AUDIT\_VALIDATE\_BLOCK  
     osclmemory, 56  
 MM\_AUDIT\_VALIDATE\_ON\_FREE\_FLAG  
     osclmemory, 56  
 MM\_AuditOverheadStats, 195  
     per\_allocation\_overhead, 195  
     stats\_overhead, 195  
 MM\_FailInsertParam, 196  
     MM\_FailInsertParam, 196  
     MM\_FailInsertParam, 196  
     nAllocNum, 197  
     operator delete, 196  
     operator new, 196  
     reset, 196  
     xsubi, 197  
 MM\_Stats\_CB, 198  
     MM\_Stats\_CB, 198  
     MM\_Stats\_CB, 198  
     num\_child\_nodes, 198  
     operator delete, 198  
     operator new, 198  
     pStats, 198  
     tag, 198  
 MM\_Stats\_t, 200  
     MM\_Stats\_t, 200  
     MM\_Stats\_t, 200  
     numAllocFails, 201  
     numAllocs, 201  
     numBytes, 201  
     operator delete, 200  
     operator new, 200  
     peakNumAllocs, 201  
     peakNumBytes, 201  
     reset, 200  
     totalNumAllocs, 201  
     totalNumBytes, 201  
     update, 201  
 MM\_StatsNodeTagTreeType  
     osclmemory, 64  
 MMAuditCharAutoPtr  
     osclmemory, 64  
 MMAuditUint8AutoPtr

osclmemory, 64  
 Mode  
     OsclNativeFile, 493  
 mode  
     oscl\_stat\_buf, 296  
 MODE\_APPEND  
     Oscl\_File, 218  
 MODE\_BINARY  
     Oscl\_File, 218  
 MODE\_READ  
     Oscl\_File, 218  
 MODE\_READ\_PLUS  
     Oscl\_File, 218  
 MODE\_READWRITE  
     Oscl\_File, 218  
 MODE\_TEXT  
     Oscl\_File, 218  
 mode\_type  
     Oscl\_File, 218  
 move\_to\_end  
     Oscl\_Linked\_List, 246  
     Oscl\_Linked\_List\_Base, 251  
     Oscl\_MTLinkedList, 262  
 move\_to\_front  
     Oscl\_Linked\_List, 246  
     Oscl\_Linked\_List\_Base, 251  
     Oscl\_MTLinkedList, 262  
 MSEC\_PER\_SEC  
     osclbase, 51  
 MSEC\_TO\_MICROSEC  
     oscl\_socket\_method.h, 791  
 MsecToTicks  
     OsclTickCount, 611  
 multicastAddr  
     OsclIpMReq, 448  
 MUTEX\_LOCKED\_ERROR  
     OsclProcStatus, 508  
  
 nAllocNum  
     MM\_FailInsertParam, 197  
 New  
     Oscl\_DefAllocWithRefCounter, 212  
 NewL  
     OsclAcceptMethod, 340  
     OsclAsyncFile, 349  
     OsclAsyncFileBuffer, 352  
     OsclBindMethod, 354  
     OsclBuf, 373  
     OsclConnectMethod, 381  
     OsclIDNS, 385  
     OsclIDNSI, 388  
     OsclGetHostByNameMethod, 444  
     OsclListenMethod, 453  
     OsclRecvFromMethod, 518  
  
     OsclRecvMethod, 522  
     OsclSendMethod, 557  
     OsclSendToMethod, 560  
     OsclShutdownMethod, 565  
     OsclSocketI, 572  
     OsclSocketServ, 589  
     OsclSocketServI, 591  
     OsclTCPSocket, 601  
     OsclTCPSocketI, 604  
     OsclUDPSocket, 635  
     OsclUDPSocketI, 638  
 NewRequest  
     OsclDNSRequestAO, 397  
     OsclSocketRequestAO, 587  
 next  
     BufFragGroup, 150  
     LinkedListElement, 176  
 nextFragPtr  
     OsclBinStream, 371  
 NO\_PERMISSION\_ERROR  
     OsclProcStatus, 508  
 Node  
     Oscl\_TagTree::Node, 202  
 node  
     Oscl\_Rb\_Tree\_Const\_Iterator, 286  
     Oscl\_Rb\_Tree\_Iterator, 289  
 node\_ptr  
     Oscl\_TagTree, 307  
 node\_type  
     Oscl\_TagTree, 307  
 NOT\_ENOUGH\_MEMORY\_ERROR  
     OsclProcStatus, 507  
 NOT\_ENOUGH\_RESOURCES\_ERROR  
     OsclProcStatus, 507  
 NOT\_ENOUGH\_SPACE  
     BufFragStatusClass, 151  
 NOT\_IMPLEMENTED  
     OsclProcStatus, 508  
 NOT\_SUSPENDED\_ERROR  
     OsclProcStatus, 508  
 notifyfreeblockavailable  
     OsclMemPoolResizableAllocator, 479  
 notifyfreechunkavailable  
     OsclMemPoolFixedChunkAllocator, 472  
 notifyfreememoryavailable  
     OsclMemPoolResizableAllocator, 479  
 NTPTime, 204  
     get\_lower32, 206  
     get\_middle32, 206  
     get\_upper32, 206  
     get\_value, 206  
     NTPTime, 205, 206  
     operator+=, 206  
     operator-, 206

operator=, 207  
 set\_from\_system\_time, 207  
 set\_to\_current\_time, 207  
 TimeValue, 686  
 to\_system\_time, 207  
 NULL  
     osclbase, 34  
 NULL\_INPUT  
     BufFragStatusClass, 151  
 num\_child\_nodes  
     MM\_Stats\_CB, 198  
 num\_elements  
     Oscl\_Linked\_List\_Base, 252  
 num\_fragments  
     BufFragGroup, 150  
 num\_reserved\_fragments  
     MediaData, 181  
 numAllocFails  
     MM\_Stats\_t, 201  
 numAllocs  
     MM\_Stats\_t, 201  
 numBytes  
     MM\_Stats\_t, 201  
 numelems  
     Oscl\_Queue\_Base, 276  
     Oscl\_Vector\_Base, 324  
 numFrags  
     OsclBinStream, 371  
  
 octet  
     osclbase, 37  
 Offset  
     OsclAsyncFileBuffer, 352  
 Open  
     Oscl\_File, 220, 221  
     OsclAsyncFile, 349  
     OsclIDNSI, 388  
     OsclDNSIBase, 390  
     OsclFileCache, 435  
     OsclNativeFile, 493  
     OsclSocketI, 572  
     OsclSocketIBase, 578  
     OsclSocketServRequestList, 595  
 OpenSession  
     OsclComponentRegistry, 377  
 operator delete  
     HeapBase, 171  
     MM\_AllocInfo, 189  
     MM\_AllocNode, 191  
     MM\_FailInsertParam, 196  
     MM\_Stats\_CB, 198  
     MM\_Stats\_t, 200  
     OsclErrorAllocator, 408  
     osclmemory, 64  
  
 OsclMemStatsNode, 486  
 operator new  
     HeapBase, 171  
     MM\_AllocInfo, 189  
     MM\_AllocNode, 191  
     MM\_FailInsertParam, 196  
     MM\_Stats\_CB, 198  
     MM\_Stats\_t, 200  
     osclconfig\_global\_placement\_new.h, 841  
     OsclErrorAllocator, 408  
     osclmemory, 65  
     OsclMemStatsNode, 486  
 operator T \*  
     OsclDoubleRunner, 403  
 operator TheClass \*  
     osclbase, 40  
 operator<  
     OSCL\_String, 299  
     Oscl\_Tag, 303  
     OSCL\_wString, 337  
     OsclAOStatus, 347  
     TimeValue, 686  
 operator<<  
     OsclBinOStreamBigEndian, 365  
     OsclBinOStreamLittleEndian, 367  
 operator<=>  
     OSCL\_String, 300  
     OSCL\_wString, 337  
     OsclAOStatus, 347  
     TimeValue, 686  
 operator>  
     OSCL\_String, 300  
     OSCL\_wString, 337  
     OsclAOStatus, 347  
     TimeValue, 686  
 operator>>  
     OsclBinIStreamBigEndian, 359  
     OsclBinIStreamLittleEndian, 362  
 operator>=  
     OSCL\_String, 300  
     OSCL\_wString, 337  
     OsclAOStatus, 347  
     TimeValue, 686  
 operator\*  
     Oscl\_Rb\_Tree\_Const\_Iterator, 285  
     Oscl\_Rb\_Tree\_Iterator, 288  
     Oscl\_TagTree::const\_iterator, 162  
     Oscl\_TagTree::iterator, 174  
     osclbase, 40  
     OsclExclusiveArrayPtr, 414  
     OsclExclusivePtr, 417  
     OsclExclusivePtrA, 420  
     OSCLMemAutoPtr, 463  
     OsclSingletonEx, 567

OsclTLS, 624  
 OsclTLSEEx, 626  
 operator\*=  
     TimeValue, 685  
 operator()  
     Oscl\_Less, 242  
     Oscl\_Map::value\_compare, 690  
     Oscl\_Select1st, 293  
     Oscl\_Tag\_Base, 304  
 operator+  
     osclbase, 40  
 operator++  
     Oscl\_Rb\_Tree\_Const\_Iterator, 286  
     Oscl\_Rb\_Tree\_Iterator, 288, 289  
     Oscl\_TagTree::const\_iterator, 162  
     Oscl\_TagTree::iterator, 174  
     OsclDoubleRunner, 403  
 operator+=  
     NTPTime, 206  
     OSCL\_String, 299  
     OSCL\_wString, 337  
     TimeValue, 685  
 operator-  
     NTPTime, 206  
     osclbase, 40  
 operator->  
     Oscl\_Rb\_Tree\_Const\_Iterator, 286  
     Oscl\_Rb\_Tree\_Iterator, 289  
     Oscl\_TagTree::const\_iterator, 163  
     Oscl\_TagTree::iterator, 175  
     osclbase, 40  
     OsclExclusiveArrayPtr, 414  
     OsclExclusivePtr, 417  
     OsclExclusivePtrA, 420  
     OSCLMemAutoPtr, 463  
     OsclSingletonEx, 567  
     OsclTLS, 624  
     OsclTLSEEx, 626  
 operator--  
     Oscl\_Rb\_Tree\_Const\_Iterator, 286  
     Oscl\_Rb\_Tree\_Iterator, 289  
     Oscl\_TagTree::const\_iterator, 162, 163  
     Oscl\_TagTree::iterator, 174, 175  
     OsclDoubleRunner, 403  
 operator==  
     TimeValue, 685  
 operator=>  
     NTPTime, 207  
     OSCL\_FastString, 215  
     OSCL\_HeapStringA, 238  
     Oscl\_Map, 258  
     Oscl\_Rb\_Tree, 282  
     OSCL\_String, 300  
     Oscl\_TagTree, 309  
 Oscl\_Vector, 317  
 OSCL\_wFastString, 326  
 OSCL\_wHeapStringA, 332  
 OSCL\_wString, 337  
 OsclAOStatus, 347  
 osclbase, 40  
 OsclComponentRegistryElement, 379  
 OsclExclusiveArrayPtr, 415  
 OsclExclusivePtr, 418  
 OsclExclusivePtrA, 421  
 OSCLMemAutoPtr, 463  
 OsclRefCounterMemFrag, 530  
 osclutil, 78–80  
 OsclUuid, 639  
 StrCSumPtrLen, 675  
 StrPtrLen, 678  
 TimeValue, 685  
 WStrPtrLen, 693  
 operator==  
     Oscl\_Rb\_Tree\_Const\_Iterator, 286  
     Oscl\_Rb\_Tree\_Iterator, 289  
     OSCL\_String, 300  
     Oscl\_TagTree::const\_iterator, 163  
     Oscl\_TagTree::iterator, 175  
     OSCL\_wString, 337  
     OsclAOStatus, 347  
 osclbase, 41  
 OsclNetworkAddress, 496  
 OsclUuid, 639  
 StrCSumPtrLen, 676  
 StrPtrLen, 678  
 TimeValue, 686  
 WStrPtrLen, 693  
 optype  
     OSCL\_FastString, 214  
     OSCL\_HeapString, 234  
     OSCL\_HeapStringA, 236  
     OSCL\_StackString, 295  
     OSCL\_wFastString, 325  
     OSCL\_wHeapString, 329  
     OSCL\_wHeapStringA, 331  
     OSCL\_wStackString, 334  
 OSCL Base, 26  
 OSCL config, 23  
 OSCL Error, 93  
 OSCL Init, 134  
 OSCL IO, 103  
 OSCL Memory, 52  
 OSCL Proc, 130  
 OSCL Util, 68  
 Oscl\_File  
     ESymbianAccessMode\_Rfile, 219  
     ESymbianAccessMode\_RfileBuf, 219  
     MODE\_APPEND, 218

MODE\_BINARY, 218  
 MODE\_READ, 218  
 MODE\_READ\_PLUS, 218  
 MODE\_READWRITE, 218  
 MODE\_TEXT, 218  
 SEEKCUR, 218  
 SEEKEND, 218  
 SEEKSET, 218  
**OSCL\_FILE\_ATTRIBUTE\_ARCHIVE**  
 OsclFileManager, 439  
**OSCL\_FILE\_ATTRIBUTE\_DIRECTORY**  
 OsclFileManager, 439  
**OSCL\_FILE\_ATTRIBUTE\_HIDDEN**  
 OsclFileManager, 439  
**OSCL\_FILE\_ATTRIBUTE\_NORMAL**  
 OsclFileManager, 440  
**OSCL\_FILE\_ATTRIBUTE\_READONLY**  
 OsclFileManager, 439  
**OSCL\_FILE\_ATTRIBUTE\_SYSTEM**  
 OsclFileManager, 439  
**Oscl\_FileFind**  
 DIR\_TYPE, 226  
 E\_BUFFER\_TOO\_SMALL, 227  
 E\_INVALID\_ARG, 227  
 E\_INVALID\_STATE, 226  
 E\_MEMORY\_ERROR, 227  
 E\_NO\_MATCH, 227  
 E\_NOT\_IMPLEMENTED, 227  
 E\_OK, 226  
 E\_OTHER, 227  
 E\_PATH\_NOT\_FOUND, 227  
 E\_PATH\_TOO\_LONG, 227  
 FILE\_TYPE, 226  
 INVALID\_TYPE, 226  
**OSCL\_FILEMGMT\_E\_ALREADY\_EXISTS**  
 osclio, 108  
**OSCL\_FILEMGMT\_E\_NO\_MATCH**  
 osclio, 108  
**OSCL\_FILEMGMT\_E\_NOT\_EMPTY**  
 osclio, 108  
**OSCL\_FILEMGMT\_E\_NOT\_IMPLEMENTED**  
 osclio, 108  
**OSCL\_FILEMGMT\_E\_OK**  
 osclio, 108  
**OSCL\_FILEMGMT\_E\_PATH\_NOT\_FOUND**  
 osclio, 108  
**OSCL\_FILEMGMT\_E\_PATH\_TOO\_LONG**  
 osclio, 108  
**OSCL\_FILEMGMT\_E\_PERMISSION\_DENIED**  
 osclio, 108  
**OSCL\_FILEMGMT\_E\_SYS\_SPECIFIC**  
 osclio, 108  
**OSCL\_FILEMGMT\_E\_UNKNOWN**  
 osclio, 108

**OSCL\_FILEMGMT\_MODE\_DIR**  
 osclio, 108  
**OSCL\_FILEMGMT\_PERMS\_EXECUTE**  
 osclio, 109  
**OSCL\_FILEMGMT\_PERMS\_READ**  
 osclio, 109  
**OSCL\_FILEMGMT\_PERMS\_WRITE**  
 osclio, 109  
**Oscl\_Rb\_Tree\_Node\_Base**  
 black, 291  
 red, 291  
**oscl\_socket\_stats.h**  
 EOscSocket\_DataRecv, 803  
 EOscSocket\_DataSent, 803  
 EOscSocket\_Except, 803  
 EOscSocket\_OS, 803  
 EOscSocket\_Readable, 803  
 EOscSocket\_RequestAO\_Canceled, 802  
 EOscSocket\_RequestAO\_Error, 802  
 EOscSocket\_RequestAO\_Success, 802  
 EOscSocket\_RequestAO\_Timeout, 802  
 EOscSocket\_ServPoll, 803  
 EOscSocket\_ServRequestCancelIssued, 803  
 EOscSocket\_ServRequestComplete, 803  
 EOscSocket\_ServRequestIssued, 802  
 EOscSocket\_Writable, 803  
 EOscSocketServ\_LastEvent, 802  
 EOscSocketServ\_LoopsockError, 803  
 EOscSocketServ\_LoopsockOk, 803  
 EOscSocketServ\_SelectActivity, 802  
 EOscSocketServ\_SelectNoActivity, 802  
 EOscSocketServ\_SelectRescheduleAsap, 802  
 EOscSocketServ\_SelectReschedulePoll, 802  
**oscl\_socket\_types.h**  
 EPVIPAddMembership, 806  
 EPVIMulticastTTL, 806  
 EPVIPProtoIP, 806  
 EPVIPProtoTCP, 806  
 EPVIPTOS, 806  
 EPVSocket, 806  
 EPVSocket\_Last, 806  
 EPVSocketAccept, 806  
 EPVSocketBind, 806  
 EPVSocketBothShutdown, 806  
 EPVSocketCancel, 806  
 EPVSocketConnect, 806  
 EPVSocketFailure, 806  
 EPVSocketListen, 806  
 EPVSocketNotImplemented, 806  
 EPVSocketPending, 805  
 EPVSocketRecv, 806  
 EPVSocketRecvFrom, 806  
 EPVSocketRecvShutdown, 806  
 EPVSocketSend, 806

EPVSocketSendShutdown, 806  
EPVSocketSendTo, 806  
EPVSocketShutdown, 806  
EPVSocketSuccess, 805  
EPVSocketTimeout, 806  
EPVSockReuseAddr, 806  
**oscl\_thread.h**  
    EOsclThreadTerminate\_Join, 820  
    EOsclThreadTerminate\_Kill, 820  
    EOsclThreadTerminate\_NOP, 820  
    Start\_on\_creation, 819  
    Suspend\_on\_creation, 819  
    ThreadPriorityAboveNormal, 820  
    ThreadPriorityBelowNormal, 820  
    ThreadPriorityHighest, 820  
    ThreadPriorityLow, 820  
    ThreadPriorityLowest, 820  
    ThreadPriorityNormal, 820  
    ThreadPriorityTimeCritical, 820  
**OSCL\_ABS**  
    osclbase, 34  
**oscl\_abs**  
    osclutil, 80  
**OSCL\_AF\_INET**  
    osclconfig\_io.h, 846  
**Oscl\_Alloc**, 208  
    ~Oscl\_Alloc, 208  
    allocate, 208  
    allocate\_fl, 208  
**OSCL\_ALLOC\_DELETE**  
    osclmemory, 56  
**OSCL\_ALLOC\_NEW**  
    osclmemory, 57  
    oscl\_aostatus.h, 695  
**OSCL\_ARRAY\_DELETE**  
    osclmemory, 57  
**OSCL\_ARRAY\_NEW**  
    osclmemory, 57  
**OSCL\_ASCII\_CASE\_MAGIC\_BIT**  
    osclutil, 92  
**oscl\_asin**  
    osclutil, 80  
**OSCL\_ASSERT**  
    osclbase, 34  
**OSCL\_Assert**  
    osclbase, 41  
**oscl\_assert.h**, 696  
**OSCL\_ASSERT\_ALWAYS**  
    osclconfig.h, 834  
**oscl\_atan**  
    osclutil, 80  
**OSCL\_AUDIT\_ARRAY\_NEW**  
    osclmemory, 58  
**OSCL\_AUDIT\_CALLOC**  
    osclmemory, 58  
**OSCL\_AUDIT\_MALLOC**  
    osclmemory, 58  
**OSCL\_AUDIT\_NEW**  
    osclmemory, 59  
**OSCL\_AUDIT\_REALLOC**  
    osclmemory, 59  
**OSCL\_BAD\_ALLOC\_EXCEPTION\_CODE**  
    osclerror, 97  
**oscl\_base.h**, 697  
**oscl\_base\_alloc.h**, 698  
**oscl\_base\_macros.h**, 699  
**oscl\_bin\_stream.h**, 700  
**OSCL\_BYPASS\_MEMMGT**  
    osclconfig\_memory.h, 857  
**oscl\_byte\_order.h**, 701  
**OSCL\_BYTE\_ORDER\_BIG\_ENDIAN**  
    osclconfig\_ix86.h, 853  
**OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN**  
    osclconfig\_ix86.h, 853  
**OSCL\_CALLOC**  
    osclmemory, 60  
**oscl\_calloc**  
    osclmemory, 59  
**OSCL\_CATCH**  
    osclerror, 97  
**OSCL\_CATCH\_ANY**  
    osclerror, 97  
**OSCL\_CHAR\_IS\_SIGNED**  
    osclconfig\_limits\_typedefs.h, 856  
**OSCL\_CHAR\_IS\_UNSIGNED**  
    osclconfig\_limits\_typedefs.h, 856  
**oscl\_chdir**  
    osclio, 117  
**oscl\_CIstrcmp**  
    osclbase, 41  
**oscl\_CIstrncmp**  
    osclbase, 41, 42  
**OSCL\_CLEANUP\_BASE\_CLASS**  
    osclmemory, 60  
**OSCL\_CLOCK\_HAS\_DRIFT\_CORRECTION**  
    osclconfig\_util.h, 877  
**OSCL\_COND\_EXPORT\_REF**  
    osclbase, 35  
**OSCL\_COND\_IMPORT\_REF**  
    osclbase, 35  
**OSCL\_CONST\_CAST**  
    osclbase, 35  
**oscl\_cos**  
    osclutil, 80  
**Oscl\_Dalloc**, 209  
    ~Oscl\_Dalloc, 209  
    deallocate, 209  
**Oscl\_DefAlloc**, 210

allocate, 210  
 allocate\_fl, 210  
 deallocate, 210  
 oscl\_defalloc.h, 702  
 Oscl\_DefAllocWithRefCounter, 211  
     addRef, 211  
     Delete, 211  
     getCount, 211  
     New, 212  
     removeRef, 212  
 OSCL\_DEFAULT\_FREE  
     osclmemory, 60  
 OSCL\_DEFAULT\_MALLOC  
     osclmemory, 60  
 OSCL\_DELETE  
     osclmemory, 60  
 Oscl\_DeleteFile  
     Oscl\_FileServer, 230, 231  
 OSCL\_DISABLE\_INLINES  
     osclconfig\_unix\_android.h, 872  
     osclconfig\_unix\_common.h, 876  
 OSCL\_DISABLE\_WARNING\_RETURN\_-  
     TYPE\_NOT\_UDT  
     osclbase, 35  
     osclmemory, 61  
 OSCL\_DISABLE\_WARNING\_TRUNCATE\_-  
     DEBUG\_MESSAGE  
     osclbase, 35  
     osclmemory, 61  
 oscl\_dll.h, 703  
 OSCL\_DLL\_ENTRY\_POINT  
     osclbase, 35  
 OSCL\_DLL\_ENTRY\_POINT\_DEFAULT  
     osclbase, 36  
 oscl\_dns.h, 704  
 oscl\_dns\_gethostbyname.h, 705  
 oscl\_dns\_imp.h, 706  
 oscl\_dns\_imp\_base.h, 707  
 oscl\_dns\_imp\_pv.h, 708  
 oscl\_dns\_method.h, 709  
 oscl\_dns\_param.h, 710  
     TDNSRequestParamAllocator, 710  
 oscl\_dns\_request.h, 711  
 oscl\_dns\_tuneables.h, 712  
 oscl\_double\_list.h, 713  
 OSCL\_DYNAMIC\_CAST  
     osclbase, 36  
 OSCL\_ERR\_NONE  
     osclerror, 98  
 oscl\_errno.h, 714  
 oscl\_error.h, 715  
 oscl\_error\_allocator.h, 716  
 oscl\_error\_codes.h, 717  
 oscl\_error\_imp.h, 718  
     oscl\_error\_imp\_cppexceptions.h, 719  
     oscl\_error\_imp\_fatalerror.h, 720  
     oscl\_error\_imp\_jumps.h, 721  
     oscl\_error\_trapcleanup.h, 722  
     oscl\_exception.h, 723  
     OSCL\_EXCEPTSET\_FLAG  
         oscl\_socket\_serv\_imp\_pv.h, 799  
 oscl\_exclusive\_ptr.h, 724  
 oscl\_exp  
     osclutil, 81  
 OSCL\_EXPORT\_REF  
     osclconfig.h, 834  
 OSCL\_FastString, 213  
     ~OSCL\_FastString, 215  
     chartype, 214  
     get\_cstr, 215  
     get\_maxsize, 215  
     get\_size, 215  
     get\_str, 215  
     operator=, 215  
     otype, 214  
     OSCL\_FastString, 214  
     OSCL\_String, 216  
     OSCL\_FastString, 214  
     other\_chartype, 214  
     set, 215, 216  
     set\_length, 216  
 Oscl\_File, 217  
     ~Oscl\_File, 219  
     AddFixedCache, 219  
     asyncfilereadcancel\_test, 225  
     asyncfilereadwrite\_test, 225  
     Close, 219  
     EndOfFile, 220  
     Flush, 220  
     GetError, 220  
     Handle, 220  
     largeasyncfilereadwrite\_test, 225  
     mode\_type, 218  
     Open, 220, 221  
     Oscl\_File, 219  
     Oscl\_File, 219  
     Oscl\_FileServer, 231  
     OsclFileCache, 225  
     OsclFileCacheBuffer, 225  
     OsclFileHandle, 438  
     Read, 221  
     RemoveFixedCache, 221  
     Seek, 222  
     seek\_type, 218  
     SetAsyncReadBufferSize, 222  
     SetCacheObserver, 222  
     SetFileHandle, 222  
     SetLoggingEnable, 223

SetNativeAccessMode, 223  
SetNativeBufferSize, 223  
SetPVCacheSize, 223  
SetSize, 223  
SetSummaryStatsLoggingEnable, 224  
Size, 224  
Tell, 224  
TSymbianAccessMode, 218  
Write, 224  
Oscl\_File::OsclCacheObserver, 374  
  ~OsclCacheObserver, 374  
  ChooseCurCache, 374  
Oscl\_File::OsclFixedCacheParam, 443  
  Contains, 443  
  iFilePosition, 443  
  iSize, 443  
oscl\_file\_async\_read.h, 725  
OSCL\_FILE\_ATTRIBUTE\_TYPE  
  OsclFileManager, 439  
OSCL\_FILE\_BUFFER\_MAX\_SIZE  
  osclconfig\_io.h, 846  
oscl\_file\_cache.h, 726  
OSCL\_FILE\_CHAR\_PATH\_DELIMITER  
  osclio, 108  
oscl\_file\_dir\_utils.h, 727  
oscl\_file\_find.h, 729  
oscl\_file\_handle.h, 730  
oscl\_file\_io.h, 731  
oscl\_file\_manager.h, 732  
oscl\_file\_native.h, 733  
oscl\_file\_server.h, 734  
oscl\_file\_stats.h, 735  
OSCL\_FILE\_STATS\_LOGGER\_NODE  
  osclio, 108  
oscl\_file\_types.h, 736  
OSCL\_FILE\_WCHAR\_PATH\_DELIMITER  
  osclio, 108  
Oscl\_FileFind, 226  
  ~Oscl\_FileFind, 227  
  Close, 227  
  element\_type, 226  
  error\_type, 226  
  FindFirst, 227, 228  
  FindNext, 228  
  GetElementType, 229  
  GetLastError, 229  
  Oscl\_FileFind, 227  
  Oscl\_FileFind, 227  
OSCL\_FILEMGMT\_ERR\_TYPE  
  osclio, 108  
OSCL\_FILEMGMT\_MODES  
  osclio, 108  
OSCL\_FILEMGMT\_PERMS  
  osclio, 108  
Oscl\_FileServer, 230  
  ~Oscl\_FileServer, 230  
  Close, 230  
  Connect, 230  
  Oscl\_DeleteFile, 230, 231  
  Oscl\_File, 231  
  Oscl\_FileServer, 230  
  Oscl\_FileServer, 230  
  OsclNativeFile, 231  
OSCL\_FIRST\_CATCH  
  osclerror, 98  
OSCL\_FIRST\_CATCH\_ANY  
  osclerror, 98  
oscl\_floor  
  osclutil, 81  
OSCL\_FREE  
  osclmemory, 61  
oscl\_free  
  osclmemory, 61  
OSCL\_FSSTAT  
  osclio, 108  
oscl\_fsstat, 232  
  freebytes, 232  
  totalbytes, 232  
OSCL\_FUNCTION\_PTR  
  osclconfig\_compiler\_warnings.h, 837  
oscl\_getcwd  
  osclio, 118  
OSCL\_GetLastError  
  osclerror, 101  
OSCL\_HAS\_ANDROID\_FILE\_IO\_SUPPORT  
  osclconfig.h, 834  
OSCL\_HAS\_ANDROID\_SUPPORT  
  osclconfig, 25  
  osclconfig.h, 834  
OSCL\_HAS\_ANSI\_64BIT\_FILE\_IO\_SUPPORT  
  osclconfig\_io.h, 846  
OSCL\_HAS\_ANSI\_FILE\_IO\_SUPPORT  
  osclconfig\_io.h, 846  
OSCL\_HAS\_ANSI\_MATH\_SUPPORT  
  osclconfig\_unix\_android.h, 872  
  osclconfig\_unix\_common.h, 876  
OSCL\_HAS\_ANSI\_MEMORY\_FUNCS  
  osclconfig\_ansi\_memory.h, 835  
OSCL\_HAS\_ANSI\_STDIO\_SUPPORT  
  osclconfig\_unix\_android.h, 872  
  osclconfig\_unix\_common.h, 876  
OSCL\_HAS\_STDLIB\_SUPPORT  
  osclconfig\_unix\_android.h, 872  
  osclconfig\_unix\_common.h, 876  
OSCL\_HAS\_ANSI\_STRING\_SUPPORT  
  osclconfig\_unix\_android.h, 872  
  osclconfig\_unix\_common.h, 876  
OSCL\_HAS\_ANSI\_WIDE\_STRING\_SUPPORT

osclconfig\_unix\_android.h, 872  
 osclconfig\_unix\_common.h, 876

**OSCL\_HAS\_BASIC\_LOCK**  
 osclconfig\_unix\_android.h, 872  
 osclconfig\_unix\_common.h, 876

**OSCL\_HAS\_BERKELEY\_SOCKETS**  
 osclconfig, 25  
 osclconfig\_io.h, 846

**OSCL\_HAS\_ERRNO\_H**  
 osclconfig\_error.h, 838

**OSCL\_HAS\_EXCEPTIONS**  
 osclconfig\_error.h, 838

**OSCL\_HAS\_GLOB**  
 osclconfig\_io.h, 846

**OSCL\_HAS\_GLOBAL\_NEW\_DELETE**  
 osclconfig\_memory.h, 857

**OSCL\_HAS\_GLOBAL\_VARIABLE\_SUPPORT**  
 osclconfig\_unix\_android.h, 872  
 osclconfig\_unix\_common.h, 876

**OSCL\_HAS\_HEAP\_BASE\_SUPPORT**  
 osclconfig\_memory.h, 857

**OSCL\_HAS\_IPHONE\_SUPPORT**  
 osclconfig, 25  
 osclconfig\_unix\_android.h, 872

**OSCL\_HAS\_LARGE\_FILE\_SUPPORT**  
 osclconfig\_io.h, 846

**OSCL\_HAS\_MSWIN\_FILE\_IO\_SUPPORT**  
 osclconfig\_io.h, 846

**OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT**  
 osclconfig, 25  
 osclconfig\_unix\_android.h, 872  
 osclconfig\_unix\_common.h, 876

**OSCL\_HAS\_MSWIN\_SUPPORT**  
 osclconfig, 25  
 osclconfig\_unix\_android.h, 872  
 osclconfig\_unix\_common.h, 876

**OSCL\_HAS\_NATIVE\_FILE\_CACHE\_ENABLE**  
 osclconfig\_io.h, 846

**OSCL\_HAS\_NON\_PREEMPTIVE\_THREAD\_SUPPORT**  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866

**OSCL\_HAS\_PACKED\_STRUCT**  
 osclconfig.h, 834

**OSCL\_HAS\_PRAGMA\_PACK**  
 osclconfig.h, 834

**OSCL\_HAS\_PTHREAD\_SUPPORT**  
 osclconfig, 25  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866

**OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_FUNCS**  
 osclconfig, 25

**OSCL\_HAS\_PV\_C\_OS\_SUPPORT**  
 osclconfig, 25

**OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS**  
 osclconfig, 25

**OSCL\_HAS\_PV\_FILE\_CACHE**  
 osclconfig\_io.h, 846

**OSCL\_HAS\_RUNTIME\_LIB\_LOADING\_SUPPORT**  
 osclconfig\_lib.h, 854

**OSCL\_HAS\_SAVAJE\_IO\_SUPPORT**  
 osclconfig, 25

**OSCL\_HAS\_SAVAJE\_SUPPORT**  
 osclconfig, 25

**OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT**  
 osclconfig, 25  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866

**OSCL\_HAS\_SETJMP\_H**  
 osclconfig\_error.h, 838

**OSCL\_HAS\_SINGLETON\_SUPPORT**  
 osclbase, 36

**OSCL\_HAS\_SNPRINTF\_LONGLONG\_SUPPORT**  
 osclconfig\_util.h, 877

**OSCL\_HAS\_SOCKET\_SUPPORT**  
 osclconfig\_io.h, 846

**OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION**  
 osclconfig, 25  
 osclconfig\_io.h, 846

**OSCL\_HAS\_SYMBIAN\_DNS\_SERVER**  
 osclconfig, 25  
 osclconfig\_io.h, 846

**OSCL\_HAS\_SYMBIAN\_ERRORTRAP**  
 osclconfig, 25  
 osclconfig\_error.h, 838

**OSCL\_HAS\_SYMBIAN\_MATH**  
 osclconfig, 25  
 osclconfig\_util.h, 877

**OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS**  
 osclconfig, 25  
 osclconfig\_memory.h, 857

**OSCL\_HAS\_SYMBIAN\_SCHEDULER**  
 osclconfig, 25  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866

**OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER**  
 osclconfig, 25  
 osclconfig\_io.h, 846

**OSCL\_HAS\_SYMBIAN\_SUPPORT**  
 osclconfig, 25  
 osclconfig\_unix\_android.h, 872  
 osclconfig\_unix\_common.h, 876

**OSCL\_HAS\_SYMBIAN\_TIMERS**  
 osclconfig, 25  
 osclconfig\_util.h, 877

OSCL\_HAS\_THREAD\_SUPPORT  
    osclconfig\_proc\_unix\_android.h, 864  
    osclconfig\_proc\_unix\_common.h, 866

OSCL\_HAS\_TLS\_SUPPORT  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876

OSCL\_HAS\_UNICODE\_SUPPORT  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876

OSCL\_HAS\_UNIX\_SUPPORT  
    osclconfig, 25  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876

OSCL\_HAS\_UNIX\_TIME\_FUNCS  
    osclconfig, 25  
    osclconfig\_time.h, 867

oscl\_heapbase.h, 737

OSCL\_HeapString, 233  
    chartype, 234  
    optype, 234  
    OSCL\_String, 234  
    osclutil, 81, 82  
    other\_chartype, 234

OSCL\_HeapStringA, 235  
    ~OSCL\_HeapStringA, 237  
    chartype, 236  
    get\_cstr, 237  
    get\_maxsize, 237  
    get\_size, 237  
    get\_str, 238  
    operator=, 238  
    optype, 236  
    OSCL\_HeapStringA, 236, 237  
    OSCL\_String, 239  
    OSCL\_HeapStringA, 236, 237  
    other\_chartype, 236  
    set, 238, 239

OSCL\_IMPORT\_REF  
    osclconfig.h, 834

oscl\_init.h, 738

OSCL\_INLINE  
    osclbase, 36

Oscl\_Int64\_Utils, 240  
    get\_int64\_lower32, 241  
    get\_int64\_middle32, 241  
    get\_int64\_upper32, 241  
    get\_uint64\_lower32, 241  
    get\_uint64\_middle32, 241  
    get\_uint64\_upper32, 241  
    set\_int64, 241  
    set\_uint64, 241

oscl\_int64\_utils.h, 739  
    \_OsclInteger64Transport, 739

OSCL\_INTEGERS\_WORD\_ALIGNED

    osclconfig\_ix86.h, 853

OSCL\_IO\_EXTENSION\_MAXLEN  
    osclio, 108

OSCL\_IO\_FILENAME\_MAXLEN  
    osclio, 108

oscl\_ip\_socket.h, 740

OSCL\_IPPROTO\_IP  
    osclconfig\_io.h, 846

OSCL\_IPPROTO\_TCP  
    osclconfig\_io.h, 846

OSCL\_IPPROTO\_UDP  
    osclconfig\_io.h, 846

oscl\_isdigit  
    osclutil, 74

OSCL\_IsErrnoSupported  
    osclerror, 102

oscl\_isLetter  
    osclbase, 42

OSCL\_JUMP\_MAX\_JUMP\_MARKS  
    osclerror, 98

OSCL\_LAST\_CATCH  
    osclerror, 98

OSCL\_LEAVE  
    osclerror, 98

Oscl\_Less, 242  
    operator(), 242

Oscl\_Linked\_List, 243  
    ~Oscl\_Linked\_List, 243  
    add\_element, 244  
    add\_to\_front, 244  
    check\_list, 244  
    clear, 244  
    dequeue\_element, 244  
    get\_element, 245  
    get\_first, 245  
    get\_index, 245  
    get\_next, 245  
    get\_num\_elements, 246  
    insert\_element, 246  
    move\_to\_end, 246  
    move\_to\_front, 246  
    Oscl\_Linked\_List, 243  
    Oscl\_Linked\_List, 243  
    remove\_element, 247

oscl\_linked\_list.h, 741

Oscl\_Linked\_List\_Base, 248  
    ~Oscl\_Linked\_List\_Base, 249  
    add\_element, 249  
    add\_to\_front, 249  
    check\_list, 249  
    construct, 249  
    destroy, 249  
    get\_element, 250  
    get\_first, 250

get\_index, 250  
 get\_next, 250  
 head, 252  
 insert\_element, 250  
 iterator, 252  
 move\_to\_end, 251  
 move\_to\_front, 251  
 num\_elements, 252  
 remove\_element, 251  
 sizeof\_T, 252  
 tail, 252  
 oscl\_lock\_base.h, 742  
 oscl\_log  
     osclutil, 82  
 oscl\_log10  
     osclutil, 82  
 OSCL\_MALLOC  
     osclmemory, 61  
 oscl\_malloc  
     osclmemory, 61  
 Oscl\_Map, 253  
     begin, 256  
     clear, 256  
     const\_iterator, 255  
     const\_reference, 255  
     count, 256  
     empty, 256  
     end, 256  
     equal\_range, 257  
     erase, 257  
     find, 257  
     insert, 257, 258  
     iterator, 255  
     key\_comp, 258  
     key\_compare, 255  
     key\_type, 255  
     lower\_bound, 258  
     max\_size, 258  
     operator=, 258  
 Oscl\_Map, 255  
 Oscl\_Map, 255  
     pair\_citerator\_citerator, 255  
     pair\_iterator\_bool, 255  
     pair\_iterator\_iterator, 255  
     pointer, 255  
     reference, 255  
     self, 255  
     size, 259  
     size\_type, 255  
     upper\_bound, 259  
     value\_comp, 259  
     value\_type, 255  
 Oscl\_Map< Key, T, Alloc, Compare >  
     Oscl\_Map::value\_compare, 691  
             oscl\_map.h, 743  
             Oscl\_Map::value\_compare, 690  
                 comp, 691  
                 operator(), 690  
                 Oscl\_Map< Key, T, Alloc, Compare >, 691  
                 value\_compare, 690  
             oscl\_math.h, 744  
             OSCL\_MAX  
                 osclbase, 36  
             OSCL\_MAX\_TRAP\_LEVELS  
                 osclerror, 99  
             oscl\_media\_data.h, 745  
             oscl\_media\_status.h, 746  
             oscl\_mem.h, 747  
             oscl\_mem\_aligned\_size  
                 osclmemory, 65  
             oscl\_mem\_audit.h, 749  
             oscl\_mem\_audit\_internals.h, 751  
             oscl\_mem\_auto\_ptr.h, 752  
             oscl\_mem\_basic\_functions.h, 753  
             oscl\_mem\_inst.h, 754  
             oscl\_mem\_mempool.h, 755  
             oscl\_memcmp  
                 osclmemory, 65  
             oscl\_memcpy  
                 osclmemory, 65  
             OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN  
                 osclconfig\_unix\_android.h, 872  
                 osclconfig\_unix\_common.h, 876  
             oscl\_memmove  
                 osclmemory, 66  
             oscl\_memmove32  
                 osclmemory, 66  
             oscl\_memset  
                 osclmemory, 66  
             oscl\_memsize\_t  
                 osclconfig\_ansi\_memory.h, 835  
             OSCL\_MIN  
                 osclbase, 36  
             oscl\_mkdir  
                 osclio, 118  
 Oscl\_MTLLinked\_List, 260  
     ~Oscl\_MTLLinked\_List, 260  
     add\_element, 261  
     add\_to\_front, 261  
     dequeue\_element, 261  
     get\_element, 261  
     get\_index, 261  
     move\_to\_end, 262  
     move\_to\_front, 262  
     Oscl\_MTLLinked\_List, 260  
     Oscl\_MTLLinked\_List, 260  
     remove\_element, 262  
     the\_list, 263

oscl\_mutex.h, 756  
    OsclNoYieldMutex, 756  
oscl\_namestring.h, 757  
OSCL\_NATIVE\_INT64\_TYPE  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876  
OSCL\_NATIVE\_UINT64\_TYPE  
    osclconfig.h, 834  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876  
OSCL\_NATIVE\_WCHAR\_TYPE  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876  
OSCL\_NEW  
    osclmemory, 61  
oscl\_opaque\_type.h, 758  
Oscl\_Opaque\_Type\_Alloc, 264  
    ~Oscl\_Opaque\_Type\_Alloc, 264  
    allocate, 264  
    construct, 264  
    deallocate, 264  
    destroy, 264  
Oscl\_Opaque\_Type\_Alloc\_LL, 266  
    ~Oscl\_Opaque\_Type\_Alloc\_LL, 266  
    allocate, 266  
    compare\_data, 266  
    construct, 266  
    deallocate, 266  
    destroy, 267  
    get\_data, 267  
    get\_next, 267  
    set\_next, 267  
Oscl\_Opaque\_Type\_Compare, 268  
    ~Oscl\_Opaque\_Type\_Compare, 268  
    compare\_EQ, 268  
    compare\_LT, 268  
    swap, 268  
OSCL\_PACKED\_STRUCT\_BEGIN  
    osclconfig.h, 834  
OSCL\_PACKED\_STRUCT\_END  
    osclconfig.h, 834  
OSCL\_PACKED\_VAR  
    osclconfig.h, 834  
Oscl\_Pair, 270  
    first, 270  
    Oscl\_Pair, 270  
    Oscl\_Pair, 270  
    second, 270  
OSCL\_PERF\_SUMMARY\_LOGGING  
    osclproc, 132  
OSCL\_PLACEMENT\_NEW  
    osclmemory, 62  
oscl\_pow  
    osclutil, 82  
oscl\_priqueue.h, 759  
oscl\_priqueue\_test  
    OsclPriorityQueue, 503  
oscl\_procstatus.h, 760  
Oscl\_Queue, 271  
    ~Oscl\_Queue, 272  
    back, 272  
    clear, 273  
    const\_reference, 272  
    front, 273  
    Oscl\_Queue, 272  
    Oscl\_Queue, 272  
    pointer, 272  
    pop, 273  
    push, 273  
    reference, 272  
    size\_type, 272  
    value\_type, 272  
oscl\_queue.h, 761  
Oscl\_Queue\_Base, 274  
    ~Oscl\_Queue\_Base, 274  
    bufsize, 276  
    capacity, 275  
    clear, 275  
    construct, 275  
    destroy, 275  
    elems, 276  
    empty, 275  
    ifront, 276  
    irear, 276  
    numelems, 276  
    pop, 275  
    push, 275  
    reserve, 275  
    size, 276  
    sizeof\_T, 276  
oscl\_rand.h, 762  
OSCL\_RAND\_MAX  
    osclconfig\_util.h, 877  
Oscl\_Rb\_Tree, 277  
    ~Oscl\_Rb\_Tree, 279  
    begin, 279  
    clear, 280  
    const\_iterator, 279  
    const\_pointer, 279  
    const\_reference, 279  
    count, 280  
    difference\_type, 279  
    empty, 280  
    end, 280  
    equal\_range, 280  
    erase, 280, 281  
    find, 281  
    insert\_unique, 281

iterator, 279  
 key\_type, 279  
 link\_type, 279  
 lower\_bound, 281, 282  
 max\_size, 282  
 operator=, 282  
 Oscl\_Rb\_Tree, 279  
 Oscl\_Rb\_Tree, 279  
 pointer, 279  
 reference, 279  
 size, 282  
 size\_type, 279  
 upper\_bound, 282  
 value\_type, 279  
 Oscl\_Rb\_Tree\_Base, 283  
 base\_link\_type, 283  
 rebalance, 283  
 rebalance\_for\_erase, 283  
 rotate\_left, 283  
 rotate\_right, 283  
 Oscl\_Rb\_Tree\_Const\_Iterator, 284  
 base\_link\_type, 285  
 const\_iterator, 285  
 link\_type, 285  
 node, 286  
 operator\*, 285  
 operator++, 286  
 operator->, 286  
 operator--, 286  
 operator==, 286  
 Oscl\_Rb\_Tree\_Const\_Iterator, 285  
 Oscl\_Rb\_Tree\_Const\_Iterator, 285  
 pointer, 285  
 reference, 285  
 self, 285  
 value\_type, 285  
 Oscl\_Rb\_Tree\_Iterator, 287  
 base\_link\_type, 288  
 iterator, 288  
 link\_type, 288  
 node, 289  
 operator\*, 288  
 operator++, 288, 289  
 operator->, 289  
 operator--, 289  
 operator==, 289  
 Oscl\_Rb\_Tree\_Iterator, 288  
 Oscl\_Rb\_Tree\_Iterator, 288  
 pointer, 288  
 reference, 288  
 self, 288  
 value\_type, 288  
 Oscl\_Rb\_Tree\_Node, 290  
 link\_type, 290  
 value, 290  
 value\_type, 290  
 Oscl\_Rb\_Tree\_Node\_Base, 291  
 base\_link\_type, 291  
 color, 292  
 color\_type, 291  
 left, 292  
 maximum, 292  
 minimum, 292  
 parent, 292  
 RedBl, 291  
 right, 292  
 OSCL\_READSET\_FLAG  
 oscl\_socket\_serv\_imp\_pv.h, 799  
 OSCL\_REALLOC  
 osclmemory, 62  
 oscl\_realloc  
 osclmemory, 62  
 oscl\_refcounter.h, 763  
 oscl\_refcounter\_memfrag.h, 764  
 oscl\_registry\_access\_client.h, 765  
 oscl\_registry\_client.h, 766  
 oscl\_registry\_client\_impl.h, 767  
 oscl\_registry\_serv\_impl.h, 768  
 oscl\_registry\_serv\_impl\_global.h, 769  
 oscl\_registry\_serv\_impl\_tls.h, 770  
 oscl\_registry\_types.h, 771  
 OSCL\_REINTERPRET\_CAST  
 osclbase, 36  
 OSCL\_RELEASE\_BUILD  
 osclconfig.h, 834  
 oscl\_rename  
 osclio, 119  
 OSCL\_REQUEST\_ERR\_CANCEL  
 osclproc, 133  
 OSCL\_REQUEST\_ERR\_GENERAL  
 osclproc, 133  
 OSCL\_REQUEST\_ERR\_NONE  
 osclproc, 133  
 OSCL\_REQUEST\_PENDING  
 osclproc, 133  
 oscl\_rmdir  
 osclio, 119  
 oscl\_scheduler.h, 772  
 oscl\_scheduler\_ao.h, 773  
 oscl\_scheduler\_aobase.h, 774  
 oscl\_scheduler\_readyq.h, 775  
 oscl\_scheduler\_threadcontext.h, 776  
 oscl\_scheduler\_tuneables.h, 777  
 oscl\_scheduler\_types.h, 778  
 OSCL\_SD\_BOTH  
 osclconfig\_io.h, 846  
 OSCL\_SD\_RECEIVE  
 osclconfig\_io.h, 846

OSCL\_SD\_SEND  
    osclconfig\_io.h, 846

Oscl\_Select1st, 293  
    operator(), 293

oscl\_semaphore.h, 779

OSCL\_SetLastError  
    osclerror, 102

oscl\_shared\_ptr.h, 780

oscl\_sin  
    osclutil, 82

oscl\_singleton.h, 781

oscl\_snprintf  
    osclutil, 82, 83

oscl\_snprintf.h, 782

OSCL SOCK\_DGRAM  
    osclconfig\_io.h, 846

OSCL SOCK\_STREAM  
    osclconfig\_io.h, 846

oscl\_socket.h, 783

oscl\_socket\_accept.h, 784

oscl\_socket\_bind.h, 785

oscl\_socket\_connect.h, 786

oscl\_socket\_imp.h, 787

oscl\_socket\_imp\_base.h, 788

oscl\_socket\_imp\_pv.h, 789  
    PVSOCK\_ERR\_BAD\_PARAM, 789  
    PVSOCK\_ERR\_NOT\_IMPLEMENTED, 789  
    PVSOCK\_ERR\_NOT\_SUPPORTED, 789  
    PVSOCK\_ERR\_SERV\_NOT\_CONNECTED,  
        789  
    PVSOCK\_ERR SOCK\_NO\_SERV, 789  
    PVSOCK\_ERR SOCK\_NOT\_-  
        CONNECTED, 789  
    PVSOCK\_ERR SOCK\_NOT\_OPEN, 789

oscl\_socket\_listen.h, 790  
    OSCL\_SOCKET\_LISTEN\_H\_INCLUDEDd,  
        790

OSCL\_SOCKET\_LISTEN\_H\_INCLUDEDd  
    oscl\_socket\_listen.h, 790

oscl\_socket\_method.h, 791  
    MSEC\_TO\_MICROSEC, 791

oscl\_socket\_recv.h, 792

oscl\_socket\_recv\_from.h, 793

oscl\_socket\_request.h, 794

oscl\_socket\_send.h, 795

oscl\_socket\_send\_to.h, 796

oscl\_socket\_serv\_imp.h, 797

oscl\_socket\_serv\_imp\_base.h, 798

oscl\_socket\_serv\_imp\_pv.h, 799  
    OSCL\_EXCEPTSET\_FLAG, 799  
    OSCL\_READSET\_FLAG, 799  
    OSCL\_WRITESET\_FLAG, 799

oscl\_socket\_serv\_imp\_reqlist.h, 800

oscl\_socket\_shutdown.h, 801

oscl\_socket\_stats.h, 802  
    TOsclSocketServStatEvent, 802  
    TOsclSocketStatEvent, 802

oscl\_socket\_tuneables.h, 804  
    PV\_OSCL\_SOCKET\_STATS\_LOGGING,  
        804

PV\_SOCKET\_SERVER, 804

oscl\_socket\_types.h, 805  
    PVNETWORKADDRESS\_LEN, 805

TPVSocketEvent, 805

TPVSocketFxn, 806

TPVSocketOptionLevel, 806

TPVSocketOptionName, 806

TPVSocketShutdown, 806

OSCL\_SOCKOPT\_IP\_ADDMEMBERSHIP  
    osclconfig\_io.h, 846

OSCL\_SOCKOPT\_IP\_MULTICAST\_TTL  
    osclconfig\_io.h, 846

OSCL\_SOCKOPT\_IP\_TOS  
    osclconfig\_io.h, 846

OSCL\_SOCKOPT\_SOL\_REUSEADDR  
    osclconfig\_io.h, 846

OSCL\_SOL\_IP  
    osclconfig\_io.h, 846

OSCL\_SOL\_SOCKET  
    osclconfig\_io.h, 846

OSCL\_SOL\_TCP  
    osclconfig\_io.h, 846

OSCL\_SOL\_UDP  
    osclconfig\_io.h, 846

oscl\_sqrt  
    osclutil, 83

OSCL\_StackString, 294  
    chartype, 295  
    otype, 295  
    OSCL\_String, 295  
    osclutil, 83  
    other\_chartype, 295

oscl\_stat  
    osclio, 120

OSCL\_STAT\_BUF  
    osclio, 108

oscl\_stat\_buf, 296  
    mode, 296  
    perms, 296

oscl\_statsfs  
    osclio, 120

OSCL\_STATIC\_CAST  
    osclbase, 36

oscl\_stdstring.h, 807

oscl\_str\_escape\_xml  
    osclutil, 84

oscl\_str\_is\_valid\_utf8  
    osclutil, 84

oscl\_str\_need\_escape\_xml  
     osclutil, 85  
 oscl\_str\_ptr\_len.h, 809  
 oscl\_str\_truncate\_utf8  
     osclutil, 85  
 oscl\_str\_unescape\_uri  
     osclutil, 85, 86  
 oscl\_strcat  
     osclbase, 42, 43  
 oscl\_strchr  
     osclbase, 43  
 oscl\_strcmp  
     osclbase, 43, 44  
 OSCL\_StrError  
     osclerror, 102  
 OSCL\_String, 297  
     ~OSCL\_String, 298  
     append\_rep, 298  
     chartype, 298  
     get\_cstr, 298  
     get\_maxsize, 298  
     get\_size, 299  
     get\_str, 299  
     hash, 299  
     is\_writable, 299  
     operator<, 299  
     operator<=, 300  
     operator>, 300  
     operator>=, 300  
     operator+=, 299  
     operator=, 300  
     operator==, 300  
 OSCL\_String, 298  
 OSCL\_FastString, 216  
 OSCL\_HeapString, 234  
 OSCL\_HeapStringA, 239  
 OSCL\_StackString, 295  
 OSCL\_String, 298  
     read, 300  
     set\_len, 300  
     set\_rep, 300, 301  
     setrep\_to\_char, 301  
     write, 301  
 oscl\_string.h, 810  
 oscl\_string\_containers.h, 811  
 oscl\_string\_rep.h, 812  
 oscl\_string\_uri.h, 813  
 oscl\_string\_utf8.h, 814  
 oscl\_string\_utils.h, 815  
 oscl\_string\_xml.h, 816  
 oscl\_strlen  
     osclbase, 44  
 oscl\_strncat  
     osclbase, 45  
     osclbase  
         osclbase, 45, 46  
     oscl\_strncpy  
         osclbase, 46  
     oscl strrchr  
         osclbase, 47  
     oscl\_strset  
         osclbase, 47  
     oscl strstr  
         osclbase, 48  
 Oscl\_Tag, 302  
     ~Oscl\_Tag, 302  
     operator<, 303  
     Oscl\_Tag, 302  
     Oscl\_Tag, 302  
     tag, 303  
     tagAllocator, 303  
 Oscl\_Tag\_Base, 304  
     operator(), 304  
     size\_type, 304  
     tag\_ancestor, 304  
     tag\_base\_type, 304  
     tag\_base\_unit, 304  
     tag\_cmp, 304  
     tag\_copy, 305  
     tag\_depth, 305  
     tag\_len, 305  
 Oscl\_TagTree, 306  
     ~Oscl\_TagTree, 307  
     begin, 307  
     children\_type, 307  
     clear, 308  
     count, 308  
     empty, 308  
     end, 308  
     erase, 308  
     find, 309  
     insert, 309  
     map\_type, 307  
     node\_ptr, 307  
     node\_type, 307  
     operator=, 309  
     Oscl\_TagTree, 307  
     Oscl\_TagTree, 307  
     pair\_iterator\_bool, 307  
     size, 309  
     size\_type, 307  
     tag\_base\_type, 307  
     tag\_type, 307  
     value\_type, 307  
 oscl\_tagtree.h, 817  
 Oscl\_TagTree::const\_iterator, 161  
     const\_iterator, 162  
     mapit, 163

mapiter, 162  
operator\*, 162  
operator++, 162  
operator->, 163  
operator--, 162, 163  
operator==, 163  
pointer, 162  
reference, 162  
self, 162  
Oscl\_TagTree::iterator, 173  
    iterator, 174  
    mapit, 175  
    mapiter, 174  
    operator\*, 174  
    operator++, 174  
    operator->, 175  
    operator--, 174, 175  
    operator==, 175  
    pointer, 174  
    reference, 174  
    self, 174  
Oscl\_TagTree::Node, 202  
    children, 203  
    children\_type, 202  
    depth, 202  
    Node, 202  
    parent, 203  
    sort\_children, 202  
    tag, 203  
    value, 203  
Oscl\_TAlloc, 311  
    ~Oscl\_TAlloc, 312  
    address, 312  
    alloc\_and\_construct, 312  
    alloc\_and\_construct\_fl, 312  
    allocate, 312  
    allocate\_fl, 312  
    const\_pointer, 312  
    const\_reference, 312  
    construct, 312  
    deallocate, 312, 313  
    destroy, 313  
    destruct\_and\_dealloc, 313  
    pointer, 312  
    reference, 312  
    size\_type, 312  
    value\_type, 312  
Oscl\_TAlloc::rebind, 665  
    other, 665  
oscl\_tan  
    osclutil, 86  
OSCL\_TCHAR  
    osclbase, 37  
oscl\_tcp\_socket.h, 818  
OSCL\_TEMPLATED\_DESTRUCTOR\_CALL  
    osclconfig.h, 834  
oscl\_thread.h, 819  
    OsclThread\_State, 819  
    OsclThreadPriority, 819  
    TOsclThreadFuncPtr, 819  
    TOsclThreadTerminate, 820  
OSCL\_THREAD DECL  
    osclconfig\_proc\_unix\_android.h, 864  
    osclconfig\_proc\_unix\_common.h, 866  
oscl\_tickcount.h, 821  
oscl\_time.h, 822  
oscl\_timer.h, 824  
oscl\_tls.h, 825  
OSCL\_TLS\_BASE\_SLOTS  
    osclbase, 36  
OSCL\_TLS\_GET\_FUNC  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876  
OSCL\_TLS\_ID\_BASE\_LAST  
    osclbase, 51  
OSCL\_TLS\_ID\_ERRORHOOK  
    osclbase, 51  
OSCL\_TLS\_ID\_MAGICNUM  
    osclbase, 51  
OSCL\_TLS\_ID\_OSCLREGISTRY  
    osclbase, 51  
OSCL\_TLS\_ID\_PAYLOADPARSER  
    osclbase, 51  
OSCL\_TLS\_ID\_PVERRORTRAP  
    osclbase, 51  
OSCL\_TLS\_ID\_PVLOGGER  
    osclbase, 51  
OSCL\_TLS\_ID\_PVMFRECOGNIZER  
    osclbase, 51  
OSCL\_TLS\_ID\_PVSCHEDULER  
    osclbase, 51  
OSCL\_TLS\_ID\_SDPMEDIAPARSER  
    osclbase, 51  
OSCL\_TLS\_ID\_SQLITE3  
    osclbase, 51  
OSCL\_TLS\_ID\_TEST  
    osclbase, 51  
OSCL\_TLS\_ID\_WMDRM  
    osclbase, 51  
OSCL\_TLS\_IS\_KEYED  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876  
OSCL\_TLS\_KEY\_CREATE\_FUNC  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876  
OSCL\_TLS\_KEY\_DELETE\_FUNC  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876

OSCL\_TLS\_MAX\_SLOTS  
     osclbase, 36  
 OSCL\_TLS\_STORE\_FUNC  
     osclconfig\_unix\_android.h, 872  
     osclconfig\_unix\_common.h, 876  
 oscl\_tolower  
     osclbase, 48, 49  
 OSCL\_TRAP\_ALLOC\_NEW  
     osclmemory, 62  
 OSCL\_TRAP\_AUDIT\_NEW  
     osclmemory, 63  
 OSCL\_TRAP\_NEW  
     osclmemory, 63  
 OSCL\_TRAPSTACK\_POP  
     osclerror, 99  
 OSCL\_TRAPSTACK\_POPDEALLOC  
     osclerror, 99  
 OSCL\_TRAPSTACK\_PUSH  
     osclerror, 99  
 oscl\_tree.h, 826  
 OSCL\_TRY  
     osclerror, 99  
 OSCL\_TRY\_NO\_TLS  
     osclerror, 99  
 OSCL\_TStrPtrLen  
     osclutil, 74  
 oscl\_types.h, 827  
 oscl\_udp\_socket.h, 828  
 oscl\_UnicodeToUTF8  
     osclutil, 86  
 OSCL\_UNSIGNED\_CONST  
     osclconfig.h, 834  
 OSCL\_UNUSED\_ARG  
     osclbase, 36  
 OSCL\_UNUSED\_RETURN  
     osclbase, 37  
 oscl\_utf8conv.h, 829  
 oscl\_UTF8ToUnicode  
     osclutil, 87  
 oscl\_uuid.h, 830  
     BYTES\_IN\_UUID\_ARRAY, 830  
     EMPTY\_UUID, 830  
     OsclUid32, 830  
 oscl\_uuid\_utils.h, 831  
     PV\_CHAR\_CLOSE\_BRACKET, 831  
     PV\_CHAR\_COMMA, 831  
 Oscl\_Vector, 314  
     ~Oscl\_Vector, 315  
     back, 316  
     begin, 316  
     clear, 316  
     const\_iterator, 315  
     const\_reference, 315  
     destroy, 316  
     end, 316  
     erase, 317  
     front, 317  
     insert, 317  
     iterator, 315  
     operator=, 317  
     Oscl\_Vector, 315  
     Oscl\_Vector, 315  
     pointer, 315  
     pop\_back, 318  
     push\_back, 318  
     push\_front, 318  
     reference, 315  
     value\_type, 315  
 oscl\_vector.h, 832  
 Oscl\_Vector\_Base, 320  
     ~Oscl\_Vector\_Base, 321  
     assign\_vector, 321  
     bufsize, 324  
     capacity, 321  
     construct, 321  
     destroy, 321  
     elems, 324  
     empty, 322  
     erase, 322  
     insert, 322  
     numelems, 324  
     OsclPriorityQueueBase, 324  
     pop\_back, 322  
     push\_back, 323  
     push\_front, 323  
     reserve, 323  
     size, 323  
     sizeof\_T, 324  
 OSCL\_VIRTUAL\_BASE  
     osclbase, 37  
 oscl\_vsnprintf  
     osclutil, 87, 88  
 oscl\_wchar  
     osclbase, 37  
 OSCL\_wFastString, 325  
     ~OSCL\_wFastString, 326  
     chartype, 325  
     get\_cstr, 326  
     get\_maxsize, 326  
     get\_size, 326  
     get\_str, 326  
     operator=, 326  
     otype, 325  
     OSCL\_wFastString, 326  
     OSCL\_wString, 327  
     OSCL\_wFastString, 326  
     other\_chartype, 326  
     set, 327

set\_length, 327  
OSCL\_wHeapString, 328  
chartype, 329  
otype, 329  
OSCL\_wString, 329  
osclutil, 88  
other\_chartype, 329  
OSCL\_wHeapStringA, 330  
~OSCL\_wHeapStringA, 331  
chartype, 331  
get\_cstr, 331  
get\_maxsize, 331  
get\_size, 331  
get\_str, 332  
operator=, 332  
otype, 331  
OSCL\_wHeapStringA, 331  
OSCL\_wString, 332  
OSCL\_wHeapStringA, 331  
other\_chartype, 331  
set, 332  
OSCL\_WRITESET\_FLAG  
oscl\_socket\_serv\_imp\_pv.h, 799  
OSCL\_wStackString, 333  
chartype, 334  
otype, 334  
OSCL\_wString, 334  
osclutil, 88  
other\_chartype, 334  
OSCL\_wString, 335  
~OSCL\_wString, 336  
append\_rep, 336  
chartype, 336  
get\_cstr, 336  
get\_maxsize, 336  
get\_size, 336  
get\_str, 336  
hash, 336  
is\_writable, 337  
operator<, 337  
operator<=, 337  
operator>, 337  
operator>=, 337  
operator+=, 337  
operator=, 337  
operator==, 337  
OSCL\_wString, 336  
OSCL\_wFastString, 327  
OSCL\_wHeapString, 329  
OSCL\_wHeapStringA, 332  
OSCL\_wStackString, 334  
OSCL\_wString, 336  
read, 337  
set\_len, 337  
set\_rep, 337  
setrep\_to\_wide\_char, 337  
write, 338  
OSCL\_ZEROIZE  
osclproc, 132  
OsclAccept  
osclconfig\_io.h, 846  
OsclAcceptMethod, 339  
~OsclAcceptMethod, 339  
Accept, 339  
AcceptRequest, 339  
DiscardAcceptedSocket, 339  
GetAcceptedSocket, 340  
NewL, 340  
OsclAcceptRequest, 341  
Accept, 341  
OsclAcceptRequest, 341  
OsclSocketI, 574  
OsclActiveObject, 342  
~OsclActiveObject, 343  
AddToScheduler, 344  
Cancel, 344  
DoCancel, 344  
EPriorityHigh, 343  
EPriorityHighest, 343  
EPriorityIdle, 343  
EPriorityLow, 343  
EPriorityNominal, 343  
IsBusy, 344  
OsclActiveObject, 343  
OsclActivePriority, 343  
OsclExecSchedulerCommonBase, 430  
PendComplete, 344  
PendForExec, 344  
Priority, 344  
PVActiveBase, 644  
PVThreadContext, 664  
RemoveFromScheduler, 344  
RunError, 345  
RunIfNotReady, 345  
SetBusy, 345  
SetStatus, 345  
Status, 345  
StatusRef, 345  
OsclActivePriority  
OsclActiveObject, 343  
OsclAllocDestructDealloc, 346  
~OsclAllocDestructDealloc, 346  
OsclAny  
osclbase, 37  
OsclAOStatus, 347  
operator<, 347  
operator<=, 347  
operator>, 347

operator $\geq$ , 347  
 operator=, 347  
 operator==, 347  
 OsclAOStatus, 347  
 Value, 347  
 OsclAsyncFile, 348  
   ~OsclAsyncFile, 349  
   Close, 349  
   Delete, 349  
   EndOfFile, 349  
   Flush, 349  
   iNumOfRun, 350  
   iNumOfRunErr, 350  
   NewL, 349  
   Open, 349  
   Read, 349  
   Seek, 349  
   Size, 349  
   Tell, 349  
   Write, 349  
 OsclAsyncFileBuffer, 351  
   ~OsclAsyncFileBuffer, 352  
   Buffer, 352  
   CleanInUse, 352  
   HasThisOffset, 352  
   Id, 352  
   IsInUse, 352  
   IsValid, 352  
   Length, 352  
   NewL, 352  
   Offset, 352  
   SetInUse, 352  
   SetOffset, 352  
   StartAsyncRead, 352  
   UpdateData, 352  
 OsclAuditCB, 353  
 OsclBase  
   OsclTLSRegistry, 628  
 osclbase  
   ~OsclSharedPtr, 50  
   \_OSCL\_Abort, 38  
   ALLOC\_AND\_CONSTRUCT, 34  
   ALLOCATE, 34  
   big\_endian\_to\_host, 38  
   Bind, 38, 39  
   c\_bool, 37  
   CTIME\_BUFFER\_SIZE, 51  
   CtimeStrBuf, 37  
   EPV\_ARM\_GNUC, 34  
   EPV\_ARM\_MSEVC, 34  
   EPV\_ARM\_RVCT, 34  
   get\_count, 39  
   GetRefCounter, 39  
   GetRep, 39  
   host\_to\_big\_endian, 39  
   host\_to\_little\_endian, 39  
   int64, 37  
   ISO8601TIME\_BUFFER\_SIZE, 51  
   ISO8601timeStrBuf, 37  
   ISO8601ToRFC822, 40  
   little\_endian\_to\_host, 40  
   mbchar, 37  
   MICROSECONDS, 38  
   MILLISECONDS, 38  
   MSEC\_PER\_SEC, 51  
   NULL, 34  
   octet, 37  
   operator TheClass \*, 40  
   operator\*, 40  
   operator+, 40  
   operator-, 40  
   operator->, 40  
   operator=, 40  
   operator==, 41  
   OSCL\_ABS, 34  
   OSCL\_ASSERT, 34  
   OSCL\_Assert, 41  
   oscl\_CIstrcmp, 41  
   oscl\_CIstrncmp, 41, 42  
   OSCL\_COND\_EXPORT\_REF, 35  
   OSCL\_COND\_IMPORT\_REF, 35  
   OSCL\_CONST\_CAST, 35  
   OSCL\_DISABLE\_WARNING\_RETURN\_-  
     TYPE\_NOT\_UDT, 35  
   OSCL\_DISABLE\_WARNING\_-  
     TRUNCATE\_DEBUG\_MESSAGE,  
     35  
   OSCL\_DLL\_ENTRY\_POINT, 35  
   OSCL\_DLL\_ENTRY\_POINT\_DEFAULT, 36  
   OSCL\_DYNAMIC\_CAST, 36  
   OSCL\_HAS\_SINGLETON\_SUPPORT, 36  
   OSCL\_INLINE, 36  
   oscl\_isLetter, 42  
   OSCL\_MAX, 36  
   OSCL\_MIN, 36  
   OSCL\_REINTERPRET\_CAST, 36  
   OSCL\_STATIC\_CAST, 36  
   oscl\_strcat, 42, 43  
   oscl\_strchr, 43  
   oscl\_strcmp, 43, 44  
   oscl\_strlen, 44  
   oscl\_strncat, 45  
   oscl\_strncmp, 45, 46  
   oscl\_strncpy, 46  
   oscl strrchr, 47  
   oscl\_strset, 47  
   oscl\_strstr, 48  
   OSCL\_TCHAR, 37

OSCL\_TLS\_BASE\_SLOTS, 36  
OSCL\_TLS\_ID\_BASE\_LAST, 51  
OSCL\_TLS\_ID\_ERRORHOOK, 51  
OSCL\_TLS\_ID\_MAGICNUM, 51  
OSCL\_TLS\_ID\_OSCLREGISTRY, 51  
OSCL\_TLS\_ID\_PAYLOADPARSER, 51  
OSCL\_TLS\_ID\_PVERRORTRAP, 51  
OSCL\_TLS\_ID\_PVLOGGER, 51  
OSCL\_TLS\_ID\_PVMFRECOGNIZER, 51  
OSCL\_TLS\_ID\_PVSCHEDULER, 51  
OSCL\_TLS\_ID\_SDPMEDIAPARSER, 51  
OSCL\_TLS\_ID\_SQLITE3, 51  
OSCL\_TLS\_ID\_TEST, 51  
OSCL\_TLS\_ID\_WMDRM, 51  
OSCL\_TLS\_MAX\_SLOTS, 36  
oscl\_tolower, 48, 49  
OSCL\_UNUSED\_ARG, 36  
OSCL\_UNUSED\_RETURN, 37  
OSCL\_VIRTUAL\_BASE, 37  
oscl\_wchar, 37  
OsclAny, 37  
OsclFloat, 37  
OsclSharedPtr, 49  
OsclSizeT, 38  
PV8601TIME\_BUFFER\_SIZE, 51  
PV8601timeStrBuf, 38  
PV8601ToRFC822, 49  
PVMEM\_INST\_LEVEL, 37  
PVOsclBase\_Cleanup, 49  
PVOsclBase\_Init, 49  
RFC822ToPV8601, 49  
SECONDS, 38  
TimeUnits, 38  
TOsclTlsKey, 38  
uint, 38  
uint64, 38  
Unbind, 50  
unix\_ntp\_offset, 51  
USEC\_PER\_SEC, 51  
OsclBasicDateTimeStruct  
    osclconfig\_time.h, 867  
OsclBasicTimeStruct  
    osclconfig\_time.h, 867  
OsclBind  
    osclconfig\_io.h, 847  
OsclBindMethod, 354  
    ~OsclBindMethod, 354  
    Bind, 354  
    BindRequest, 354  
    NewL, 354  
OsclBindRequest, 355  
    Bind, 355  
    OsclBindRequest, 355  
OsclBinIStream, 356  
    ~OsclBinIStream, 356  
    get, 356  
    OsclBinIStream, 356  
    Read\_uint8, 356  
    OsclBinIStreamBigEndian, 358  
        operator>>, 359  
        OsclBinIStreamBigEndian, 359  
        Read, 359  
        Read\_uint16, 359  
        Read\_uint32, 359  
    OsclBinIStreamLittleEndian, 361  
        operator>>, 362  
        OsclBinIStreamLittleEndian, 362  
        Read\_uint16, 362  
        Read\_uint32, 362  
    OsclBinOStream, 363  
        ~OsclBinOStream, 363  
        OsclBinOStream, 363  
        write, 363  
    OsclBinOStreamBigEndian, 364  
        operator<<, 365  
        OsclBinOStreamBigEndian, 365  
        WriteUnsignedLong, 365  
        WriteUnsignedShort, 365  
    OsclBinOStreamLittleEndian, 366  
        operator<<, 367  
        OsclBinOStreamLittleEndian, 367  
        WriteUnsignedLong, 367  
        WriteUnsignedShort, 367  
    OsclBinStream, 368  
        Attach, 369  
        eof, 370  
        EOF\_STATE, 369  
        fail, 370  
        FAIL\_STATE, 369  
        firstFragPtr, 371  
        fragsLeft, 371  
        good, 370  
        GOOD\_STATE, 369  
        HaveRoomInCurrentBlock, 370  
        length, 371  
        nextFragPtr, 371  
        numFrags, 371  
        OsclBinStream, 369  
        pBasePosition, 371  
        PositionInBlock, 370  
        pPosition, 371  
        ReserveSpace, 370  
        Seek, 370  
        seekFromCurrentPosition, 370  
        specialFragBuffer, 371  
        state, 371  
        state\_t, 369  
        tellg, 371

OsclBuf, 372  
     Delete, 372  
     Des, 372  
     DesC, 372  
     iBuffer, 373  
     iLength, 373  
     iMaxLength, 373  
     Length, 373  
     NewL, 373  
     OsclBuf, 372  
 OsclCloseSocket  
     osclconfig\_io.h, 847  
 OsclCoeActiveScheduler  
     OsclExecSchedulerBase, 425  
     OsclExecSchedulerCommonBase, 430  
     PVThreadContext, 664  
 OsclCoeActiveSchedulerBase  
     PVThreadContext, 664  
 OsclCompareLess, 375  
     compare, 375  
 OsclComponentFactory  
     osclutil, 74  
 OsclComponentRegistry, 376  
     ~OsclComponentRegistry, 377  
     CloseSession, 377  
     FindExact, 377  
     FindHierarchical, 377  
     iComponentIdCounter, 377  
     iData, 377  
     iMutex, 377  
     iNumSessions, 377  
     OpenSession, 377  
     OsclComponentRegistry, 377  
     Register, 377  
     Unregister, 377  
 OsclComponentRegistryData, 378  
     Find, 378  
     iVec, 378  
 OsclComponentRegistryElement, 379  
     ~OsclComponentRegistryElement, 379  
     iComponentId, 379  
     iFactory, 379  
     iId, 379  
     Match, 379  
     operator=, 379  
     OsclComponentRegistryElement, 379  
 osclconfig  
     \_\_int16\_check\_\_, 25  
     \_\_int32\_check\_\_, 25  
     \_\_int8\_check\_\_, 25  
     \_\_uint16\_check\_\_, 25  
     \_\_uint32\_check\_\_, 25  
     \_\_uint8\_check\_\_, 25  
     OSCL\_HAS\_ANDROID\_SUPPORT, 25  
     OSCL\_HAS\_BERKELEY\_SOCKETS, 25  
     OSCL\_HAS\_IPHONE\_SUPPORT, 25  
     OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT, 25  
     OSCL\_HAS\_MSWIN\_SUPPORT, 25  
     OSCL\_HAS\_PTHREAD\_SUPPORT, 25  
     OSCL\_HAS\_PV\_C\_OS\_API\_MEMORY\_FUNCS, 25  
     OSCL\_HAS\_PV\_C\_OS\_SUPPORT, 25  
     OSCL\_HAS\_PV\_C\_OS\_TIME\_FUNCS, 25  
     OSCL\_HAS\_SAVAJE\_IO\_SUPPORT, 25  
     OSCL\_HAS\_SAVAJE\_SUPPORT, 25  
     OSCL\_HAS\_SEM\_TIMEDWAIT\_SUPPORT, 25  
     OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_IO\_FUNCTION, 25  
     OSCL\_HAS\_SYMBIAN\_DNS\_SERVER, 25  
     OSCL\_HAS\_SYMBIAN\_ERRORTRAP, 25  
     OSCL\_HAS\_SYMBIAN\_MATH, 25  
     OSCL\_HAS\_SYMBIAN\_MEMORY\_FUNCS, 25  
     OSCL\_HAS\_SYMBIAN\_SCHEDULER, 25  
     OSCL\_HAS\_SYMBIAN\_SOCKET\_SERVER, 25  
     OSCL\_HAS\_SYMBIAN\_SUPPORT, 25  
     OSCL\_HAS\_SYMBIAN\_TIMERS, 25  
     OSCL\_HAS\_UNIX\_SUPPORT, 25  
     OSCL\_HAS\_UNIX\_TIME\_FUNCS, 25  
 osclconfig.h, 833  
     \_\_TFS\_\_, 834  
     OSCL\_ASSERT\_ALWAYS, 834  
     OSCL\_EXPORT\_REF, 834  
     OSCL\_HAS\_ANDROID\_FILE\_IO\_SUPPORT, 834  
     OSCL\_HAS\_ANDROID\_SUPPORT, 834  
     OSCL\_HAS\_PACKED\_STRUCT, 834  
     OSCL\_HAS\_PRAGMA\_PACK, 834  
     OSCL\_IMPORT\_REF, 834  
     OSCL\_NATIVE\_UINT64\_TYPE, 834  
     OSCL\_PACKED\_STRUCT\_BEGIN, 834  
     OSCL\_PACKED\_STRUCT\_END, 834  
     OSCL\_PACKED\_VAR, 834  
     OSCL\_RELEASE\_BUILD, 834  
     OSCL\_TEMPLATED\_DESTRUCTOR\_CALL, 834  
     OSCL\_UNSIGNED\_CONST, 834  
 osclconfig\_ansi\_memory.h, 835  
     OSCL\_HAS\_ANSI\_MEMORY\_FUNCS, 835  
     oscl\_memsize\_t, 835  
 osclconfig\_check.h, 836  
 osclconfig\_compiler\_warnings.h, 837  
     OSCL\_FUNCTION\_PTR, 837  
 osclconfig\_error.h, 838  
     OSCL\_HAS\_ERRNO\_H, 838

OSCL\_HAS\_EXCEPTIONS, 838  
 OSCL\_HAS\_SETJMP\_H, 838  
 OSCL\_HAS\_SYMBIAN\_ERRORTRAP, 838  
 osclconfig\_error\_check.h, 839  
 osclconfig\_global\_new\_delete.h, 840  
 osclconfig\_global\_placement\_new.h, 841  
     operator new, 841  
 osclconfig\_io.h, 842  
     MAX\_TOSCLFILEOFFSET\_VALUE, 846  
     OSCL\_AF\_INET, 846  
     OSCL\_FILE\_BUFFER\_MAX\_SIZE, 846  
     OSCL\_HAS\_ANSI\_64BIT\_FILE\_IO\_-  
         SUPPORT, 846  
     OSCL\_HAS\_ANSI\_FILE\_IO\_SUPPORT,  
         846  
     OSCL\_HAS\_BERKELEY\_SOCKETS, 846  
     OSCL\_HAS\_GLOB, 846  
     OSCL\_HAS\_LARGE\_FILE\_SUPPORT, 846  
     OSCL\_HAS\_MSWIN\_FILE\_IO\_SUPPORT,  
         846  
     OSCL\_HAS\_NATIVE\_FILE\_CACHE\_-  
         ENABLE, 846  
     OSCL\_HAS\_PV\_FILE\_CACHE, 846  
     OSCL\_HAS\_SOCKET\_SUPPORT, 846  
     OSCL\_HAS\_SYMBIAN\_COMPATIBLE\_-  
         IO\_FUNCTION, 846  
     OSCL\_HAS\_SYMBIAN\_DNS\_SERVER,  
         846  
     OSCL\_HAS\_SYMBIAN\_SOCKET\_-  
         SERVER, 846  
     OSCL IPPROTO\_IP, 846  
     OSCL IPPROTO\_TCP, 846  
     OSCL IPPROTO\_UDP, 846  
     OSCL\_SD\_BOTH, 846  
     OSCL\_SD\_RECEIVE, 846  
     OSCL\_SD\_SEND, 846  
     OSCL SOCK\_DGRAM, 846  
     OSCL SOCK\_STREAM, 846  
     OSCL SOCKOPT\_IP\_ADDMEMBERSHIP,  
         846  
     OSCL SOCKOPT\_IP\_MULTICAST\_TTL,  
         846  
     OSCL SOCKOPT\_IP\_TOS, 846  
     OSCL SOCKOPT\_SOL\_REUSEADDR, 846  
     OSCL SOL\_IP, 846  
     OSCL SOL\_SOCKET, 846  
     OSCL SOL\_TCP, 846  
     OSCL SOL\_UDP, 846  
     OsclAccept, 846  
     OsclBind, 847  
     OsclCloseSocket, 847  
     OsclConnect, 847  
     OsclConnectComplete, 847  
     OsclGetAsyncSockErr, 847  
     OsclGetDottedAddr, 847  
     OsclGetDottedAddrVector, 848  
     OsclGethostbyname, 848  
     OsclGetPeerName, 848  
     OsclJoin, 848  
     OsclListen, 849  
     OsclMakeInAddr, 849  
     OsclMakeSockAddr, 849  
     OsclPipe, 849  
     OsclReadFD, 849  
     OsclRecv, 849  
     OsclRecvFrom, 849  
     OsclSend, 849  
     OsclSendTo, 850  
     OsclSetNonBlocking, 850  
     OsclSetRecvBufferSize, 850  
     OsclSetSockOpt, 850  
     OsclShutdown, 850  
     OsclSocket, 850  
     OsclSocketCleanup, 851  
     OsclSocketSelect, 851  
     OsclSocketStartup, 851  
     OsclUnMakeInAddr, 851  
     OsclUnMakeSockAddr, 851  
     OsclValidInetAddr, 851  
     OsclWriteFD, 851  
     TIpMReq, 851  
     TOsclFileOffset, 851  
     TOsclHostent, 851  
     TOsclSockAddr, 851  
     TOsclSockAddrLen, 851  
     TOsclSocket, 851  
 osclconfig\_io\_check.h, 852  
     \_\_verify\_\_TOsclFileOffset\_\_defined\_\_, 852  
 osclconfig\_ix86.h, 853  
     OSCL\_BYTE\_ORDER\_BIG\_ENDIAN, 853  
     OSCL\_BYTE\_ORDER\_LITTLE\_ENDIAN,  
         853  
     OSCL\_INTEGERS\_WORD\_ALIGNED, 853  
 osclconfig\_lib.h, 854  
     OSCL\_HAS\_RUNTIME\_LIB\_LOADING\_-  
         SUPPORT, 854  
     PV\_DYNAMIC\_LOADING\_CONFIG\_-  
         FILE\_PATH, 854  
     PV\_RUNTIME\_LIB\_FILENAME\_-  
         EXTENSION, 854  
 osclconfig\_lib\_check.h, 855  
 osclconfig\_limits\_typedefs.h, 856  
     OSCL\_CHAR\_IS\_SIGNED, 856  
     OSCL\_CHAR\_IS\_UNSIGNED, 856  
 osclconfig\_memory.h, 857  
     OSCL\_BYPASS\_MEMMGT, 857  
     OSCL\_HAS\_GLOBAL\_NEW\_DELETE, 857  
     OSCL\_HAS\_HEAP\_BASE\_SUPPORT, 857

OSCL\_HAS\_SYMBIAN\_MEMORY\_-  
 FUNCS, 857  
 PVMEM\_INST\_LEVEL, 857  
 osclconfig\_memory\_check.h, 858  
 osclconfig\_no\_os.h, 859  
 osclconfig\_proc.h, 860  
 osclconfig\_proc\_check.h, 861  
   \_\_verify\_TOsclConditionObject\_defined\_-  
     \_, 861  
   \_\_verify\_TOsclMutexObject\_defined\_\_,  
     861  
   \_\_verify\_TOsclSemaphoreObject\_-  
     defined\_, 861  
   \_\_verify\_TOsclThreadFuncArg\_defined\_\_,  
     861  
   \_\_verify\_TOsclThreadFuncRet\_defined\_\_,  
     861  
   \_\_verify\_TOsclThreadId\_defined\_\_, 861  
   \_\_verify\_TOsclThreadObject\_defined\_\_,  
     862  
 osclconfig\_proc\_unix\_android.h, 863  
   OSCL\_HAS\_NON\_PREEMPTIVE\_-  
     THREAD\_SUPPORT, 864  
   OSCL\_HAS\_PTHREAD\_SUPPORT, 864  
   OSCL\_HAS\_SEM\_TIMEDWAIT\_-  
     SUPPORT, 864  
   OSCL\_HAS\_SYMBIAN\_SCHEDULER, 864  
   OSCL\_HAS\_THREAD\_SUPPORT, 864  
   OSCL\_THREAD\_DECL, 864  
 TOsclConditionObject, 864  
 TOsclMutexObject, 864  
 TOsclSemaphoreObject, 864  
 TOsclThreadFuncArg, 864  
 TOsclThreadFuncRet, 864  
 TOsclThreadId, 864  
 TOsclThreadObject, 864  
 osclconfig\_proc\_unix\_common.h, 865  
   OSCL\_HAS\_NON\_PREEMPTIVE\_-  
     THREAD\_SUPPORT, 866  
   OSCL\_HAS\_PTHREAD\_SUPPORT, 866  
   OSCL\_HAS\_SEM\_TIMEDWAIT\_-  
     SUPPORT, 866  
   OSCL\_HAS\_SYMBIAN\_SCHEDULER, 866  
   OSCL\_HAS\_THREAD\_SUPPORT, 866  
   OSCL\_THREAD\_DECL, 866  
 TOsclConditionObject, 866  
 TOsclMutexObject, 866  
 TOsclSemaphoreObject, 866  
 TOsclThreadFuncArg, 866  
 TOsclThreadFuncRet, 866  
 TOsclThreadId, 866  
 TOsclThreadObject, 866  
 osclconfig\_time.h, 867  
   OSCL\_HAS\_UNIX\_TIME\_FUNCS, 867  
 OsclBasicDateStruct, 867  
 OsclBasicTimeStruct, 867  
 osclconfig\_time\_check.h, 868  
   \_\_Validate\_BasicTimeDateStruct\_\_, 868  
   \_\_Validate\_BasicTimeStruct\_\_, 868  
 osclconfig\_unix\_android.h, 869  
   \_STRLIT, 872  
   \_STRLIT\_CHAR, 872  
   \_STRLIT\_WCHAR, 872  
 INT64, 872  
 INT64\_HILO, 872  
 OSCL\_DISABLE\_INLINES, 872  
 OSCL\_HAS\_ANSI\_MATH\_SUPPORT, 872  
 OSCL\_HAS\_ANSI\_STDIO\_SUPPORT, 872  
 OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT, 872  
 OSCL\_HAS\_ANSI\_STRING\_SUPPORT,  
   872  
 OSCL\_HAS\_ANSI\_WIDE\_STRING\_-  
   SUPPORT, 872  
 OSCL\_HAS\_BASIC\_LOCK, 872  
 OSCL\_HAS\_GLOBAL\_VARIABLE\_-  
   SUPPORT, 872  
 OSCL\_HAS\_IPHONE\_SUPPORT, 872  
 OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT,  
   872  
 OSCL\_HAS\_MSWIN\_SUPPORT, 872  
 OSCL\_HAS\_SYMBIAN\_SUPPORT, 872  
 OSCL\_HAS\_TLS\_SUPPORT, 872  
 OSCL\_HAS\_UNICODE\_SUPPORT, 872  
 OSCL\_HAS\_UNIX\_SUPPORT, 872  
 OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN,  
   872  
 OSCL\_NATIVE\_INT64\_TYPE, 872  
 OSCL\_NATIVE\_UINT64\_TYPE, 872  
 OSCL\_NATIVE\_WCHAR\_TYPE, 872  
 OSCL\_TLS\_GET\_FUNC, 872  
 OSCL\_TLS\_IS\_KEYED, 872  
 OSCL\_TLS\_KEY\_CREATE\_FUNC, 872  
 OSCL\_TLS\_KEY\_DELETE\_FUNC, 872  
 OSCL\_TLS\_STORE\_FUNC, 872  
 TOsclBasicLockObject, 872  
 TOsclTlsKey, 872  
 INT64, 872  
 INT64\_HILO, 872  
 osclconfig\_unix\_common.h, 873  
   \_STRLIT, 876  
   \_STRLIT\_CHAR, 876  
   \_STRLIT\_WCHAR, 876  
 INT64, 876  
 INT64\_HILO, 876  
 OSCL\_DISABLE\_INLINES, 876  
 OSCL\_HAS\_ANSI\_MATH\_SUPPORT, 876  
 OSCL\_HAS\_ANSI\_STDIO\_SUPPORT, 876  
 OSCL\_HAS\_ANSI\_STDLIB\_SUPPORT, 876

OSCL\_HAS\_ANSI\_STRING\_SUPPORT, 876  
 OSCL\_HAS\_ANSI\_WIDE\_STRING\_SUPPORT, 876  
 OSCL\_HAS\_BASIC\_LOCK, 876  
 OSCL\_HAS\_GLOBAL\_VARIABLE\_SUPPORT, 876  
 OSCL\_HAS\_MSWIN\_PARTIAL\_SUPPORT, 876  
 OSCL\_HAS\_MSWIN\_SUPPORT, 876  
 OSCL\_HAS\_SYMBIAN\_SUPPORT, 876  
 OSCL\_HAS\_TLS\_SUPPORT, 876  
 OSCL\_HAS\_UNICODE\_SUPPORT, 876  
 OSCL\_HAS\_UNIX\_SUPPORT, 876  
 OSCL\_MEMFRAG\_PTR\_BEFORE\_LEN, 876  
 OSCL\_NATIVE\_INT64\_TYPE, 876  
 OSCL\_NATIVE\_UINT64\_TYPE, 876  
 OSCL\_NATIVE\_WCHAR\_TYPE, 876  
 OSCL\_TLS\_GET\_FUNC, 876  
 OSCL\_TLS\_IS\_KEYED, 876  
 OSCL\_TLS\_KEY\_CREATE\_FUNC, 876  
 OSCL\_TLS\_KEY\_DELETE\_FUNC, 876  
 OSCL\_TLS\_STORE\_FUNC, 876  
 TOsclBasicLockObject, 876  
 TOsclTlsKey, 876  
 UINT64, 876  
 UINT64\_HILO, 876  
 osclconfig\_util.h, 877  
     OSCL\_CLOCK\_HAS\_DRIFT\_-CORRECTION, 877  
     OSCL\_HAS\_SNPRINTF\_LONGLONG\_SUPPORT, 877  
     OSCL\_HAS\_SYMBIAN\_MATH, 877  
     OSCL\_HAS\_SYMBIAN\_TIMERS, 877  
     OSCL RAND\_MAX, 877  
     SLEEP\_ONE\_SEC, 877  
 osclconfig\_util\_check.h, 878  
 OsclConnect  
     osclconfig\_io.h, 847  
 OsclConnectComplete  
     osclconfig\_io.h, 847  
 OsclConnectMethod, 381  
     ~OsclConnectMethod, 381  
     Connect, 381  
     ConnectRequest, 381  
     NewL, 381  
 OsclConnectRequest, 383  
     Connect, 383  
     OsclConnectRequest, 383  
     OsclSocketI, 574  
 OsclDestructDealloc, 384  
     ~OsclDestructDealloc, 384  
     destruct\_and\_dealloc, 384  
 OsclDNS, 385  
     NewL, 385  
     osclio, 129  
 OsclDNSI, 387  
     ~OsclDNSI, 387  
     Close, 387  
     DNSRequestParam, 388  
     GetHostName, 387  
     GetHostNameResponseContainsAliasInfo, 388  
     GetHostNameSuccess, 388  
     GetNextHost, 388  
     GetNextHostSuccess, 388  
     NewL, 388  
     Open, 388  
     OsclDNSRequest, 388  
     OsclDNSRequestAO, 398  
     OsclGetHostNameRequest, 388  
     OsclSocketServI, 591  
 OsclDNSIBase, 389  
     ~OsclDNSIBase, 390  
     CancelFxn, 390  
     CancelGetHostName, 390  
     Close, 390  
     GetHostName, 390  
     GetHostNameResponseContainsAliasInfo, 390  
     GetHostNameSuccess, 390  
     GetNextHost, 390  
     GetNextHostSuccess, 390  
     iAlloc, 391  
     iSocketServ, 391  
     IsReady, 390  
     Open, 390  
     OsclDNSIBase, 390  
     OsclDNSRequest, 391  
     OsclGetHostNameRequest, 391  
 OsclDNSMethod, 392  
     Abort, 393  
     AbortAll, 393  
     CancelMethod, 393  
     ConstructL, 393  
     iAlloc, 394  
     iDNSFxn, 394  
     iDNSObserver, 394  
     iDNSRequestAO, 394  
     iId, 394  
     iLogger, 394  
     MethodDone, 393  
     OsclDNSMethod, 393  
     OsclDNSRequestAO, 398  
     Run, 393  
     StartMethod, 393  
 OsclDNSObserver, 395

HandleDNSEvent, 395  
OsclDNSRequest  
    OsclDNSI, 388  
    OsclDNSIBase, 391  
    OsclDNSRequestAO, 398  
OsclDNSRequestAO, 396  
    Abort, 397  
    Cancelled, 397  
    ConstructL, 397  
    DoCancel, 397  
    Failure, 397  
    GetHostNameParam, 398  
    GetSocketError, 397  
    iDNSI, 398  
    iDNSMethod, 398  
    iLogger, 398  
    iSocketError, 398  
    NewRequest, 397  
    OsclDNSI, 398  
    OsclDNSMethod, 398  
    OsclDNSRequest, 398  
    OsclDNSRequestAO, 397  
    osclio, 129  
    RequestDone, 397  
    Run, 397  
    Serv, 398  
    Success, 398  
OsclDoubleLink, 399  
    iNext, 399  
    InsertAfter, 399  
    InsertBefore, 399  
    iPrev, 399  
    OsclDoubleLink, 399  
    Remove, 399  
OsclDoubleList, 400  
    Head, 400  
    InsertHead, 400  
    InsertTail, 400  
    IsHead, 400  
    IsTail, 400  
    OsclDoubleList, 400  
    Tail, 400  
OsclDoubleListBase, 401  
    getHead, 401  
    getOffset, 401  
    iHead, 402  
    Insert, 402  
    InsertHead, 402  
    InsertTail, 402  
    iOffset, 402  
    IsEmpty, 402  
    OsclDoubleListBase, 401  
    Reset, 402  
    SetOffset, 402  
OsclDoubleRunner, 403  
    iHead, 404  
    iNext, 404  
    iOffset, 404  
    operator T \*, 403  
    operator++, 403  
    operator--, 403  
    OsclDoubleRunner, 403  
    Set, 403  
    SetToHead, 403  
    SetToTail, 404  
OsclErrAlreadyExists  
    osclerror, 99  
OsclErrAlreadyInstalled  
    osclerror, 99  
OsclErrArgument  
    osclerror, 99  
OsclErrBadHandle  
    osclerror, 99  
OsclErrBusy  
    osclerror, 100  
OsclErrCancelled  
    osclerror, 100  
OsclErrCorrupt  
    osclerror, 100  
OsclErrGeneral  
    osclerror, 100  
OsclErrInvalidState  
    osclerror, 100  
OsclErrNoHandler  
    osclerror, 100  
OsclErrNoMemory  
    osclerror, 100  
OsclErrNone  
    osclerror, 100  
OsclErrNoResources  
    osclerror, 100  
OsclErrNotInstalled  
    osclerror, 100  
OsclErrNotReady  
    osclerror, 100  
OsclErrNotSupported  
    osclerror, 100  
OsclError, 405  
    Leave, 405  
    LeaveIfError, 405  
    LeaveIfNull, 405  
    OsclErrorTrapImp, 411  
    OsclExecSchedulerCommonBase, 430  
    OsclTrapStack, 631  
    Pop, 405  
    PopDealloc, 406  
    PushL, 406  
osclerror

\_PV\_TRAP, 96  
\_PV\_TRAP\_NO\_TLS, 96, 97  
internalLeave, 97  
OSCL\_BAD\_ALLOC\_EXCEPTION\_CODE, 97  
OSCL\_CATCH, 97  
OSCL\_CATCH\_ANY, 97  
OSCL\_ERR\_NONE, 98  
OSCL\_FIRST\_CATCH, 98  
OSCL\_FIRST\_CATCH\_ANY, 98  
OSCL\_GetLastError, 101  
OSCL\_IsErrnoSupported, 102  
OSCL\_JUMP\_MAX\_JUMP\_MARKS, 98  
OSCL\_LAST\_CATCH, 98  
OSCL\_LEAVE, 98  
OSCL\_MAX\_TRAP\_LEVELS, 99  
OSCL\_SetLastError, 102  
OSCL\_StrError, 102  
OSCL\_TRAPSTACK\_POP, 99  
OSCL\_TRAPSTACK\_POPDEALLOC, 99  
OSCL\_TRAPSTACK\_PUSH, 99  
OSCL\_TRY, 99  
OSCL\_TRY\_NO\_TLS, 99  
OsclErrAlreadyExists, 99  
OsclErrAlreadyInstalled, 99  
OsclErrArgument, 99  
OsclErrBadHandle, 99  
OsclErrBusy, 100  
OsclErrCancelled, 100  
OsclErrCorrupt, 100  
OsclErrGeneral, 100  
OsclErrInvalidState, 100  
OsclErrNoHandler, 100  
OsclErrNoMemory, 100  
OsclErrNone, 100  
OsclErrNoResources, 100  
OsclErrNotInstalled, 100  
OsclErrNotReady, 100  
OsclErrNotSupported, 100  
OsclErrOverflow, 100  
OsclErrSystemCallFailed, 101  
OsclErrThreadContextIncorrect, 101  
OsclErrTimeout, 101  
OsclErrUnderflow, 101  
OsclFailure, 101  
OsclLeaveCode, 101  
OsclPending, 101  
OsclReturnCode, 101  
OsclSuccess, 101  
OsclTrapOperation, 101  
PVERROR\_DOLEAVE, 101  
PVERROR\_IMP\_JUMPS, 101  
PVERRORTRAP\_REGISTRY, 101  
PVERRORTRAP\_REGISTRY\_ID, 101  
OsclErrorAllocator, 407  
allocate, 407  
deallocate, 408  
operator delete, 408  
operator new, 408  
OsclErrorAllocator, 407  
OsclErrorTrap, 409  
Cleanup, 409  
GetErrorTrapImp, 409  
Init, 409  
OsclErrorTrapImp, 411  
OsclTrapStack, 631  
OsclErrorTrapImp, 410  
CPVInterfaceProxy, 411  
iJumpData, 411  
iLeave, 411  
iTrapStack, 411  
OsclError, 411  
OsclErrorTrap, 411  
OsclExecScheduler, 411  
OsclExecSchedulerCommonBase, 411  
OsclJump, 411, 452  
OsclJumpMark, 411  
OsclScheduler, 411  
OsclTrapStack, 411, 631  
Trap, 410  
TrapNoTls, 410  
UnTrap, 410  
OsclErrOverflow  
  osclerror, 100  
OsclErrSystemCallFailed  
  osclerror, 101  
OsclErrThreadContextIncorrect  
  osclerror, 101  
OsclErrTimeout  
  osclerror, 101  
OsclErrUnderflow  
  osclerror, 101  
OsclException, 412  
  getLeaveCode, 412  
  OsclException, 412  
OsclExclusiveArrayPtr, 413  
  ~OsclExclusiveArrayPtr, 414  
  \_Ptr, 415  
  get, 414  
  operator\*, 414  
  operator->, 414  
  operator=, 415  
  OsclExclusiveArrayPtr, 414  
  release, 415  
  set, 415  
OsclExclusivePtr, 416  
  ~OsclExclusivePtr, 417  
  \_Ptr, 418

get, 417  
 operator\*, 417  
 operator->, 417  
 operator=, 418  
 OsclExclusivePtr, 417  
 release, 418  
 set, 418  
**OsclExclusivePtrA**, 419  
 ~OsclExclusivePtrA, 420  
 \_Ptr, 421  
 get, 420  
 operator\*, 420  
 operator->, 420  
 operator=, 421  
 OsclExclusivePtrA, 420  
 release, 421  
 set, 421  
**OsclExecScheduler**, 423  
 Current, 423  
 OsclErrorTrapImp, 411  
 OsclExecSchedulerBase, 425  
 OsclExecSchedulerCommonBase, 430  
 OsclScheduler, 424  
 PVActiveBase, 644  
 PVThreadContext, 664  
 RegisterForCallback, 423  
 RunSchedulerNonBlocking, 423  
**OsclExecSchedulerBase**, 425  
 OsclCoeActiveScheduler, 425  
 OsclExecScheduler, 425  
 PVActiveBase, 425  
 PVThreadContext, 664  
 OsclExecSchedulerCommonBase, 426  
 ~OsclExecSchedulerCommonBase, 428  
 AddToExecTimerQ, 428  
 BeginScheduling, 428  
 BlockingLoopL, 428  
 CallRunExec, 428  
 CleanupExecQ, 428  
 ConstructL, 428  
 EndScheduling, 428  
 Error, 428  
 FindPVBBase, 428  
 GetId, 428  
 GetName, 428  
 GetScheduler, 428  
 iAlloc, 432  
 iBlockingMode, 432  
 iDebugLogger, 432  
 iDefAlloc, 432  
 iDoStop, 432  
 iDoSuspend, 432  
 iErrorTrapImp, 432  
 iExecTimerQ, 432  
 iLogger, 432  
 iLogPerfIndentStr, 432  
 iLogPerfIndentStrLen, 432  
 iLogPerfTotal, 432  
 iName, 432  
 iNativeMode, 432  
 IncLogPerf, 429  
 InitExecQ, 429  
 InstallScheduler, 429  
 iNumAOAdded, 432  
 iReadyQ, 432  
 iResumeSem, 432  
 IsInstalled, 429  
 IsStarted, 429  
 iStopper, 432  
 iStopperCrit, 432  
 iSuspended, 432  
 iThreadContext, 432  
 iTimeCompareThreshold, 432  
 OsclActiveObject, 430  
 OsclCoeActiveScheduler, 430  
 OsclError, 430  
 OsclErrorTrapImp, 411  
 OsclExecScheduler, 430  
 OsclExecSchedulerCommonBase, 428  
 OsclReadyQ, 430  
 OsclScheduler, 430  
 OsclTimerCompare, 430  
 OsclTimerObject, 432  
 PendComplete, 429  
 PVActiveBase, 432  
 PVSchedulerStopper, 432  
 PVThreadContext, 432, 664  
 RequestCanceled, 429  
 ResetLogPerf, 429  
 ResumeScheduler, 429  
 SetScheduler, 429  
 StartNativeScheduler, 429  
 StartScheduler, 429  
 StopScheduler, 429  
 SuspendScheduler, 429  
 UninstallScheduler, 430  
 UpdateTimers, 430  
 UpdateTimersMsec, 430  
 WaitForReadyAO, 430  
**OsclExtractFilenameFromFullPath**  
 osclio, 121  
**OsclFailure**  
 oscrror, 101  
**OsclFileCache**, 434  
 ~OsclFileCache, 435  
 \_fixedCaches, 435  
 \_movableCache, 435  
 AddFixedCache, 435

Close, 435  
EndOfFile, 435  
FileSize, 435  
Flush, 435  
Open, 435  
Oscl\_File, 225  
OsclFileCache, 435  
OsclFileCacheBuffer, 435  
Read, 435  
Seek, 435  
Tell, 435  
Write, 435  
OsclFileCacheBuffer, 436  
capacity, 437  
Contains, 436  
currentPos, 437  
endPos, 437  
filePosition, 437  
FillFromFile, 436  
iContainer, 437  
isFixed, 437  
IsUpdated, 436  
Oscl\_File, 225  
OsclFileCache, 435  
OsclFileCacheBuffer, 436  
pBuffer, 437  
Preceeds, 436  
PrepRead, 436  
PrepWrite, 437  
SetPosition, 437  
updateEnd, 437  
updateStart, 437  
usableSize, 437  
WriteUpdatesToFile, 437  
OsclHandle, 438  
Handle, 438  
Oscl\_File, 438  
OsclHandle, 438  
OsclFileManager, 439  
OSCL\_FILE\_ATTRIBUTE\_ARCHIVE, 439  
OSCL\_FILE\_ATTRIBUTE\_DIRECTORY,  
    439  
OSCL\_FILE\_ATTRIBUTE\_HIDDEN, 439  
OSCL\_FILE\_ATTRIBUTE\_NORMAL, 440  
OSCL\_FILE\_ATTRIBUTE\_READONLY,  
    439  
OSCL\_FILE\_ATTRIBUTE\_SYSTEM, 439  
OSCL\_FILE\_ATTRIBUTE\_TYPE, 439  
OsclFileStats, 441  
End, 441  
Log, 441  
LogAll, 441  
OsclFileStats, 441  
Start, 441  
OsclFileStatsItem, 442  
iOpCount, 442  
iParam, 442  
iParam2, 442  
iStartTick, 442  
iTTotalTicks, 442  
OsclFloat  
    osclbase, 37  
OsclGetAsyncSockErr  
    osclconfig\_io.h, 847  
OsclGetDottedAddr  
    osclconfig\_io.h, 847  
OsclGetDottedAddrVector  
    osclconfig\_io.h, 848  
OsclGetFileAttributes  
    osclio, 121  
OsclGetFileCreationTime  
    osclio, 122  
OsclGetFileLastAccessTime  
    osclio, 122, 123  
OsclGetFileLastWriteTime  
    osclio, 123  
OsclGetFileSize  
    osclio, 124  
OsclGethostbyname  
    osclconfig\_io.h, 848  
OsclGetHostByNameMethod, 444  
    ~OsclGetHostByNameMethod, 444  
    GetHostName, 444  
    NewL, 444  
    OsclGetHostByNameRequest, 445  
OsclGetHostByNameRequest, 445  
    OsclDNSI, 388  
    OsclDNSIBase, 391  
    OsclGetHostByNameMethod, 445  
OsclGetPeerName  
    osclconfig\_io.h, 848  
OsclInit, 446  
    Cleanup, 446  
    Init, 446  
OsclInteger64Transport, 447  
    iHigh, 447  
    iLow, 447  
osclio  
    ~OsclDNS, 128  
    ~OsclDNSObserver, 128  
    ~OsclSocketServ, 129  
    ~OsclTCPSocket, 129  
    ~OsclUDPSocket, 129  
    Accept, 110  
    Bind, 110  
    BindAsync, 110, 111  
    CancelAccept, 111  
    CancelBind, 111

CancelConnect, 112  
 CancelGetHostName, 112  
 CancelListen, 112  
 CancelRecv, 112  
 CancelRecvFrom, 112  
 CancelSend, 112  
 CancelSendTo, 112  
 CancelShutdown, 113  
 Close, 113  
 Connect, 113, 114  
 EOscIFileOp\_Close, 109  
 EOscIFileOp\_EndOfFile, 109  
 EOscIFileOp\_Flush, 109  
 EOscIFileOp\_Last, 109  
 EOscIFileOp\_NativeClose, 109  
 EOscIFileOp\_NativeEndOfFile, 109  
 EOscIFileOp\_NativeFlush, 109  
 EOscIFileOp\_NativeOpen, 109  
 EOscIFileOp\_NativeRead, 109  
 EOscIFileOp\_NativeSeek, 109  
 EOscIFileOp\_NativeSetSize, 109  
 EOscIFileOp\_NativeSize, 109  
 EOscIFileOp\_NativeTell, 109  
 EOscIFileOp\_NativeWrite, 109  
 EOscIFileOp\_Open, 109  
 EOscIFileOp\_Read, 109  
 EOscIFileOp\_Seek, 109  
 EOscIFileOp\_SetSize, 109  
 EOscIFileOp\_Size, 109  
 EOscIFileOp\_Tell, 109  
 EOscIFileOp\_Write, 109  
 EPVDNSCancel, 109  
 EPVDNSFailure, 109  
 EPVDNSGetHostName, 110  
 EPVDNSPending, 109  
 EPVDNSSuccess, 109  
 EPVDNSTimeout, 109  
 GetAcceptedSocketL, 114  
 GetHostName, 114  
 GetPeerName, 115  
 GetRecvData, 115  
 GetSendData, 116  
 Join, 116  
 JoinMulticastGroup, 116  
 Listen, 117  
 ListenAsync, 117  
 OSCL\_FILEMGMT\_E\_ALREADY\_EXISTS, 108  
 OSCL\_FILEMGMT\_E\_NO\_MATCH, 108  
 OSCL\_FILEMGMT\_E\_NOT\_EMPTY, 108  
 OSCL\_FILEMGMT\_E\_NOT\_IMPLEMENTED, 108  
 OSCL\_FILEMGMT\_E\_OK, 108  
 OSCL\_FILEMGMT\_E\_PATH\_NOT\_FOUND, 108  
 OSCL\_FILEMGMT\_E\_PATH\_TOO\_LONG, 108  
 OSCL\_FILEMGMT\_E\_PERMISSION\_DENIED, 108  
 OSCL\_FILEMGMT\_E\_SYS\_SPECIFIC, 108  
 OSCL\_FILEMGMT\_E\_UNKNOWN, 108  
 OSCL\_FILEMGMT\_MODE\_DIR, 108  
 OSCL\_FILEMGMT\_PERMS\_EXECUTE, 109  
 OSCL\_FILEMGMT\_PERMS\_READ, 109  
 OSCL\_FILEMGMT\_PERMS\_WRITE, 109  
 oscl\_chdir, 117  
 OSCL\_FILE\_CHAR\_PATH\_DELIMITER, 108  
 OSCL\_FILE\_STATS\_LOGGER\_NODE, 108  
 OSCL\_FILE\_WCHAR\_PATH\_DELIMITER, 108  
 OSCL\_FILEMGMT\_ERR\_TYPE, 108  
 OSCL\_FILEMGMT\_MODES, 108  
 OSCL\_FILEMGMT\_PERMS, 108  
 OSCL\_FSSTAT, 108  
 oscl\_getcwd, 118  
 OSCL\_IO\_EXTENSION\_MAXLEN, 108  
 OSCL\_IO\_FILENAME\_MAXLEN, 108  
 oscl\_mkdir, 118  
 oscl\_rename, 119  
 oscl\_rmdir, 119  
 oscl\_stat, 120  
 OSCL\_STAT\_BUF, 108  
 oscl\_statfs, 120  
 OsclDNS, 129  
 OsclDNSRequestAO, 129  
 OsclExtractFilenameFromFullPath, 121  
 OsclGetFileAttributes, 121  
 OsclGetFileCreationTime, 122  
 OsclGetFileLastAccessTime, 122, 123  
 OsclGetFileLastWriteTime, 123  
 OsclGetFileSize, 124  
 OsclTCPSocket, 129  
 OsclUDPSocket, 129  
 Recv, 124  
 RecvFrom, 125  
 Send, 125  
 SendTo, 125  
 SetMulticastTTL, 126  
 SetOptionToReuseAddress, 126  
 SetRecvBufferSize, 127  
 SetTOS, 127  
 Shutdown, 127  
 ThreadLogoff, 128  
 ThreadLogon, 128  
 TOscIFileHandle, 108

TOsclFileOffsetInt32, 108  
TOsclFileOp, 109  
TPVDNSEvent, 109  
TPVDNSFxN, 109  
OsclIpMReq, 448  
    interfaceAddr, 448  
    multicastAddr, 448  
    OsclIpMReq, 448  
OsclIPSocketI, 449  
    ~OsclIPSocketI, 450  
    Alloc, 450  
    Bind, 450  
    Close, 450  
    ConstructL, 450  
    GetPeerName, 450  
    GetRecvData, 450  
    GetSendData, 450  
    iAddress, 451  
    iAlloc, 451  
    iId, 451  
    iLogger, 451  
    iObserver, 451  
    iSocket, 451  
    iSocketServ, 451  
    Join, 450  
    OsclIPSocketI, 450  
    OsclSocketMethod, 451  
    OsclSocketRequestAO, 451  
    SetOptionToReuseAddress, 450  
    SetRecvBufferSize, 450  
    SetTOS, 450  
    SocketServ, 450  
    ThreadLogoff, 450  
    ThreadLogon, 450  
OsclJoin  
    osclconfig\_io.h, 848  
OsclJump, 452  
    ~OsclJump, 452  
    Jump, 452  
    OsclErrorTrapImp, 411, 452  
    StaticJump, 452  
    Top, 452  
OsclJumpMark  
    OsclErrorTrapImp, 411  
OsclLeaveCode  
    osclerror, 101  
OsclListen  
    osclconfig\_io.h, 849  
OsclListenMethod, 453  
    ~OsclListenMethod, 453  
    Listen, 453  
    ListenRequest, 453  
    NewL, 453  
OsclListenRequest, 454  
    Listen, 454  
    OsclListenRequest, 454  
    OsclLockBase, 455  
        ~OsclLockBase, 455  
        Lock, 455  
        Unlock, 455  
    OsclMakeInAddr  
        osclconfig\_io.h, 849  
    OsclMakeSockAddr  
        osclconfig\_io.h, 849  
    OsclMem, 456  
        Cleanup, 456  
        Init, 456  
        OsclMemGlobalAuditObject, 468  
    OsclMemAllocator, 457  
        allocate, 457  
        deallocate, 457  
    OsclMemAllocDestructDealloc, 458  
        allocate, 458  
        deallocate, 458  
        destruct\_and\_dealloc, 458  
    OsclMemAudit, 460  
    OSCLMemAutoPtr, 461  
        ~OSCLMemAutoPtr, 462  
        \_Ownership, 464  
        allocate, 462  
        deallocate, 462  
        get, 463  
        operator\*, 463  
        operator->, 463  
        operator=, 463  
        OSCLMemAutoPtr, 462  
        release, 463  
        setWithoutOwnership, 464  
        takeOwnership, 464  
    OsclMemBasicAllocator, 465  
        allocate, 465  
        deallocate, 465  
    OsclMemBasicAllocDestructDealloc, 466  
        allocate, 466  
        deallocate, 466  
        destruct\_and\_dealloc, 466  
    OsclMemGlobalAuditObject, 468  
        audit\_type, 468  
        getGlobalMemAuditObject, 468  
        OsclMem, 468  
    OsclMemInit  
        osclmemory, 67  
    osclmemory  
        \_OSCL\_CLEANUP\_BASE\_CLASS, 55  
        \_OSCL\_TRAP\_NEW, 55  
        \_oscl\_calloc, 64  
        \_oscl\_default\_new, 64  
        \_oscl\_free, 64

\_oscl\_malloc, 64  
 \_oscl\_realloc, 64  
 ALLOC\_NODE\_FLAG, 67  
 COMPUTE\_MEM\_ALIGN\_SIZE, 55  
 DEFAULT\_MM\_AUDIT\_MODE, 56  
 DEFAULT\_POSTFILL\_PATTERN, 56  
 DEFAULT\_PREFILL\_PATTERN, 56  
 FENCE\_PATTERN, 56  
 MEM\_ALIGN\_SIZE, 56  
 MIN\_FENCE\_SIZE, 56  
 MM\_ALLOC\_MAX\_QUERY\_FILENAME\_LEN, 56  
 MM\_ALLOC\_MAX\_QUERY\_TAG\_LEN, 56  
 MM\_AllocNodeAutoPtr, 64  
 MM\_AUDIT\_ALLOC\_NODE\_ENABLE\_FLAG, 56  
 MM\_AUDIT\_ALLOC\_NODE\_SUPPORT, 56  
 MM\_AUDIT\_FAILURE\_SIMULATION\_SUPPORT, 56  
 MM\_AUDIT\_FENCE\_SUPPORT, 56  
 MM\_AUDIT\_FILL\_SUPPORT, 56  
 MM\_AUDIT\_INCLUDE\_ALL\_HEAP\_VALIDATION, 56  
 MM\_AUDIT\_POSTFILL\_FLAG, 56  
 MM\_AUDIT\_PREFILL\_FLAG, 56  
 MM\_AUDIT\_SUPPRESS\_FILENAME\_FLAG, 56  
 MM\_AUDIT\_VALIDATE\_ALL\_HEAP\_FLAG, 56  
 MM\_AUDIT\_VALIDATE\_BLOCK, 56  
 MM\_AUDIT\_VALIDATE\_ON\_FREE\_FLAG, 56  
 MM\_StatsNodeTagTreeType, 64  
 MMAuditCharAutoPtr, 64  
 MMAuditUint8AutoPtr, 64  
 operator delete, 64  
 operator new, 65  
 OSCL\_ALLOC\_DELETE, 56  
 OSCL\_ALLOC\_NEW, 57  
 OSCL\_ARRAY\_DELETE, 57  
 OSCL\_ARRAY\_NEW, 57  
 OSCL\_AUDIT\_ARRAY\_NEW, 58  
 OSCL\_AUDIT\_CALLOC, 58  
 OSCL\_AUDIT\_MALLOC, 58  
 OSCL\_AUDIT\_NEW, 59  
 OSCL\_AUDIT\_REALLOC, 59  
 OSCL\_CALLOC, 60  
 oscl\_calloc, 59  
 OSCL\_CLEANUP\_BASE\_CLASS, 60  
 OSCL\_DEFAULT\_FREE, 60  
 OSCL\_DEFAULT\_MALLOC, 60  
 OSCL\_DELETE, 60  
 OSCL\_DISABLE\_WARNING\_RETURN\_TYPE\_NOT\_UDT, 61  
 OSCL\_DISABLE\_WARNING\_TRUNCATE\_DEBUG\_MESSAGE, 61  
 OSCL\_FREE, 61  
 oscl\_free, 61  
 OSCL\_MALLOC, 61  
 oscl\_malloc, 61  
 oscl\_mem\_aligned\_size, 65  
 oscl\_memcmp, 65  
 oscl\_memcpy, 65  
 oscl\_memmove, 66  
 oscl\_memmove32, 66  
 oscl\_memset, 66  
 OSCL\_NEW, 61  
 OSCL\_PLACEMENT\_NEW, 62  
 OSCL\_REALLOC, 62  
 oscl\_realloc, 62  
 OSCL\_TRAP\_ALLOC\_NEW, 62  
 OSCL\_TRAP\_AUDIT\_NEW, 63  
 OSCL\_TRAP\_NEW, 63  
 OsclMemInit, 67  
 OsclMemStatsNodeAutoPtr, 64  
 OsclTagTreeType, 64  
 TagTreeAllocator, 64  
 OsclMemoryFragment, 469  
 len, 469  
 ptr, 469  
 OsclMemPoolFixedChunkAllocator, 470  
 ~OsclMemPoolFixedChunkAllocator, 471  
 addRef, 471  
 allocate, 471  
 CancelFreeChunkAvailableCallback, 471  
 createmempool, 472  
 deallocate, 472  
 destroymempool, 472  
 enablenullpointerreturn, 472  
 iCheckNextAvailableFreeChunk, 473  
 iChunkAlignment, 473  
 iChunkSize, 473  
 iChunkSizeMemAligned, 473  
 iEnableNullPtrReturn, 473  
 iFreeMemChunkList, 473  
 iMemPool, 473  
 iMemPoolAligned, 473  
 iMemPoolAllocator, 473  
 iNextAvailableContextData, 473  
 iNumChunk, 473  
 iObserver, 473  
 iRefCount, 473  
 notifyfreechunkavailable, 472  
 OsclMemPoolFixedChunkAllocator, 471  
 removeRef, 472  
 OsclMemPoolFixedChunkAllocatorObserver, 474

~OsclMemPoolFixedChunkAllocatorObserver, 474  
freechunkavailable, 474  
**OsclMemPoolResizableAllocator**, 475  
  ~OsclMemPoolResizableAllocator, 476  
  addnewmempoolbuffer, 477  
  addRef, 477  
  allocate, 477  
  allocateblock, 477  
  CancelFreeChunkAvailableCallback, 477  
  CancelFreeMemoryAvailableCallback, 477  
  deallocate, 477  
  deallocateblock, 478  
  destroyallmempoolbuffers, 478  
  enablenullpointerreturn, 478  
  findfreeblock, 478  
  getAllocatedSize, 478  
  getAvailableSize, 478  
  getBufferSize, 478  
  getLargestContiguousFreeBlockSize, 478  
  getMemPoolBufferAllocatedSize, 478  
  getMemPoolBufferSize, 479  
  iBlockInfoAlignedSize, 482  
  iBufferInfoAlignedSize, 482  
  iCheckFreeMemoryAvailable, 482  
  iCheckNextAvailable, 482  
  iDebugLogger, 482  
  iEnableNullPtrReturn, 482  
  iExpectedNumBlocksPerBuffer, 482  
  iFreeMemContextData, 482  
  iFreeMemPoolObserver, 482  
  iMaxNewMemPoolBufferSz, 482  
  iMemPoolBufferAllocator, 482  
  iMemPoolBufferList, 482  
  iMemPoolBufferNumLimit, 482  
  iMemPoolBufferSize, 482  
  iMemPoolPrevAllocBufferIndex, 482  
  iNextAvailableContextData, 482  
  iObserver, 482  
  iRefCount, 482  
  iRequestedAvailableFreeMemSize, 482  
  iRequestedNextAvailableSize, 482  
  memoryPoolBufferMgmtOverhead, 479  
  notifyfreeblockavailable, 479  
  notifyfreememoryavailable, 479  
**OsclMemPoolResizableAllocator**, 476  
  removeRef, 479  
  setMaxSzForNewMemPoolBuffer, 479  
  trim, 479  
  validateblock, 480  
**OsclMemPoolResizableAllocator::MemPoolBlockInfo**, 184  
  iBlockBuffer, 184  
  iBlockPostFence, 184  
  iBlockPreFence, 184  
  iBlockSize, 184  
  iNextFreeBlock, 184  
  iParentBuffer, 184  
  iPrevFreeBlock, 184  
**OsclMemPoolResizableAllocator::MemPoolBufferInfo**, 185  
  iAllocatedSz, 185  
  iBufferPostFence, 185  
  iBufferPreFence, 185  
  iBufferSize, 185  
  iEndAddr, 185  
  iNextFreeBlock, 185  
  iNumOutstanding, 185  
  iPrevAllocBlock, 185  
  iStartAddr, 185  
**OsclMemPoolResizableAllocatorMemoryObserver**, 484  
  ~OsclMemPoolResizableAllocatorMemoryObserver, 484  
    freememoryavailable, 484  
**OsclMemPoolResizableAllocatorObserver**, 485  
  ~OsclMemPoolResizableAllocatorObserver, 485  
    freeblockavailable, 485  
**OsclMemStatsNode**, 486  
  ~OsclMemStatsNode, 486  
  operator delete, 486  
  operator new, 486  
  OsclMemStatsNode, 486  
  pMMFIParam, 487  
  pMMStats, 487  
  reset, 486  
  tag, 487  
**OsclMemStatsNodeAutoPtr**  
  osclmemory, 64  
**OsclMutex**, 488  
  ~OsclMutex, 488  
  Close, 488  
  Create, 488  
  Lock, 489  
  OsclMutex, 488  
  TryLock, 489  
  Unlock, 489  
**OsclNameString**, 490  
  MaxLen, 490  
  OsclNameString, 490  
  Set, 490  
  Str, 490  
**OsclNativeFile**, 492  
  ~OsclNativeFile, 493  
  Close, 493  
  EndOfFile, 493  
  Flush, 493

GetError, 493  
 GetReadAsyncNumElements, 493  
 HasAsyncRead, 493  
 Mode, 493  
 Open, 493  
 Oscl\_FileServer, 231  
 OsclNativeFile, 493  
 Read, 493  
 ReadAsync, 493  
 ReadAsyncCancel, 494  
 Seek, 494  
 SetSize, 494  
 Size, 494  
 Tell, 494  
 Write, 494  
 OsclNativeFileParams, 495  
 iAsyncReadBufferSize, 495  
 iNativeAccessMode, 495  
 iNativeBufferSize, 495  
 OsclNativeFileParams, 495  
 OsclNetworkAddress, 496  
     ipAddr, 496  
     operator==, 496  
     OsclNetworkAddress, 496  
     port, 496  
 OsclNoYieldMutex  
     oscl\_mutex.h, 756  
 OsclNullLock, 497  
     ~OsclNullLock, 497  
     Lock, 497  
     Unlock, 497  
 OsclPending  
     osclerror, 101  
 OsclPipe  
     osclconfig\_io.h, 849  
 OsclPriorityLink, 498  
     iPriority, 498  
 OsclPriorityList, 499  
     Head, 499  
     Insert, 499  
     IsHead, 499  
     IsTail, 499  
     OsclPriorityList, 499  
     Tail, 499  
 OsclPriorityQueue, 500  
     ~OsclPriorityQueue, 501  
     c, 503  
     comp, 503  
     compare\_EQ, 501  
     compare\_LT, 501  
     const\_reference, 501  
     container\_type, 501  
     empty, 502  
     find\_heap, 502  
     iterator, 501  
     oscl\_priqueue\_test, 503  
     OsclPriorityQueue, 501  
     pop, 502  
     pop\_heap, 502  
     push, 502  
     push\_heap, 502  
     remove, 502  
     reserve, 502  
     size, 503  
     swap, 503  
     top, 503  
     validate, 503  
     value\_type, 501  
     vec, 503  
 OsclPriorityQueueBase, 505  
     ~OsclPriorityQueueBase, 505  
     construct, 505  
     find\_heap, 505  
     Oscl\_Vector\_Base, 324  
     pop\_heap, 505  
     push\_heap, 505  
     remove, 505  
 osclproc  
     EPVThreadContext\_InThread, 133  
     EPVThreadContext\_NonOsclThread, 133  
     EPVThreadContext\_OsclThread, 133  
     EPVThreadContext\_Undetermined, 133  
     OSCL\_PERF\_SUMMARY\_LOGGING, 132  
     OSCL\_REQUEST\_ERR\_CANCEL, 133  
     OSCL\_REQUEST\_ERR\_GENERAL, 133  
     OSCL\_REQUEST\_ERR\_NONE, 133  
     OSCL\_REQUEST\_PENDING, 133  
     OSCL\_ZEROIZE, 132  
     OsclPtrAdd, 133  
     OsclPtrSub, 133  
     PV\_SCHED\_CHECK\_Q, 132  
     PV\_SCHED\_ENABLE\_LOOP\_STATS, 132  
     PV\_SCHED\_ENABLE\_PERF\_LOGGING,  
         132  
     PV\_SCHED\_ENABLE\_THREAD\_-  
         CONTEXT\_CHECKS, 132  
     PV\_SCHED\_FAIR\_SCHEDULING, 132  
     PV\_SCHED\_LOG\_Q, 132  
     PVEEXECNAMELEN, 132  
     PVSCHEDNAMELEN, 132  
     QUE\_ITER\_BEGIN, 132  
     QUE\_ITER\_END, 132  
     TOsclReady, 133  
     TPVThreadContext, 133  
 OsclProcStatus, 507  
     ALREADY\_SUSPENDED\_ERROR, 508  
     BAD\_THREADID\_ADDR\_ERROR, 507  
     eOsclProcError, 507

EXCEED\_MAX\_COUNT\_VARIABLE\_-  
  ERROR, 508  
EXCEED\_MAX\_SEM\_COUNT\_ERROR,  
  508  
INVALID\_ACCESS\_ERROR, 508  
INVALID\_ARGUMENT\_ERROR, 508  
INVALID\_FUNCTION\_ERROR, 508  
INVALID\_HANDLE\_ERROR, 508  
INVALID\_OPERATION\_ERROR, 508  
INVALID\_PARAM\_ERROR, 508  
INVALID\_POINTER\_ERROR, 508  
INVALID\_PRIORITY\_ERROR, 508  
INVALID\_THREAD\_ERROR, 508  
INVALID\_THREAD\_ID\_ERROR, 507  
MAX\_THRDS\_REACHED\_ERROR, 507  
MUTEX\_LOCKED\_ERROR, 508  
NO\_PERMISSION\_ERROR, 508  
NOT\_ENOUGH\_MEMORY\_ERROR, 507  
NOT\_ENOUGH\_RESOURCES\_ERROR,  
  507  
NOT\_IMPLEMENTED, 508  
NOT\_SUSPENDED\_ERROR, 508  
OTHER\_ERROR, 507  
OUTOFMEMORY\_ERROR, 507  
PSHARED\_ATTRIBUTE\_SETTING\_-  
  ERROR, 508  
PSHARED\_NOT\_ZERO\_ERROR, 508  
RELOCK\_MUTEX\_ERROR, 508  
SEM\_NOT\_SIGNALLED\_ERROR, 508  
SUCCESS\_ERROR, 507  
SYSTEM\_RESOURCES\_UNAVAILABLE\_-  
  ERROR, 508  
THREAD\_1\_INACTIVE\_ERROR, 508  
THREAD\_BLOCK\_ERROR, 508  
THREAD\_NOT\_OWN\_MUTEX\_ERROR,  
  508  
TOO\_MANY\_THREADS\_ERROR, 507  
WAIT\_ABANDONED\_ERROR, 508  
WAIT\_TIMEOUT\_ERROR, 508  
  
OsclPtr, 509  
  Append, 509  
  Length, 509  
  OsclPtr, 509  
  Ptr, 509  
  Set, 509  
  SetLength, 509  
  Zero, 509  
  
OsclPtrAdd  
  osclproc, 133  
  
OsclPtrC, 511  
  Left, 511  
  Length, 511  
  OsclPtrC, 511  
  Ptr, 511  
  
Right, 511  
Set, 512  
SetLength, 512  
Zero, 512  
  
OsclPtrSub  
  osclproc, 133  
  
OsclRand, 513  
  Rand, 513  
  Seed, 513  
  
OsclReadFD  
  osclconfig\_io.h, 849  
  
OsclReadyAlloc, 514  
  allocate, 514  
  allocate\_fl, 514  
  deallocate, 514  
  
OsclReadyCompare, 515  
  compare, 515  
  PVActiveBase, 644  
  
OsclReadyQ, 516  
  Callback, 516  
  Construct, 516  
  Depth, 516  
  IsIn, 516  
  OsclExecSchedulerCommonBase, 430  
  PendComplete, 517  
  PopTop, 517  
  PVActiveBase, 644  
  RegisterForCallback, 517  
  Remove, 517  
  ThreadLogoff, 517  
  ThreadLogon, 517  
  TimerCallback, 517  
  Top, 517  
  WaitAndPopTop, 517  
  WaitForRequestComplete, 517  
  
OsclReadySetPosition  
  PVActiveBase, 644  
  
OsclRecv  
  osclconfig\_io.h, 849  
  
OsclRecvFrom  
  osclconfig\_io.h, 849  
  
OsclRecvFromMethod, 518  
  ~OsclRecvFromMethod, 518  
  GetRecvData, 518  
  NewL, 518  
  RecvFrom, 519  
  RecvFromRequest, 519  
  
OsclRecvFromRequest, 520  
  GetRecvData, 520  
  OsclRecvFromRequest, 520  
  OsclSocketI, 574  
  RecvFrom, 520  
  Success, 520  
  
OsclRecvMethod, 522

~OsclRecvMethod, 522  
 GetRecvData, 522  
 NewL, 522  
 Recv, 522  
 RecvRequest, 522  
 OsclRecvRequest, 524  
     GetRecvData, 524  
     OsclRecvRequest, 524  
     OsclSocketI, 574  
     Recv, 524  
     Success, 524  
 OsclRefCounter, 525  
     ~OsclRefCounter, 525  
     addRef, 525  
     getCount, 525  
     removeRef, 525  
 OsclRefCounterDA, 527  
     ~OsclRefCounterDA, 527  
     addRef, 528  
     getCount, 528  
     OsclRefCounterDA, 527  
     removeRef, 528  
 OsclRefCounterMemFrag, 529  
     ~OsclRefCounterMemFrag, 529  
     getCapacity, 530  
     getCount, 530  
     getMemFrag, 530  
     getMemFragPtr, 530  
     getMemFragSize, 530  
     getRefCounter, 530  
     operator=, 530  
     OsclRefCounterMemFrag, 529  
 OsclRefCounterMTDA, 531  
     ~OsclRefCounterMTDA, 531  
     addRef, 532  
     getCount, 532  
     OsclRefCounterMTDA, 531  
     removeRef, 532  
 OsclRefCounterMTSA, 533  
     ~OsclRefCounterMTSA, 533  
     addRef, 534  
     getCount, 534  
     OsclRefCounterMTSA, 533  
     removeRef, 534  
 OsclRefCounterSA, 535  
     ~OsclRefCounterSA, 535  
     addRef, 536  
     getCount, 536  
     OsclRefCounterSA, 535  
     removeRef, 536  
 OsclRegistryAccessClient, 537  
     ~OsclRegistryAccessClient, 537  
     Close, 537  
     Connect, 537  
     GetFactories, 537  
     GetFactory, 537  
     OsclRegistryAccessClient, 537  
     OsclRegistryClientImpl, 545  
     OsclRegistryServTlsImpl, 548  
     OsclRegistryAccessClientImpl, 539  
     OsclRegistryAccessClientTlsImpl, 540  
     OsclRegistryAccessElement, 541  
         iFactory, 541  
         iMimeType, 541  
     OsclRegistryClient, 542  
         ~OsclRegistryClient, 542  
         Close, 542  
         Connect, 542  
         OsclRegistryClient, 542  
         OsclRegistryClientImpl, 545  
         OsclRegistryServTlsImpl, 548  
         Register, 542  
         UnRegister, 543  
     OsclRegistryClientImpl, 544  
         Close, 544  
         Connect, 544  
         GetFactories, 544  
         GetFactory, 544  
         OsclRegistryAccessClient, 545  
         OsclRegistryClient, 545  
         Register, 544  
         UnRegister, 544  
     OsclRegistryClientTlsImpl, 546  
     OsclRegistryServTlsImpl, 547  
         ~OsclRegistryServTlsImpl, 548  
         Close, 548  
         Connect, 548  
         GetFactories, 548  
         GetFactory, 548  
         OsclRegistryAccessClient, 548  
         OsclRegistryClient, 548  
         OsclRegistryServTlsImpl, 548  
         Register, 548  
         UnRegister, 548  
 OsclReturnCode  
     osclerror, 101  
 OsclScheduler, 549  
     Cleanup, 549  
     Init, 549  
     OsclErrorTrapImpl, 411  
     OsclExecScheduler, 424  
     OsclExecSchedulerCommonBase, 430  
 OsclSchedulerCommonBase  
     PVActiveBase, 644  
 OsclSchedulerObserver, 550  
     ~OsclSchedulerObserver, 550  
     OsclSchedulerReadyCallback, 550  
     OsclSchedulerTimerCallback, 550

OsclSchedulerReadyCallback  
    OsclSchedulerObserver, 550

OsclSchedulerTimerCallback  
    OsclSchedulerObserver, 550

OsclScopedLock, 551  
    ~OsclScopedLock, 551  
    OsclScopedLock, 551

OsclSelect, 552  
    iErrAlloc, 553  
    iHeapCheck, 553  
    iOsclBase, 553  
    iOsclErrorTrap, 553  
    iOsclLogger, 553  
    iOsclMemory, 553  
    iOsclScheduler, 553  
    iOutputFile, 553  
    iSchedulerAlloc, 553  
    iSchedulerName, 553  
    iSchedulerReserve, 553  
    OsclSelect, 553

OsclSemaphore, 554  
    ~OsclSemaphore, 554  
    Close, 554  
    Create, 554  
    OsclSemaphore, 554  
    Signal, 555  
    TryWait, 555  
    Wait, 555

OsclSend  
    osclconfig\_io.h, 849

OsclSendMethod, 557  
    ~OsclSendMethod, 557  
    GetSendData, 557  
    NewL, 557  
    Send, 557  
    SendRequest, 557

OsclSendRequest, 559  
    GetSendData, 559  
    OsclSendRequest, 559  
    OsclSocketI, 574  
    Send, 559  
    Success, 559

OsclSendTo  
    osclconfig\_io.h, 850

OsclSendToMethod, 560  
    ~OsclSendToMethod, 560  
    GetSendData, 560  
    NewL, 560  
    SendTo, 560  
    SendToRequest, 560

OsclSendToRequest, 562  
    GetSendData, 562  
    OsclSendToRequest, 562  
    OsclSocketI, 574

SendTo, 562  
Success, 562

OsclSetNonBlocking  
    osclconfig\_io.h, 850

OsclSetRecvBufferSize  
    osclconfig\_io.h, 850

OsclSetSockOpt  
    osclconfig\_io.h, 850

OsclSharedPtr, 563  
    osclbase, 49  
    OsclSharedPtr, 564

OsclShutdown  
    osclconfig\_io.h, 850

OsclShutdownMethod, 565  
    ~OsclShutdownMethod, 565  
    NewL, 565  
    Shutdown, 565  
    ShutdownRequest, 565

OsclShutdownRequest, 566

OsclShutdownRequest, 566

OsclSocketI, 574  
    Shutdown, 566

OsclSingletonEx, 567  
    ~OsclSingletonEx, 567  
    \_Ptr, 568  
    operator\*, 567  
    operator->, 567  
    OsclSingletonEx, 567  
    set, 568

OsclSingletonRegistryEx, 569  
    getInstance, 569  
    lockAndGetInstance, 569  
    registerInstance, 569  
    registerInstanceAndUnlock, 569

OsclSizeT  
    osclbase, 38

OsclSocket  
    osclconfig\_io.h, 850

OsclSocketCleanup  
    osclconfig\_io.h, 851

OsclSocketI, 570  
    ~OsclSocketI, 571  
    Accept, 571  
    Bind, 571  
    Close, 571  
    Connect, 571  
    GetPeerName, 571  
    Join, 572  
    Listen, 572  
    Logger, 572  
    MakeAddr, 572  
    MakeMulticastGroupInformation, 572  
    NewL, 572  
    Open, 572

OsclAcceptRequest, 574  
 OsclConnectRequest, 574  
 OsclRecvFromRequest, 574  
 OsclRecvRequest, 574  
 OsclSendRequest, 574  
 OsclSendToRequest, 574  
 OsclShutdownRequest, 574  
 OsclSocketRequestAO, 588  
 OsclSocketServI, 591  
 OsclTCPSocket, 574  
 OsclUDPSocket, 574  
 ProcessAccept, 572  
 ProcessConnect, 573  
 ProcessRecv, 573  
 ProcessRecvFrom, 573  
 ProcessSend, 573  
 ProcessSendTo, 573  
 ProcessShutdown, 573  
 Recv, 573  
 RecvFrom, 573  
 RecvFromSuccess, 573  
 RecvSuccess, 573  
 Send, 573  
 SendSuccess, 573  
 SendTo, 573  
 SendToSuccess, 573  
 SetRecvBufferSize, 573  
 SetSockOpt, 574  
 Shutdown, 574  
 Socket, 574  
 ThreadLogoff, 574  
 ThreadLogon, 574  
 OsclSocketIBase, 575  
 ~OsclSocketIBase, 576  
 Accept, 576  
 Bind, 576  
 BindAsync, 576  
 CancelAccept, 577  
 CancelBind, 577  
 CancelConnect, 577  
 CancelFxn, 577  
 CancelListen, 577  
 CancelRecv, 577  
 CancelRecvFrom, 577  
 CancelSend, 577  
 CancelSendTo, 577  
 CancelShutdown, 577  
 Close, 577  
 Connect, 577  
 GetShutdown, 577  
 HasAsyncBind, 577  
 HasAsyncListen, 577  
 iAlloc, 579  
 iSocketServ, 579  
 IsOpen, 577  
 Join, 578  
 Listen, 578  
 ListenAsync, 578  
 Open, 578  
 OsclSocketIBase, 576  
 OsclSocketMethod, 579  
 OsclSocketRequest, 579  
 OsclSocketRequestAO, 579  
 OsclTCPSocket, 579  
 OsclUDPSocket, 579  
 Recv, 578  
 RecvFrom, 578  
 RecvFromSuccess, 578  
 RecvSuccess, 578  
 Send, 578  
 SendSuccess, 578  
 SendTo, 579  
 SendToSuccess, 579  
 Shutdown, 579  
 OsclSocketMethod, 580  
 ~OsclSocketMethod, 581  
 Abort, 581  
 AbortAll, 581  
 Alloc, 581  
 CancelMethod, 581  
 ConstructL, 581  
 iContainer, 582  
 iSocketFxn, 582  
 iSocketRequestAO, 582  
 MethodDone, 582  
 OsclIPSocketI, 451  
 OsclSocketIBase, 579  
 OsclSocketMethod, 581  
 OsclSocketRequestAO, 588  
 Run, 582  
 StartMethod, 582  
 ThreadLogoff, 582  
 ThreadLogon, 582  
 OsclSocketObserver, 584  
 ~OsclSocketObserver, 584  
 HandleSocketEvent, 584  
 OsclSocketRequest  
     OsclSocketIBase, 579  
     OsclSocketRequestAO, 588  
     OsclSocketServI, 591  
 OsclSocketRequestAO, 585  
     ~OsclSocketRequestAO, 586  
     Abort, 586  
     Alloc, 586  
     CleanupParam, 586  
     ConstructL, 586  
     DoCancel, 587  
     GetSocketError, 587

iContainer, 588  
Id, 587  
iParam, 588  
iParamSize, 588  
iSocketError, 588  
NewRequest, 587  
OsclIPSocketI, 451  
OsclSocketI, 588  
OsclSocketIBase, 579  
OsclSocketMethod, 588  
OsclSocketRequest, 588  
OsclSocketRequestAO, 586  
RequestDone, 587  
Run, 587  
SocketI, 587  
SocketObserver, 587  
Success, 588  
OsclSocketSelect  
  osclconfig\_io.h, 851  
OsclSocketServ, 589  
  NewL, 589  
  OsclSocketServI, 591  
OsclSocketServI, 590  
  Close, 590  
  Connect, 590  
  IsServerThread, 591  
  LoopbackSocket, 591  
  NewL, 591  
  OsclIDNSI, 591  
  OsclSocketI, 591  
  OsclSocketRequest, 591  
  OsclSocketServ, 591  
  OsclSocketServRequestList, 591, 596  
  OsclTCPSocketI, 591  
  OsclUDPSocketI, 591  
OsclSocketServIBase, 592  
  ~OsclSocketServIBase, 593  
  Close, 593  
  Connect, 593  
  ESocketServ\_Connected, 592  
  ESocketServ\_Error, 593  
  ESocketServ\_Idle, 592  
  iAlloc, 593  
  iLogger, 593  
  iServerError, 593  
  iServState, 593  
  IsServConnected, 593  
  OsclSocketServIBase, 593  
  State, 593  
  TSocketServState, 592  
OsclSocketServRequestList, 595  
  Add, 595  
  Close, 595  
  Open, 595  
OsclSocketServI, 591, 596  
OsclSocketServRequestList, 595  
Remove, 595  
StartCancel, 595  
WaitOnRequests, 596  
Wakeup, 596  
OsclSocketServRequestQElem, 597  
  iCancel, 597  
  iSelect, 597  
  iSocketRequest, 597  
  OsclSocketServRequestQElem, 597  
OsclSocketStartup  
  osclconfig\_io.h, 851  
OsclSocketTOS, 598  
  ClearTOS, 599  
  EPVCritic\_Ecp, 598  
  EPVFlash, 598  
  EPVHiRel, 598  
  EPVHiThrpt, 598  
  EPVImmediate, 598  
  EPVInetControl, 598  
  EPVLDelay, 598  
  EPVNetControl, 598  
  EPVNoTOS, 598  
  EPVOverrideFlash, 598  
  EPVPriority, 598  
  EPVRoutine, 598  
  GetTOS, 599  
  OsclSocketTOS, 599  
  SetPrecedence, 599  
  SetPriority, 599  
  TPVServicePrecedence, 598  
  TPVServicePriority, 598  
OsclSuccess  
  osclerror, 101  
OsclTagTreeType  
  osclmemory, 64  
OsclTCPSocket, 600  
  NewL, 601  
  osclo, 129  
  OsclSocketI, 574  
  OsclSocketIBase, 579  
OsclTCPSocketI, 602  
  ~OsclTCPSocketI, 603  
  Accept, 603  
  BindAsync, 603  
  CancelAccept, 603  
  CancelBind, 603  
  CancelConnect, 603  
  CancelListen, 603  
  CancelRecv, 603  
  CancelSend, 603  
  CancelShutdown, 603  
  Close, 603

Connect, 604  
 GetAcceptedSocketL, 604  
 GetRecvData, 604  
 GetSendData, 604  
 Listen, 604  
 ListenAsync, 604  
 NewL, 604  
 OsclSocketServI, 591  
 Recv, 604  
 Send, 604  
 Shutdown, 604  
 ThreadLogoff, 605  
 ThreadLogon, 605  
 OsclThread, 606  
     ~OsclThread, 606  
     CanTerminate, 606  
     CompareId, 607  
     Create, 607  
     Exit, 607  
     GetId, 607  
     GetPriority, 608  
     OsclThread, 606  
     Resume, 608  
     SetPriority, 608  
     SleepMillisec, 608  
     Suspend, 609  
     Terminate, 609  
 OsclThread\_State  
     oscl\_thread.h, 819  
 OsclThreadLock, 610  
     ~OsclThreadLock, 610  
     Lock, 610  
     OsclThreadLock, 610  
     Unlock, 610  
 OsclThreadPriority  
     oscl\_thread.h, 819  
 OsclTickCount, 611  
     MsecToTicks, 611  
     TickCount, 611  
     TickCountFrequency, 611  
     TickCountPeriod, 611  
     TicksToMsec, 612  
 OSCLTICKCOUNT\_MAX\_TICKS  
     osclutil, 74  
 OsclTimer, 613  
     ~OsclTimer, 614  
     callback\_timer\_type, 614  
     CallbackTimer< Alloc >, 616  
     Cancel, 614  
     Clear, 614  
     OsclTimer, 614  
     Request, 614  
     SetExactFrequency, 615  
     SetFrequency, 615  
     SetObserver, 615  
     TimerBaseElapsed, 615  
 OsclTimerCompare, 617  
     compare, 617  
     OsclExecSchedulerCommonBase, 430  
 OsclTimerObject, 618  
     ~OsclTimerObject, 619  
     AddToScheduler, 619  
     After, 619  
     Cancel, 619  
     DoCancel, 619  
     IsBusy, 620  
     OsclExecSchedulerCommonBase, 432  
     OsclTimerObject, 619  
     Priority, 620  
     PVActiveBase, 644  
     PVThreadContext, 664  
     RemoveFromScheduler, 620  
     RunError, 620  
     RunIfNotReady, 620  
     SetBusy, 620  
     SetStatus, 621  
     Status, 621  
     StatusRef, 621  
 OsclTimerObserver, 622  
     ~OsclTimerObserver, 622  
     TimeoutOccurred, 622  
 OsclTimerQ, 623  
     Add, 623  
     Construct, 623  
     IsIn, 623  
     Pop, 623  
     PopTop, 623  
     Remove, 623  
     Top, 623  
 OsclTLS, 624  
     ~OsclTLS, 624  
     \_Ptr, 625  
     operator\*, 624  
     operator->, 624  
     OsclTLS, 624  
     set, 625  
 OsclTLSEx, 626  
     ~OsclTLSEx, 626  
     \_Ptr, 627  
     operator\*, 626  
     operator->, 626  
     OsclTLSEx, 626  
     set, 627  
 OsclTLSRegistry, 628  
     getInstance, 628  
     OsclBase, 628  
     registerInstance, 628  
 OsclTLSRegistryEx, 629

getInstance, 629  
registerInstance, 629  
OsclTrapItem, 630  
    OsclTrapItem, 630  
    OsclTrapStack, 630  
        OsclTrapStackItem, 630  
OsclTrapOperation  
    osclerror, 101  
OsclTrapStack, 631  
    OsclError, 631  
    OsclErrorTrap, 631  
    OsclErrorTrapImp, 411, 631  
    OsclTrapItem, 630  
OsclTrapStackItem, 632  
    iCBase, 632  
    iNext, 632  
    iTAny, 632  
    iTrapOperation, 633  
    OsclTrapItem, 630  
    OsclTrapStackItem, 632  
OsclUDPSocket, 634  
    NewL, 635  
    osclio, 129  
    OsclSocketI, 574  
    OsclSocketIBase, 579  
OsclUDPSocketI, 636  
    ~OsclUDPSocketI, 637  
    BindAsync, 637  
    CancelBind, 637  
    CancelRecvFrom, 637  
    CancelSendTo, 637  
    Close, 637  
    GetRecvData, 637  
    GetSendData, 637  
    JoinMulticastGroup, 637  
    NewL, 638  
    OsclSocketServI, 591  
    RecvFrom, 638  
    SendTo, 638  
    SetMulticastTTL, 638  
    ThreadLogoff, 638  
    ThreadLogon, 638  
OsclUid32  
    oscl\_uuid.h, 830  
OsclUnMakeInAddr  
    osclconfig\_io.h, 851  
OsclUnMakeSockAddr  
    osclconfig\_io.h, 851  
osclutil  
    ~OSCL\_HeapString, 91  
    ~OSCL\_StackString, 91  
    ~OSCL\_wHeapString, 91  
    ~OSCL\_wStackString, 91  
    APPEND\_MEDIA\_AT\_END, 92  
    BufferFreeFuncPtr, 74  
    EOSCL\_StringOp\_CompressASCII, 75  
    EOSCL\_StringOp\_UTF16ToUTF8, 75  
    EOSCL\_wStringOp\_ExpandASCII, 75  
    EOSCL\_wStringOp\_UTF8ToUTF16, 75  
    extract\_string, 75  
    get\_cstr, 75, 76  
    get\_maxsize, 76  
    get\_size, 77  
    get\_str, 77, 78  
    GetBufferState, 78  
    GetFragment, 78  
    MAX\_NUMBER\_OF\_BYTE\_PER\_UTF8, 74  
    MediaTimestamp, 74  
    operator=, 78–80  
    oscl\_abs, 80  
    OSCL\_ASCII\_CASE\_MAGIC\_BIT, 92  
    oscl\_asin, 80  
    oscl\_atan, 80  
    oscl\_cos, 80  
    oscl\_exp, 81  
    oscl\_floor, 81  
    OSCL\_HeapString, 81, 82  
    oscl\_isdigit, 74  
    oscl\_log, 82  
    oscl\_log10, 82  
    oscl\_pow, 82  
    oscl\_sin, 82  
    oscl\_snprintf, 82, 83  
    oscl\_sqrt, 83  
    OSCL\_StackString, 83  
    oscl\_str\_escape\_xml, 84  
    oscl\_str\_is\_valid\_utf8, 84  
    oscl\_str\_need\_escape\_xml, 85  
    oscl\_str\_truncate\_utf8, 85  
    oscl\_str\_unescape\_uri, 85, 86  
    oscl\_tan, 86  
    OSCL\_TStrPtrLen, 74  
    oscl\_UncodeToUTF8, 86  
    oscl\_UTF8ToUnicode, 87  
    oscl\_vsnprintf, 87, 88  
    OSCL\_wHeapString, 88  
    OSCL\_wStackString, 88  
    OsclComponentFactory, 74  
    OSCLTICKCOUNT\_MAX\_TICKS, 74  
    PV\_atof, 89  
    PV\_atoi, 89  
    set, 89–91  
    skip\_to\_line\_term, 91  
    skip\_to\_whitespace, 91  
    skip\_whitespace, 91  
    skip\_whitespace\_and\_line\_term, 91  
    StrCSumPtrLen, 74  
    StrPtrLen, 74

TOSCL\_StringOp, 75  
 TOSCL\_wStringOp, 75  
 WStrPtrLen, 75  
 OsclUuid, 639  
     data1, 640  
     data2, 640  
     data3, 640  
     data4, 640  
     operator=, 639  
     operator==, 639  
     OsclUuid, 639  
 OsclValidInetAddr  
     osclconfig\_io.h, 851  
 OsclWriteFD  
     osclconfig\_io.h, 851  
 other  
     Oscl\_TAlloc::rebind, 665  
 OTHER\_ERROR  
     OsclProcStatus, 507  
 other\_chartype  
     OSCL\_FastString, 214  
     OSCL\_HeapString, 234  
     OSCL\_HeapStringA, 236  
     OSCL\_StackString, 295  
     OSCL\_wFastString, 326  
     OSCL\_wHeapString, 329  
     OSCL\_wHeapStringA, 331  
     OSCL\_wStackString, 334  
 OUTOFMEMORY\_ERROR  
     OsclProcStatus, 507  
 overwrite  
     CFastRep, 156  
 pad  
     MM\_AllocBlockFence, 186  
     MM\_AllocBlockHdr, 187  
 pair\_citerator\_citerator  
     Oscl\_Map, 255  
 pair\_iterator\_bool  
     Oscl\_Map, 255  
     Oscl\_TagTree, 307  
 pair\_iterator\_iterator  
     Oscl\_Map, 255  
 pAllocInfo  
     MM\_AllocNode, 192  
 parent  
     Oscl\_Rb\_Tree\_Node\_Base, 292  
     Oscl\_TagTree::Node, 203  
 pBasePosition  
     OsclBinStream, 371  
 pBuffer  
     OsclFileCacheBuffer, 437  
 peakNumAllocs  
     MM\_Stats\_t, 201  
 peakNumBytes  
     MM\_Stats\_t, 201  
 PendComplete  
     OsclActiveObject, 344  
     OsclExecSchedulerCommonBase, 429  
     OsclReadyQ, 517  
 PendForExec  
     OsclActiveObject, 344  
 per\_allocation\_overhead  
     MM\_AuditOverheadStats, 195  
 perms  
     oscl\_stat\_buf, 296  
 PersistHostAddress  
     GetHostNameParam, 169  
 pFileName  
     MM\_AllocInfo, 190  
 pMemBlock  
     MM\_AllocInfo, 190  
     MM\_AllocQueryInfo, 193  
 pMMFIParam  
     OsclMemStatsNode, 487  
 pMMStats  
     OsclMemStatsNode, 487  
 pNext  
     MM\_AllocNode, 192  
 pNode  
     MM\_AllocBlockHdr, 187  
 pointer  
     MemAllocator, 183  
     Oscl\_Map, 255  
     Oscl\_Queue, 272  
     Oscl\_Rb\_Tree, 279  
     Oscl\_Rb\_Tree\_Const\_Iterator, 285  
     Oscl\_Rb\_Tree\_Iterator, 288  
     Oscl\_TagTree::const\_iterator, 162  
     Oscl\_TagTree::iterator, 174  
     Oscl\_TAlloc, 312  
     Oscl\_Vector, 315  
 Pop  
     OsclError, 405  
     OsclTimerQ, 623  
 pop  
     Oscl\_Queue, 273  
     Oscl\_Queue\_Base, 275  
     OsclPriorityQueue, 502  
 pop\_back  
     Oscl\_Vector, 318  
     Oscl\_Vector\_Base, 322  
 pop\_heap  
     OsclPriorityQueue, 502  
     OsclPriorityQueueBase, 505  
 PopDealloc  
     OsclError, 406  
 PopTop

OsclReadyQ, 517  
OsclTimerQ, 623  
port  
    OsclNetworkAddress, 496  
PositionInBlock  
    OsclBinStream, 370  
pPosition  
    OsclBinStream, 371  
pPrev  
    MM\_AllocNode, 192  
Preceeds  
    OsclFileCacheBuffer, 436  
PrepRead  
    OsclFileCacheBuffer, 436  
PrepWrite  
    OsclFileCacheBuffer, 437  
Priority  
    OsclActiveObject, 344  
    OsclTimerObject, 620  
ProcessAccept  
    OsclSocketI, 572  
ProcessConnect  
    OsclSocketI, 573  
ProcessRecv  
    OsclSocketI, 573  
ProcessRecvFrom  
    OsclSocketI, 573  
ProcessSend  
    OsclSocketI, 573  
ProcessSendTo  
    OsclSocketI, 573  
ProcessShutdown  
    OsclSocketI, 573  
pRootNode  
    MM\_AllocBlockHdr, 187  
PSHARED\_ATTRIBUTE\_SETTING\_ERROR  
    OsclProcStatus, 508  
PSHARED\_NOT\_ZERO\_ERROR  
    OsclProcStatus, 508  
pStats  
    MM\_Stats\_CB, 198  
pStatsNode  
    MM\_AllocInfo, 190  
Ptr  
    OsclPtr, 509  
    OsclPtrC, 511  
ptr  
    OsclMemoryFragment, 469  
    StrPtrLen, 679  
    WStrPtrLen, 693  
push  
    Oscl\_Queue, 273  
    Oscl\_Queue\_Base, 275  
    OsclPriorityQueue, 502  
push\_back  
    Oscl\_Vector, 318  
    Oscl\_Vector\_Base, 323  
push\_front  
    Oscl\_Vector, 318  
    Oscl\_Vector\_Base, 323  
push\_heap  
    OsclPriorityQueue, 502  
    OsclPriorityQueueBase, 505  
PushL  
    OsclError, 406  
PV8601TIME\_BUFFER\_SIZE  
    osclbase, 51  
PV8601timeStrBuf  
    osclbase, 38  
PV8601ToRFC822  
    osclbase, 49  
PV\_atof  
    osclutil, 89  
PV\_atoi  
    osclutil, 89  
PV\_CHAR\_CLOSE\_BRACKET  
    oscl\_uuid\_utils.h, 831  
PV\_CHAR\_COMMA  
    oscl\_uuid\_utils.h, 831  
PV\_DYNAMIC\_LOADING\_CONFIG\_FILE\_-  
    PATH  
    osclconfig\_lib.h, 854  
PV\_OSCL\_SOCKET\_STATS\_LOGGING  
    oscl\_socket\_tuneables.h, 804  
PV\_RUNTIME\_LIB\_FILENAME\_EXTENSION  
    osclconfig\_lib.h, 854  
PV\_SCHED\_CHECK\_Q  
    osclproc, 132  
PV\_SCHED\_ENABLE\_LOOP\_STATS  
    osclproc, 132  
PV\_SCHED\_ENABLE\_PERF\_LOGGING  
    osclproc, 132  
PV\_SCHED\_ENABLE\_THREAD\_CONTEXT\_-  
    CHECKS  
    osclproc, 132  
PV\_SCHED\_FAIR\_SCHEDULING  
    osclproc, 132  
PV\_SCHED\_LOG\_Q  
    osclproc, 132  
PV\_SOCKET\_SERVER  
    oscl\_socket\_tuneables.h, 804  
PVActiveBase, 641  
    ~PVActiveBase, 642  
    Activate, 642  
    AddToScheduler, 642  
    Cancel, 642  
    Destroy, 642  
    DoCancel, 642

iAddedNum, 644  
 iBusy, 644  
 iName, 644  
 iPVReadyQLink, 644  
 IsAdded, 642  
 IsInAnyQ, 642  
 iStatus, 644  
 iThreadContext, 644  
 OsclActiveObject, 644  
 OsclExecScheduler, 644  
 OsclExecSchedulerBase, 425  
 OsclExecSchedulerCommonBase, 432  
 OsclReadyCompare, 644  
 OsclReadyQ, 644  
 OsclReadySetPosition, 644  
 OsclSchedulerCommonBase, 644  
 OsclTimerObject, 644  
 PVActiveBase, 642  
 PVThreadContext, 664  
 RemoveFromScheduler, 642  
 Run, 643  
 RunError, 643  
 PVCleanupStack  
 \_OsclHeapBase, 138  
 PVERROR\_Doleave  
 osclerror, 101  
 PVERROR\_IMP\_JUMPS  
 osclerror, 101  
 PVERRORTRAP\_REGISTRY  
 osclerror, 101  
 PVERRORTRAP\_REGISTRY\_ID  
 osclerror, 101  
 PVEXECNAMELEN  
 osclproc, 132  
 PVLogger, 645  
 ~PVLogger, 646  
 AddAppender, 646  
 AddFilter, 646  
 alloc\_type, 646  
 Cleanup, 647  
 DisableAppenderInheritance, 647  
 filter\_status\_type, 646  
 GetLoggerObject, 647  
 GetLogLevel, 647  
 GetNumAppenders, 648  
 GetParent, 648  
 Init, 648  
 IsActive, 648  
 log\_level\_type, 646  
 LogMsgBuffers, 648  
 LogMsgBuffersV, 649  
 LogMsgString, 649  
 LogMsgStringV, 649  
 message\_id\_type, 646  
 PVLogger, 646  
 PVLoggerRegistry, 651  
 RemoveAppender, 650  
 SetLogLevel, 650  
 SetLogLevelAndPropagate, 650  
 SetParent, 650  
 pvlogger.h, 879  
 \_PVLOGGER\_LOGBIN, 881  
 \_PVLOGGER\_LOGBIN\_V, 881  
 \_PVLOGGER\_LOGMSG, 881  
 \_PVLOGGER\_LOGMSG\_V, 881  
 PVLOGGER\_ENABLE, 882  
 PVLOGGER\_INST\_LEVEL, 882  
 PVLOGGER\_INST\_LEVEL\_SUPPORT, 882  
 PVLOGGER\_LEVEL\_UNINITIALIZED, 885  
 PVLOGGER\_LOG\_USE\_ONLY, 882  
 PVLOGGER\_LOGBIN, 882  
 PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
 INST\_HLDBG, 882  
 PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
 INST\_LLDBG, 883  
 PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
 INST\_MLDBG, 883  
 PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
 INST\_PROF, 883  
 PVLOGGER\_LOGBIN\_PVLOGMSG\_-  
 INST\_REL, 883  
 PVLOGGER\_LOGBIN\_V, 883  
 PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_-  
 INST\_HLDBG, 883  
 PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_-  
 INST\_LLDBG, 883  
 PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_-  
 INST\_MLDBG, 883  
 PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_-  
 INST\_PROF, 883  
 PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_-  
 INST\_REL, 883  
 PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_-  
 V\_INST\_MLDBG, 883  
 PVLOGGER\_LOGMSG, 883  
 PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
 INST\_HLDBG, 883  
 PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
 INST\_LLDBG, 884  
 PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
 INST\_MLDBG, 884  
 PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
 INST\_PROF, 884  
 PVLOGGER\_LOGMSG\_PVLOGMSG\_-  
 INST\_REL, 884  
 PVLOGGER\_LOGMSG\_V, 884  
 PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
 INST\_HLDBG, 884  
 PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
 INST\_LLDBG, 884

PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
INST\_MLDBG, 884  
PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
INST\_PROF, 884  
PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
INST\_REL, 884  
PVLOGMSG\_ALERT, 885  
PVLOGMSG\_CRIT, 885  
PVLOGMSG\_DEBUG, 885  
PVLOGMSG\_EMERG, 885  
PVLOGMSG\_ERR, 885  
PVLOGMSG\_FATAL\_ERROR, 885  
PVLOGMSG\_INFO, 886  
PVLOGMSG\_INST\_HLDBG, 884  
PVLOGMSG\_INST\_LLDBG, 884  
PVLOGMSG\_INST\_MLDBG, 884  
PVLOGMSG\_INST\_PROF, 885  
PVLOGMSG\_INST\_REL, 885  
PVLOGMSG\_NONFATAL\_ERROR, 886  
PVLOGMSG\_NOTICE, 886  
PVLOGMSG\_STACK\_TRACE, 886  
PVLOGMSG\_STATISTIC, 886  
PVLOGMSG\_VERBOSE, 886  
PVLOGMSG\_WARNING, 886  
pvlogger\_accessories.h, 887  
    PVLOGGER\_FILTER\_ACCEPT, 887  
    PVLOGGER\_FILTER\_NEUTRAL, 887  
    PVLOGGER\_FILTER\_REJECT, 887  
pvlogger\_c.h, 888  
    PVLOGGER\_C\_INST\_LEVEL, 889  
    pvLogger\_GetLoggerObject, 889  
    pvLogger\_IsActive, 889  
    pvLogger\_LogMsgString, 889  
    PVLOGMSG\_C\_ALERT, 889  
    PVLOGMSG\_C\_CRIT, 889  
    PVLOGMSG\_C\_EMERG, 889  
    PVLOGMSG\_C\_ERR, 889  
    PVLOGMSG\_C\_INFO, 889  
    PVLOGMSG\_C\_INST\_HLDBG, 889  
    PVLOGMSG\_C\_INST\_LLDBG, 889  
    PVLOGMSG\_C\_INST\_MLDBG, 889  
    PVLOGMSG\_C\_INST\_PROF, 889  
    PVLOGMSG\_C\_INST\_REL, 889  
    PVLOGMSG\_C\_NOTICE, 889  
    PVLOGMSG\_C\_STACK\_DEBUG, 889  
    PVLOGMSG\_C\_STACK\_TRACE, 889  
    PVLOGMSG\_C\_WARNING, 889  
PVLOGGER\_C\_INST\_LEVEL  
    pvlogger\_c.h, 889  
PVLOGGER\_ENABLE  
    pvlogger.h, 882  
PVLOGGER\_FILTER\_ACCEPT  
    pvlogger\_accessories.h, 887  
PVLOGGER\_FILTER\_NEUTRAL

pvlogger\_accessories.h, 887  
PVLOGGER\_FILTER\_REJECT  
    pvlogger\_accessories.h, 887  
pvLogger\_GetLoggerObject  
    pvlogger\_c.h, 889  
PVLOGGER\_INST\_LEVEL  
    pvlogger.h, 882  
PVLOGGER\_INST\_LEVEL\_SUPPORT  
    pvlogger.h, 882  
pvLogger\_IsActive  
    pvlogger\_c.h, 889  
PVLOGGER\_LEVEL\_UNINITIALIZED  
    pvlogger.h, 885  
PVLOGGER\_LOG\_USE\_ONLY  
    pvlogger.h, 882  
PVLOGGER\_LOGBIN  
    pvlogger.h, 882  
PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_-  
    HLDBG  
        pvlogger.h, 882  
PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_-  
    LLDBG  
        pvlogger.h, 883  
PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_-  
    MLDBG  
        pvlogger.h, 883  
PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_-  
    PROF  
        pvlogger.h, 883  
PVLOGGER\_LOGBIN\_PVLOGMSG\_INST\_REL  
    pvlogger.h, 883  
PVLOGGER\_LOGBIN\_V  
    pvlogger.h, 883  
PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_-  
    HLDBG  
        pvlogger.h, 883  
PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_-  
    LLDBG  
        pvlogger.h, 883  
PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_-  
    PROF  
        pvlogger.h, 883  
PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_INST\_-  
    REL  
        pvlogger.h, 883  
PVLOGGER\_LOGBIN\_V\_PVLOGMSG\_V\_-  
    INST\_MLDBG  
        pvlogger.h, 883  
PVLOGGER\_LOGMSG  
    pvlogger.h, 883  
PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_-  
    HLDBG  
        pvlogger.h, 883

PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_-  
 LLDBG  
 pvlogger.h, 884

PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_-  
 MLDBG  
 pvlogger.h, 884

PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_-  
 PROF  
 pvlogger.h, 884

PVLOGGER\_LOGMSG\_PVLOGMSG\_INST\_-  
 REL  
 pvlogger.h, 884

PVLOGGER\_LOGMSG\_V  
 pvlogger.h, 884

PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
 INST\_HLDBG  
 pvlogger.h, 884

PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
 INST\_LLDBG  
 pvlogger.h, 884

PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
 INST\_MLDBG  
 pvlogger.h, 884

PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
 INST\_PROF  
 pvlogger.h, 884

PVLOGGER\_LOGMSG\_V\_PVLOGMSG\_-  
 INST\_REL  
 pvlogger.h, 884

pvLogger\_LogMsgString  
 pvlogger\_c.h, 889

pvlogger\_registry.h, 890

PVLoggerAppender, 652  
 ~PVLoggerAppender, 652

PVLoggerFilter, 653  
 ~PVLoggerFilter, 654

filter\_status\_type, 653

FilterOpaqueMessge, 654

FilterString, 654

log\_level\_type, 653

message\_id\_type, 653

PVLoggerLayout, 655  
 ~PVLoggerLayout, 655

FormatOpaqueMessage, 655

FormatString, 655

message\_id\_type, 655

PVLoggerRegistry, 657  
 ~PVLoggerRegistry, 657

alloc\_type, 657

CreatePVLogger, 658

GetPVLoggerObject, 658

GetPVLoggerRegistry, 658

log\_level\_type, 657

PVLogger, 651

PVLoggerRegistry, 657

SetNodeLogLevelExplicit, 658

PVLOGMSG\_ALERT  
 pvlogger.h, 885

PVLOGMSG\_C\_ALERT  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_CRIT  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_EMERG  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_ERR  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_INFO  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_INST\_HLDBG  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_INST\_LLDBG  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_INST\_MLDBG  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_INST\_PROF  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_INST\_REL  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_NOTICE  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_STACK\_DEBUG  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_STACK\_TRACE  
 pvlogger\_c.h, 889

PVLOGMSG\_C\_WARNING  
 pvlogger\_c.h, 889

PVLOGMSG\_CRIT  
 pvlogger.h, 885

PVLOGMSG\_DEBUG  
 pvlogger.h, 885

PVLOGMSG\_EMERG  
 pvlogger.h, 885

PVLOGMSG\_ERR  
 pvlogger.h, 885

PVLOGMSG\_FATAL\_ERROR  
 pvlogger.h, 885

PVLOGMSG\_INFO  
 pvlogger.h, 886

PVLOGMSG\_INST\_HLDBG  
 pvlogger.h, 884

PVLOGMSG\_INST\_LLDBG  
 pvlogger.h, 884

PVLOGMSG\_INST\_MLDBG  
 pvlogger.h, 884

PVLOGMSG\_INST\_PROF

pvlogger.h, 885  
 PVLOGMSG\_INST\_REL  
     pvlogger.h, 885  
 PVLOGMSG\_NONFATAL\_ERROR  
     pvlogger.h, 886  
 PVLOGMSG\_NOTICE  
     pvlogger.h, 886  
 PVLOGMSG\_STACK\_TRACE  
     pvlogger.h, 886  
 PVLOGMSG\_STATISTIC  
     pvlogger.h, 886  
 PVLOGMSG\_VERBOSE  
     pvlogger.h, 886  
 PVLOGMSG\_WARNING  
     pvlogger.h, 886  
 PVMEM\_INST\_LEVEL  
     osclbase, 37  
     osclconfig\_memory.h, 857  
 PVNETWORKADDRESS\_LEN  
     oscl\_socket\_types.h, 805  
 PVOsclBase\_Cleanup  
     osclbase, 49  
 PVOsclBase\_Init  
     osclbase, 49  
 PVSCHEDNAMELEN  
     osclproc, 132  
 PVSchedulerStopper, 660  
     ~PVSchedulerStopper, 660  
     OsclExecSchedulerCommonBase, 432  
     PVSchedulerStopper, 660  
 PVSOCK\_ERR\_BAD\_PARAM  
     oscl\_socket\_imp\_pv.h, 789  
 PVSOCK\_ERR\_NOT\_IMPLEMENTED  
     oscl\_socket\_imp\_pv.h, 789  
 PVSOCK\_ERR\_NOT\_SUPPORTED  
     oscl\_socket\_imp\_pv.h, 789  
 PVSOCK\_ERR\_SERV\_NOT\_CONNECTED  
     oscl\_socket\_imp\_pv.h, 789  
 PVSOCK\_ERR SOCK\_NO\_SERV  
     oscl\_socket\_imp\_pv.h, 789  
 PVSOCK\_ERR SOCK\_NOT\_CONNECTED  
     oscl\_socket\_imp\_pv.h, 789  
 PVSOCK\_ERR SOCK\_NOT\_OPEN  
     oscl\_socket\_imp\_pv.h, 789  
 PVSockBufRecv, 661  
     iLen, 661  
     iMaxLen, 661  
     iPtr, 661  
     PVSockBufRecv, 661  
 PVSockBufSend, 662  
     iLen, 662  
     iPtr, 662  
     PVSockBufSend, 662  
 PVThreadContext, 663  
     ~PVThreadContext, 663  
     EnterThreadContext, 663  
     ExitThreadContext, 663  
     Id, 663  
     IsSameThreadContext, 663  
     OsclActiveObject, 664  
     OsclCoeActiveScheduler, 664  
     OsclCoeActiveSchedulerBase, 664  
     OsclExecScheduler, 664  
     OsclExecSchedulerBase, 664  
     OsclExecSchedulerCommonBase, 432, 664  
     OsclTimerObject, 664  
     PVActiveBase, 664  
     PVThreadContext, 663  
     ThreadHasScheduler, 664  
  
 QUE\_ITER\_BEGIN  
     osclproc, 132  
 QUE\_ITER\_END  
     osclproc, 132  
  
 Rand  
     OsclRand, 513  
 Read  
     Oscl\_File, 221  
     OsclAsyncFile, 349  
     OsclBinIStreamBigEndian, 359  
     OsclFileCache, 435  
     OsclNativeFile, 493  
 read  
     OSCL\_String, 300  
     OSCL\_wString, 337  
 Read\_uint16  
     OsclBinIStreamBigEndian, 359  
     OsclBinIStreamLittleEndian, 362  
 Read\_uint32  
     OsclBinIStreamBigEndian, 359  
     OsclBinIStreamLittleEndian, 362  
 Read\_uint8  
     OsclBinIStream, 356  
 ReadAsync  
     OsclNativeFile, 493  
 ReadAsyncCancel  
     OsclNativeFile, 494  
 rebalance  
     Oscl\_Rb\_Tree\_Base, 283  
 rebalance\_for\_erase  
     Oscl\_Rb\_Tree\_Base, 283  
 Recv  
     osclo, 124  
     OsclRecvMethod, 522  
     OsclRecvRequest, 524  
     OsclSocketI, 573  
     OsclSocketIBase, 578

**OsclTCPSocketI**, 604  
**RecvFrom**  
 osclio, 125  
**OsclRecvFromMethod**, 519  
**OsclRecvFromRequest**, 520  
**OsclSocketI**, 573  
**OsclSocketIBase**, 578  
**OsclUDPSocketI**, 638  
**RecvFromParam**, 666  
 iAddr, 666  
 iBufRecv, 666  
 iFlags, 666  
 iMultiMaxLen, 666  
 iPacketLen, 666  
 iPacketSource, 666  
**RecvFromParam**, 666  
**RecvFromRequest**  
 OsclRecvFromMethod, 519  
**RecvFromSuccess**  
 OsclSocketI, 573  
 OsclSocketIBase, 578  
**RecvParam**, 668  
 iBufRecv, 668  
 iFlags, 668  
 RecvParam, 668  
**RecvRequest**  
 OsclRecvMethod, 522  
**RecvSuccess**  
 OsclSocketI, 573  
 OsclSocketIBase, 578  
**red**  
 Oscl\_Rb\_Tree\_Node\_Base, 291  
**RedBl**  
 Oscl\_Rb\_Tree\_Node\_Base, 291  
**refcount**  
 CHeapRep, 159  
**reference**  
 Oscl\_Map, 255  
 Oscl\_Queue, 272  
 Oscl\_Rb\_Tree, 279  
 Oscl\_Rb\_Tree\_Const\_Iterator, 285  
 Oscl\_Rb\_Tree\_Iterator, 288  
 Oscl\_TagTree::const\_iterator, 162  
 Oscl\_TagTree::iterator, 174  
 Oscl\_TAlloc, 312  
 Oscl\_Vector, 315  
**Register**  
 OsclComponentRegistry, 377  
 OsclRegistryClient, 542  
 OsclRegistryClientImpl, 544  
 OsclRegistryServTlsImpl, 548  
**RegisterForCallback**  
 OsclExecScheduler, 423  
 OsclReadyQ, 517  
**registerInstance**  
 OsclSingletonRegistryEx, 569  
**OsclTLSRegistry**, 628  
**OsclTLSRegistryEx**, 629  
**registerInstanceAndUnlock**  
 OsclSingletonRegistryEx, 569  
**release**  
 OsclExclusiveArrayPtr, 415  
 OsclExclusivePtr, 418  
 OsclExclusivePtrA, 421  
 OSCLMemAutoPtr, 463  
**RELOCK\_MUTEX\_ERROR**  
 OsclProcStatus, 508  
**Remove**  
 OsclDoubleLink, 399  
 OsclReadyQ, 517  
 OsclSocketServRequestList, 595  
 OsclTimerQ, 623  
**remove**  
 OsclPriorityQueue, 502  
 OsclPriorityQueueBase, 505  
**remove\_element**  
 Oscl\_Linked\_List, 247  
 Oscl\_Linked\_List\_Base, 251  
 Oscl\_MTLINKED\_List, 262  
**remove\_ref**  
 CHeapRep, 158  
**RemoveAppender**  
 PVLogger, 650  
**RemoveFixedCache**  
 Oscl\_File, 221  
**RemoveFromScheduler**  
 OsclActiveObject, 344  
 OsclTimerObject, 620  
 PVActiveBase, 642  
**RemoveRef**  
 DNSRequestParam, 166  
**removeRef**  
 Oscl\_DefAllocWithRefCounter, 212  
 OsclMemPoolFixedChunkAllocator, 472  
 OsclMemPoolResizableAllocator, 479  
 OsclRefCounter, 525  
 OsclRefCounterDA, 528  
 OsclRefCounterMTDA, 532  
 OsclRefCounterMTSA, 534  
 OsclRefCounterSA, 536  
**Request**  
 OsclTimer, 614  
**RequestCanceled**  
 OsclExecSchedulerCommonBase, 429  
**RequestDone**  
 OsclDNSRequestAO, 397  
 OsclSocketRequestAO, 587  
**reserve**

Oscl\_Queue\_Base, 275  
Oscl\_Vector\_Base, 323  
OsclPriorityQueue, 502  
ReserveSpace  
    OsclBinStream, 370  
Reset  
    OsclDoubleListBase, 402  
reset  
    BufferState, 146  
    MM\_FailInsertParam, 196  
    MM\_Stats\_t, 200  
    OsclMemStatsNode, 486  
ResetLogPerf  
    OsclExecSchedulerCommonBase, 429  
Resume  
    OsclThread, 608  
ResumeScheduler  
    OsclExecSchedulerCommonBase, 429  
RFC822ToPV8601  
    osclbase, 49  
Right  
    OsclPtrC, 511  
right  
    Oscl\_Rb\_Tree\_Node\_Base, 292  
rotate\_left  
    Oscl\_Rb\_Tree\_Base, 283  
rotate\_right  
    Oscl\_Rb\_Tree\_Base, 283  
Run  
    CallbackTimer, 152  
    OsclIDNSMethod, 393  
    OsclIDNSRequestAO, 397  
    OsclSocketMethod, 582  
    OsclSocketRequestAO, 587  
    PVActiveBase, 643  
RunError  
    OsclActiveObject, 345  
    OsclTimerObject, 620  
    PVActiveBase, 643  
RunIfNotReady  
    OsclActiveObject, 345  
    OsclTimerObject, 620  
RunSchedulerNonBlocking  
    OsclExecScheduler, 423  
save\_registry  
    TLSStorageOps, 687  
second  
    Oscl\_Pair, 270  
SECONDS  
    osclbase, 38  
Seed  
    OsclRand, 513  
Seek  
    Oscl\_File, 222  
    OsclAsyncFile, 349  
    OsclBinStream, 370  
    OsclFileCache, 435  
    OsclNativeFile, 494  
    seek\_type  
        Oscl\_File, 218  
SEEKCUR  
    Oscl\_File, 218  
SEEKEND  
    Oscl\_File, 218  
seekFromCurrentPosition  
    OsclBinStream, 370  
SEEKSET  
    Oscl\_File, 218  
self  
    Oscl\_Map, 255  
    Oscl\_Rb\_Tree\_Const\_Iterator, 285  
    Oscl\_Rb\_Tree\_Iterator, 288  
    Oscl\_TagTree::const\_iterator, 162  
    Oscl\_TagTree::iterator, 174  
SEM\_NOT\_SIGNALLED\_ERROR  
    OsclProcStatus, 508  
Send  
    osclo, 125  
    OsclSendMethod, 557  
    OsclSendRequest, 559  
    OsclSocketI, 573  
    OsclSocketIBase, 578  
    OsclTCPSocketI, 604  
SendParam, 669  
    iBufSend, 669  
    iFlags, 669  
    iXferLen, 669  
    SendParam, 669  
SendRequest  
    OsclSendMethod, 557  
SendSuccess  
    OsclSocketI, 573  
    OsclSocketIBase, 578  
SendTo  
    osclo, 125  
    OsclSendToMethod, 560  
    OsclSendToRequest, 562  
    OsclSocketI, 573  
    OsclSocketIBase, 579  
    OsclUDPSocketI, 638  
SendToParam, 670  
    ~SendToParam, 670  
    iAddr, 670  
    iBufSend, 670  
    iFlags, 670  
    iXferLen, 670  
    SendToParam, 670

**SendToRequest**  
 OsclSendToMethod, 560  
**SendToSuccess**  
 OsclSocketI, 573  
 OsclSocketIBase, 579  
**Serv**  
 OsclIDNSRequestAO, 398  
**Set**  
 OsclDoubleRunner, 403  
 OsclNameString, 490  
 OsclPtr, 509  
 OsclPtrC, 512  
**set**  
 CHeapRep, 158  
 CStackRep, 164  
 OSCL\_FastString, 215, 216  
 OSCL\_HeapStringA, 238, 239  
 OSCL\_wFastString, 327  
 OSCL\_wHeapStringA, 332  
 OsclExclusiveArrayPtr, 415  
 OsclExclusivePtr, 418  
 OsclExclusivePtrA, 421  
 OsclSingletonEx, 568  
 OsclTLS, 625  
 OsclTLSEx, 627  
 oscutil, 89–91  
**set\_from\_ntp\_time**  
 TimeValue, 686  
**set\_from\_system\_time**  
 NTPTime, 207  
**set\_int64**  
 Oscl\_Int64\_Utils, 241  
**set\_len**  
 OSCL\_String, 300  
 OSCL\_wString, 337  
**set\_length**  
 OSCL\_FastString, 216  
 OSCL\_wFastString, 327  
**set\_next**  
 Oscl\_Opaque\_Type\_Alloc\_LL, 267  
**set\_r**  
 CFastRep, 158  
**set\_rep**  
 CHeapRep, 158  
 OSCL\_String, 300, 301  
 OSCL\_wString, 337  
**set\_to\_current\_time**  
 NTPTime, 207  
 TimeValue, 686  
**set\_to\_zero**  
 TimeValue, 686  
**set\_uint64**  
 Oscl\_Int64\_Utils, 241  
**set\_w**  
 CFastRep, 156  
 set\_zulu  
 TimeValue, 686  
**setAllocNodeFlag**  
 MM\_AllocBlockHdr, 187  
**SetAsyncReadBufferSize**  
 Oscl\_File, 222  
**SetBusy**  
 OsclActiveObject, 345  
 OsclTimerObject, 620  
**SetCacheObserver**  
 Oscl\_File, 222  
**setCheckSum**  
 StrCSumPtrLen, 676  
**SetExactFrequency**  
 OsclTimer, 615  
**SetFileHandle**  
 Oscl\_File, 222  
**SetFrequency**  
 OsclTimer, 615  
**SetInUse**  
 OsclAsyncFileBuffer, 352  
**SetLength**  
 OsclPtr, 509  
 OsclPtrC, 512  
**SetLoggingEnable**  
 Oscl\_File, 223  
**SetLogLevel**  
 PVLogger, 650  
**SetLogLevelAndPropagate**  
 PVLogger, 650  
**setMaxSzForNewMemPoolBuffer**  
 OsclMemPoolResizableAllocator, 479  
**SetMulticastTTL**  
 oscilio, 126  
 OsclUDPSocketI, 638  
**SetNativeAccessMode**  
 Oscl\_File, 223  
**SetNativeBufferSize**  
 Oscl\_File, 223  
**SetNodeLogLevelExplicit**  
 PVLoggerRegistry, 658  
**SetObserver**  
 OsclTimer, 615  
**SetOffset**  
 OsclAsyncFileBuffer, 352  
 OsclDoubleListBase, 402  
**SetOptionToReuseAddress**  
 oscilio, 126  
 OsclIPSocketI, 450  
**SetParent**  
 PVLogger, 650  
**SetPosition**  
 OsclFileCacheBuffer, 437

SetPrecedence  
    OsclSocketTOS, 599

SetPriority  
    OsclSocketTOS, 599  
    OsclThread, 608

setPtrLen  
    StrCSumPtrLen, 676  
    StrPtrLen, 678  
    WStrPtrLen, 693

SetPVCacheSize  
    Oscl\_File, 223

SetRecvBufferSize  
    osclio, 127  
    OsclIPSocketI, 450  
    OsclSocketI, 573

setrep\_to\_char  
    OSCL\_String, 301

setrep\_to\_wide\_char  
    OSCL\_wString, 337

SetScheduler  
    OsclExecSchedulerCommonBase, 429

SetSize  
    Oscl\_File, 223  
    OsclNativeFile, 494

SetSockOpt  
    OsclSocketI, 574

SetStatus  
    OsclActiveObject, 345  
    OsclTimerObject, 621

SetSummaryStatsLoggingEnable  
    Oscl\_File, 224

SetTimestamp  
    MediaData, 180

SetToHead  
    OsclDoubleRunner, 403

SetTOS  
    osclio, 127  
    OsclIPSocketI, 450

SetToTail  
    OsclDoubleRunner, 404

setWithoutOwnership  
    OSCLMemAutoPtr, 464

Shutdown  
    osclio, 127  
    OsclShutdownMethod, 565  
    OsclShutdownRequest, 566  
    OsclSocketI, 574  
    OsclSocketIBase, 579  
    OsclTCPSocketI, 604

ShutdownParam, 671  
    iHow, 671  
    ShutdownParam, 671

ShutdownRequest  
    OsclShutdownMethod, 565

Signal  
    OsclSemaphore, 555

Size  
    Oscl\_File, 224  
    OsclAsyncFile, 349  
    OsclNativeFile, 494

size  
    CFastRep, 156  
    CHeapRep, 159  
    CStackRep, 165  
    MM\_AllocBlockHdr, 187  
    MM\_AllocInfo, 190  
    MM\_AllocQueryInfo, 193  
    Oscl\_Map, 259  
    Oscl\_Queue\_Base, 276  
    Oscl\_Rb\_Tree, 282  
    Oscl\_TagTree, 309  
    Oscl\_Vector\_Base, 323  
    OsclPriorityQueue, 503  
    StrPtrLen, 679  
    WStrPtrLen, 693

size\_type  
    Oscl\_Map, 255  
    Oscl\_Queue, 272  
    Oscl\_Rb\_Tree, 279  
    Oscl\_Tag\_Base, 304  
    Oscl\_TagTree, 307  
    Oscl\_TAlloc, 312

sizeof\_T  
    Oscl\_Linked\_List\_Base, 252  
    Oscl\_Queue\_Base, 276  
    Oscl\_Vector\_Base, 324

skip\_to\_line\_term  
    osclutil, 91

skip\_to\_whitespace  
    osclutil, 91

skip\_whitespace  
    osclutil, 91

skip\_whitespace\_and\_line\_term  
    osclutil, 91

SLEEP\_ONE\_SEC  
    osclconfig\_util.h, 877

SleepMillisec  
    OsclThread, 608

Socket  
    OsclSocketI, 574

SocketI  
    OsclSocketRequestAO, 587

SocketObserver  
    OsclSocketRequestAO, 587

SocketRequestParam, 672  
    iFxn, 673

SocketRequestParam, 673

SocketServ

OsclIPSocketI, 450  
 sort\_children  
     Oscl\_TagTree::Node, 202  
 specialFragBuffer  
     OsclBinStream, 371  
 Start  
     OsclFileStats, 441  
 Start\_on\_creation  
     oscl\_thread.h, 819  
 StartAsyncRead  
     OsclAsyncFileBuffer, 352  
 StartCancel  
     OsclSocketServRequestList, 595  
 StartMethod  
     OsclIDNSMethod, 393  
     OsclSocketMethod, 582  
 StartNativeScheduler  
     OsclExecSchedulerCommonBase, 429  
 StartScheduler  
     OsclExecSchedulerCommonBase, 429  
 State  
     OsclSocketServIBase, 593  
 state  
     OsclBinStream, 371  
 state\_t  
     OsclBinStream, 369  
 StaticJump  
     OsclJump, 452  
 stats\_overhead  
     MM\_AuditOverheadStats, 195  
 Status  
     OsclActiveObject, 345  
     OsclTimerObject, 621  
 status\_t  
     BufFragStatusClass, 151  
 StatusRef  
     OsclActiveObject, 345  
     OsclTimerObject, 621  
 StopScheduler  
     OsclExecSchedulerCommonBase, 429  
 Str  
     OsclNameString, 490  
 StrCSumPtrLen, 674  
     checkSum, 676  
     CheckSumType, 675  
     getCheckSum, 675  
     isCIEquivalentTo, 675  
     operator=, 675  
     operator==, 676  
     osclutil, 74  
     setCheckSum, 676  
     setPtrLen, 676  
     StrCSumPtrLen, 675  
 StrPtrLen, 677  
     c\_str, 678  
     isCIEquivalentTo, 678  
     isCIPrefixOf, 678  
     isLetter, 678  
     len, 679  
     length, 678  
     operator=, 678  
     operator==, 678  
     osclutil, 74  
     ptr, 679  
     setPtrLen, 678  
     size, 679  
     StrPtrLen, 677, 678  
 Success  
     OsclIDNSRequestAO, 398  
     OsclRecvFromRequest, 520  
     OsclRecvRequest, 524  
     OsclSendRequest, 559  
     OsclSendToRequest, 562  
     OsclSocketRequestAO, 588  
 SUCCESS\_ERROR  
     OsclProcStatus, 507  
 Suspend  
     OsclThread, 609  
 Suspend\_on\_creation  
     oscl\_thread.h, 819  
 SuspendScheduler  
     OsclExecSchedulerCommonBase, 429  
 swap  
     Oscl\_Opaque\_Type\_Compare, 268  
     OsclPriorityQueue, 503  
 SYSTEM\_RESOURCES\_UNAVAILABLE\_-  
     ERROR  
     OsclProcStatus, 508  
 tag  
     MM\_AllocQueryInfo, 193  
     MM\_Stats\_CB, 198  
     Oscl\_Tag, 303  
     Oscl\_TagTree::Node, 203  
     OsclMemStatsNode, 487  
 tag\_ancestor  
     Oscl\_Tag\_Base, 304  
 tag\_base\_type  
     Oscl\_Tag\_Base, 304  
     Oscl\_TagTree, 307  
 tag\_base\_unit  
     Oscl\_Tag\_Base, 304  
 tag\_cmp  
     Oscl\_Tag\_Base, 304  
 tag\_copy  
     Oscl\_Tag\_Base, 305  
 tag\_depth  
     Oscl\_Tag\_Base, 305

tag\_len  
    Oscl\_Tag\_Base, 305

tag\_type  
    Oscl\_TagTree, 307

tagAllocator  
    Oscl\_Tag, 303

TagTree\_Allocator  
    osclmemory, 64

Tail  
    OsclDoubleList, 400  
    OsclPriorityList, 499

tail  
    Oscl\_Linked\_List\_Base, 252

takeOwnership  
    OSCLMemAutoPtr, 464

TDNSRequestParamAllocator  
    oscl\_dns\_param.h, 710

Tell  
    Oscl\_File, 224  
    OsclAsyncFile, 349  
    OsclFileCache, 435  
    OsclNativeFile, 494

tellg  
    OsclBinStream, 371

Terminate  
    OsclThread, 609

the\_list  
    Oscl\_MTLinked\_List, 263

THREAD\_1\_INACTIVE\_ERROR  
    OsclProcStatus, 508

THREAD\_BLOCK\_ERROR  
    OsclProcStatus, 508

THREAD\_NOT\_OWN\_MUTEX\_ERROR  
    OsclProcStatus, 508

ThreadHasScheduler  
    PVThreadContext, 664

ThreadLogoff  
    osclio, 128  
    OsclIPSocketI, 450  
    OsclReadyQ, 517  
    OsclSocketI, 574  
    OsclSocketMethod, 582  
    OsclTCPSocketI, 605  
    OsclUDPSocketI, 638

ThreadLogon  
    osclio, 128  
    OsclIPSocketI, 450  
    OsclReadyQ, 517  
    OsclSocketI, 574  
    OsclSocketMethod, 582  
    OsclTCPSocketI, 605  
    OsclUDPSocketI, 638

ThreadPriorityAboveNormal  
    oscl\_thread.h, 820

ThreadPriorityBelowNormal  
    oscl\_thread.h, 820

ThreadPriorityHighest  
    oscl\_thread.h, 820

ThreadPriorityLow  
    oscl\_thread.h, 820

ThreadPriorityLowest  
    oscl\_thread.h, 820

ThreadPriorityNormal  
    oscl\_thread.h, 820

ThreadPriorityTimeCritical  
    oscl\_thread.h, 820

TickCount  
    OsclTickCount, 611

TickCountFrequency  
    OsclTickCount, 611

TickCountPeriod  
    OsclTickCount, 611

TicksToMsec  
    OsclTickCount, 612

TimeoutOccurred  
    OsclTimerObserver, 622

TimerBaseElapsed  
    CallbackTimerObserver, 154  
    OsclTimer, 615

TimerCallback  
    OsclReadyQ, 517

timestamp  
    MediaData, 181

TimeUnits  
    osclbase, 38

TimeValue, 680  
    get\_ISO8601\_str\_time, 683  
    get\_local\_time, 683  
    get\_pv8601\_str\_time, 683  
    get\_rfc822\_gmtime\_str, 684  
    get\_sec, 684  
    get\_str\_ctime, 684  
    get\_timeval\_ptr, 684  
    get\_timevalue\_in\_usec, 685  
    get\_usec, 685  
    is\_zero, 685  
    is\_zulu, 685  
    NTPTime, 686  
    operator<, 686  
    operator<=, 686  
    operator>, 686  
    operator>=, 686  
    operator\*=, 685  
    operator+=, 685  
    operator-=, 685  
    operator=, 685  
    operator==, 686  
    set\_from\_ntp\_time, 686

set\_to\_current\_time, 686  
 set\_to\_zero, 686  
 set\_zulu, 686  
 TimeValue, 682, 683  
 to\_msec, 686  
**TIpMReq**  
 osclconfig\_io.h, 851  
**TLSStorageOps**, 687  
 get\_registry, 687  
 save\_registry, 687  
**to\_msec**  
 TimeValue, 686  
**to\_system\_time**  
 NTPTime, 207  
**TOO\_MANY\_FRAGS**  
 BufFragStatusClass, 151  
**TOO\_MANY\_THREADS\_ERROR**  
 OsclProcStatus, 507  
**Top**  
 OsclJump, 452  
 OsclReadyQ, 517  
 OsclTimerQ, 623  
**top**  
 OsclPriorityQueue, 503  
**TOSCL\_StringOp**  
 oscutil, 75  
**TOSCL\_wStringOp**  
 oscutil, 75  
**TOsclBasicLockObject**  
 osclconfig\_unix\_android.h, 872  
 osclconfig\_unix\_common.h, 876  
**TOsclConditionObject**  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866  
**TOsclFileHandle**  
 osclio, 108  
**TOsclFileOffset**  
 osclconfig\_io.h, 851  
**TOsclFileOffsetInt32**  
 osclio, 108  
**TOsclFileOp**  
 osclio, 109  
**TOsclHostent**  
 osclconfig\_io.h, 851  
**TOsclMutexObject**  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866  
**TOsclReady**  
 osclproc, 133  
**TOsclSemaphoreObject**  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866  
**TOsclSockAddr**  
 osclconfig\_io.h, 851  
**TOsclSockAddrLen**  
 osclconfig\_io.h, 851  
**TOsclSocket**  
 osclconfig\_io.h, 851  
**TOsclSocketServStatEvent**  
 oscl\_socket\_stats.h, 802  
**TOsclSocketStatEvent**  
 oscl\_socket\_stats.h, 802  
**TOsclThreadFuncArg**  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866  
**TOsclThreadFuncPtr**  
 oscl\_thread.h, 819  
**TOsclThreadFuncRet**  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866  
**TOsclThreadId**  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866  
**TOsclThreadObject**  
 osclconfig\_proc\_unix\_android.h, 864  
 osclconfig\_proc\_unix\_common.h, 866  
**TOsclThreadTerminate**  
 oscl\_thread.h, 820  
**TOsclTlsKey**  
 oscibase, 38  
 osclconfig\_unix\_android.h, 872  
 osclconfig\_unix\_common.h, 876  
**totalbytes**  
 oscl\_fsstat, 232  
**totalNumAllocs**  
 MM\_Stats\_t, 201  
**totalNumBytes**  
 MM\_Stats\_t, 201  
**TPVDNSEvent**  
 osclio, 109  
**TPVDNSFxN**  
 osclio, 109  
**TPVServicePrecedence**  
 OsclSocketTOS, 598  
**TPVServicePriority**  
 OsclSocketTOS, 598  
**TPVSocketEvent**  
 oscl\_socket\_types.h, 805  
**TPVSocketFxn**  
 oscl\_socket\_types.h, 806  
**TPVSocketOptionLevel**  
 oscl\_socket\_types.h, 806  
**TPVSocketOptionName**  
 oscl\_socket\_types.h, 806  
**TPVSocketShutdown**  
 oscl\_socket\_types.h, 806  
**TPVThreadContext**  
 osclproc, 133

Trap  
    OsclErrorTrapImp, 410

TrapNoTls  
    OsclErrorTrapImp, 410

TReadyQueLink, 688  
    iAOPriority, 688  
    iIsIn, 688  
    iSeqNum, 688  
    iTimeQueuedTicks, 688  
    iTimeToRunTicks, 688  
    TReadyQueLink, 688

trim  
    OsclMemPoolResizableAllocator, 479

TryLock  
    OsclMutex, 489

TryWait  
    OsclSemaphore, 555

TSocketServState  
    OsclSocketServIBase, 592

TSymbianAccessMode  
    Oscl\_File, 218

uint  
    osclbase, 38

UINT64  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876

uint64  
    osclbase, 38

UINT64\_HILO  
    osclconfig\_unix\_android.h, 872  
    osclconfig\_unix\_common.h, 876

Unbind  
    osclbase, 50

UninstallScheduler  
    OsclExecSchedulerCommonBase, 430

unix\_ntp\_offset  
    osclbase, 51

Unlock  
    OsclLockBase, 455  
    OsclMutex, 489  
    OsclNullLock, 497  
    OsclThreadLock, 610

UnRegister  
    OsclRegistryClient, 543  
    OsclRegistryClientImpl, 544  
    OsclRegistryServTlsImpl, 548

Unregister  
    OsclComponentRegistry, 377

UnTrap  
    OsclErrorTrapImp, 410

update  
    MM\_Stats\_t, 201

UpdateData

    OsclAsyncFileBuffer, 352

updateEnd  
    OsclFileCacheBuffer, 437

updateStart  
    OsclFileCacheBuffer, 437

UpdateTimers  
    OsclExecSchedulerCommonBase, 430

UpdateTimersMsec  
    OsclExecSchedulerCommonBase, 430

upper\_bound  
    Oscl\_Map, 259  
    Oscl\_Rb\_Tree, 282

usableSize  
    OsclFileCacheBuffer, 437

USEC\_PER\_SEC  
    osclbase, 51

validate  
    OsclPriorityQueue, 503

validateblock  
    OsclMemPoolResizableAllocator, 480

Value  
    OsclAOStatus, 347

value  
    Oscl\_Rb\_Tree\_Node, 290  
    Oscl\_TagTree::Node, 203

value\_comp  
    Oscl\_Map, 259

value\_compare  
    Oscl\_Map::value\_compare, 690

value\_type  
    Oscl\_Map, 255  
    Oscl\_Queue, 272  
    Oscl\_Rb\_Tree, 279  
    Oscl\_Rb\_Tree\_Const\_Iterator, 285  
    Oscl\_Rb\_Tree\_Iterator, 288  
    Oscl\_Rb\_Tree\_Node, 290  
    Oscl\_TagTree, 307  
    Oscl\_TAlloc, 312  
    Oscl\_Vector, 315  
    OsclPriorityQueue, 501

vec  
    OsclPriorityQueue, 503

Wait  
    OsclSemaphore, 555

WAIT\_ABANDONED\_ERROR  
    OsclProcStatus, 508

WAIT\_TIMEOUT\_ERROR  
    OsclProcStatus, 508

WaitAndPopTop  
    OsclReadyQ, 517

WaitForReadyAO  
    OsclExecSchedulerCommonBase, 430

WaitForRequestComplete  
    OsclReadyQ, [517](#)

WaitOnRequests  
    OsclSocketServRequestList, [596](#)

Wakeup  
    OsclSocketServRequestList, [596](#)

writable  
    CFastRep, [156](#)

Write  
    Oscl\_File, [224](#)  
    OsclAsyncFile, [349](#)  
    OsclFileCache, [435](#)  
    OsclNativeFile, [494](#)

write  
    OSCL\_String, [301](#)  
    OSCL\_wString, [338](#)  
    OsclBinOStream, [363](#)

WriteUnsignedLong  
    OsclBinOStreamBigEndian, [365](#)  
    OsclBinOStreamLittleEndian, [367](#)

WriteUnsignedShort  
    OsclBinOStreamBigEndian, [365](#)  
    OsclBinOStreamLittleEndian, [367](#)

WriteUpdatesToFile  
    OsclFileCacheBuffer, [437](#)

WStrPtrLen, [692](#)  
    c\_str, [692](#)  
    isCIEquivalentTo, [692](#)  
    len, [693](#)  
    length, [693](#)  
    operator=, [693](#)  
    operator==, [693](#)  
    osclutil, [75](#)  
    ptr, [693](#)  
    setPtrLen, [693](#)  
    size, [693](#)  
    WStrPtrLen, [692](#)

xsubi  
    MM\_FailInsertParam, [197](#)

Zero  
    OsclPtr, [509](#)  
    OsclPtrC, [512](#)