

LibBeamApi

0.3.8

Generated by Doxygen 1.8.5

Mon Dec 11 2017 13:25:07

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	7
3.1	Class List	7
4	File Index	11
4.1	File List	11
5	Namespace Documentation	15
5.1	Boapns Namespace Reference	15
5.1.1	Variable Documentation	15
5.1.1.1	apiVersion	15
6	Class Documentation	17
6.1	BArray< T > Class Template Reference	17
6.1.1	Detailed Description	17
6.1.2	Member Typedef Documentation	18
6.1.2.1	SortFunc	18
6.1.3	Constructor & Destructor Documentation	18
6.1.3.1	BArray	18
6.1.3.2	BArray	18
6.1.3.3	BArray	18
6.1.4	Member Function Documentation	18
6.1.4.1	append	18
6.1.4.2	append	18
6.1.4.3	del	18
6.1.4.4	insert	18
6.1.4.5	number	18
6.1.4.6	rear	18

6.1.4.7	sort	18
6.2	BAtomic< Type > Class Template Reference	18
6.2.1	Detailed Description	19
6.2.2	Constructor & Destructor Documentation	19
6.2.2.1	BAtomic	19
6.2.3	Member Function Documentation	19
6.2.3.1	add	19
6.2.3.2	getValue	19
6.2.3.3	operator Type	19
6.2.3.4	operator++	19
6.2.3.5	operator++	19
6.2.3.6	operator--	19
6.2.3.7	operator--	19
6.2.4	Member Data Documentation	19
6.2.4.1	ovalue	19
6.3	BAtomicCount Class Reference	19
6.3.1	Detailed Description	20
6.3.2	Constructor & Destructor Documentation	20
6.3.2.1	BAtomicCount	20
6.3.3	Member Function Documentation	20
6.3.3.1	add	20
6.3.3.2	getValue	20
6.3.3.3	operator long	20
6.3.3.4	operator++	20
6.3.3.5	operator++	20
6.3.3.6	operator--	20
6.3.3.7	operator--	20
6.3.4	Member Data Documentation	20
6.3.4.1	ovalue	20
6.4	BBuffer Class Reference	20
6.4.1	Constructor & Destructor Documentation	21
6.4.1.1	BBuffer	21
6.4.1.2	~BBuffer	21
6.4.2	Member Function Documentation	21
6.4.2.1	data	21
6.4.2.2	resize	21
6.4.2.3	setData	21
6.4.2.4	setSize	21
6.4.2.5	size	21
6.4.2.6	writeData	22

6.4.3	Member Data Documentation	22
6.4.3.1	odata	22
6.4.3.2	odataSize	22
6.4.3.3	osize	22
6.5	BBufferStore Class Reference	22
6.5.1	Constructor & Destructor Documentation	23
6.5.1.1	BBufferStore	23
6.5.1.2	~BBufferStore	23
6.5.2	Member Function Documentation	23
6.5.2.1	getHexString	23
6.5.2.2	getPos	23
6.5.2.3	pop	23
6.5.2.4	pop	23
6.5.2.5	pop	23
6.5.2.6	pop	23
6.5.2.7	pop	23
6.5.2.8	pop	23
6.5.2.9	pop	23
6.5.2.10	pop	23
6.5.2.11	pop	23
6.5.2.12	pop	23
6.5.2.13	pop	24
6.5.2.14	pop	24
6.5.2.15	pop	24
6.5.2.16	pop	24
6.5.2.17	pop	24
6.5.2.18	push	24
6.5.2.19	push	24
6.5.2.20	push	24
6.5.2.21	push	24
6.5.2.22	push	24
6.5.2.23	push	24
6.5.2.24	push	24
6.5.2.25	push	24
6.5.2.26	push	24
6.5.2.27	push	24
6.5.2.28	push	24
6.5.2.29	push	24
6.5.2.30	push	24
6.5.2.31	push	24

6.5.2.32	push	24
6.5.2.33	setHexString	24
6.5.2.34	setPos	24
6.5.3	Member Data Documentation	24
6.5.3.1	opos	24
6.5.3.2	oswapBytes	24
6.6	BComms Class Reference	25
6.6.1	Member Enumeration Documentation	25
6.6.1.1	Wait	25
6.6.2	Constructor & Destructor Documentation	26
6.6.2.1	BComms	26
6.6.2.2	~BComms	26
6.6.3	Member Function Documentation	26
6.6.3.1	eventQueue	26
6.6.3.2	init	26
6.6.3.3	packetMode	26
6.6.3.4	read	26
6.6.3.5	readAvailable	26
6.6.3.6	setPacketMode	26
6.6.3.7	setTimeout	26
6.6.3.8	wait	26
6.6.3.9	write	26
6.6.3.10	writeAvailable	26
6.6.4	Member Data Documentation	26
6.6.4.1	oevent	26
6.6.4.2	oeventNum	26
6.6.4.3	oeventQueue	26
6.6.4.4	opacketMode	26
6.6.4.5	otimeout	26
6.7	BCond Class Reference	27
6.7.1	Constructor & Destructor Documentation	27
6.7.1.1	BCond	27
6.7.1.2	~BCond	27
6.7.2	Member Function Documentation	27
6.7.2.1	signal	27
6.7.2.2	timedWait	27
6.7.2.3	wait	27
6.7.3	Member Data Documentation	27
6.7.3.1	ocond	27
6.7.3.2	omutex	27

6.8	BCondBool Class Reference	27
6.8.1	Detailed Description	28
6.8.2	Constructor & Destructor Documentation	28
6.8.2.1	BCondBool	28
6.8.2.2	~BCondBool	28
6.8.3	Member Function Documentation	28
6.8.3.1	clear	28
6.8.3.2	operator int	28
6.8.3.3	set	28
6.8.3.4	timedWait	28
6.8.3.5	value	28
6.8.3.6	wait	29
6.8.4	Member Data Documentation	29
6.8.4.1	ocond	29
6.8.4.2	omutex	29
6.8.4.3	ovalue	29
6.9	BCondInt Class Reference	29
6.9.1	Detailed Description	30
6.9.2	Constructor & Destructor Documentation	30
6.9.2.1	BCondInt	30
6.9.2.2	~BCondInt	30
6.9.3	Member Function Documentation	30
6.9.3.1	decrement	30
6.9.3.2	increment	30
6.9.3.3	operator++	30
6.9.3.4	operator+=	30
6.9.3.5	operator--	30
6.9.3.6	operator-=	30
6.9.3.7	setValue	30
6.9.3.8	value	30
6.9.3.9	waitLessThan	31
6.9.3.10	waitLessThanOrEqual	31
6.9.3.11	waitMoreThanOrEqual	31
6.9.4	Member Data Documentation	31
6.9.4.1	ocond	31
6.9.4.2	omutex	31
6.9.4.3	ovalue	31
6.10	BCondResource Class Reference	31
6.10.1	Detailed Description	32
6.10.2	Constructor & Destructor Documentation	32

6.10.2.1	BCondResource	32
6.10.2.2	~BCondResource	32
6.10.3	Member Function Documentation	32
6.10.3.1	end	32
6.10.3.2	inUse	32
6.10.3.3	lock	32
6.10.3.4	locked	32
6.10.3.5	start	32
6.10.3.6	unlock	32
6.10.4	Member Data Documentation	32
6.10.4.1	ocond	32
6.10.4.2	olock	32
6.10.4.3	omutex	32
6.10.4.4	ouse	32
6.11	BCondValue Class Reference	32
6.11.1	Detailed Description	33
6.11.2	Constructor & Destructor Documentation	33
6.11.2.1	BCondValue	33
6.11.2.2	~BCondValue	33
6.11.3	Member Function Documentation	33
6.11.3.1	decrement	33
6.11.3.2	increment	33
6.11.3.3	operator++	34
6.11.3.4	operator+=	34
6.11.3.5	operator--	34
6.11.3.6	operator-=	34
6.11.3.7	setValue	34
6.11.3.8	value	34
6.11.3.9	waitLessThan	34
6.11.3.10	waitLessThanOrEqual	34
6.11.3.11	waitMoreThanOrEqual	34
6.11.4	Member Data Documentation	34
6.11.4.1	ocond	34
6.11.4.2	omutex	34
6.11.4.3	ovalue	34
6.12	BCondWrap Class Reference	35
6.12.1	Constructor & Destructor Documentation	35
6.12.1.1	BCondWrap	35
6.12.1.2	~BCondWrap	35
6.12.2	Member Function Documentation	35

6.12.2.1	decrement	35
6.12.2.2	diff	36
6.12.2.3	increment	36
6.12.2.4	operator++	36
6.12.2.5	operator+=	36
6.12.2.6	operator--	36
6.12.2.7	operator-=	36
6.12.2.8	setValue	36
6.12.2.9	value	36
6.12.2.10	waitLessThan	36
6.12.2.11	waitLessThanOrEqual	36
6.12.2.12	waitMoreThanOrEqual	36
6.12.3	Member Data Documentation	36
6.12.3.1	ocond	36
6.12.3.2	omutex	36
6.12.3.3	ovalue	37
6.13	BConfig Class Reference	37
6.13.1	Detailed Description	37
6.13.2	Member Function Documentation	37
6.13.2.1	close	37
6.13.2.2	fileName	37
6.13.2.3	findValue	37
6.13.2.4	open	38
6.13.2.5	read	38
6.13.2.6	write	38
6.13.3	Member Data Documentation	38
6.13.3.1	ofile	38
6.13.3.2	ofilename	38
6.13.3.3	oclock	38
6.14	BDate Class Reference	38
6.14.1	Constructor & Destructor Documentation	39
6.14.1.1	BDate	39
6.14.1.2	BDate	39
6.14.1.3	~BDate	39
6.14.2	Member Function Documentation	39
6.14.2.1	clear	39
6.14.2.2	compare	39
6.14.2.3	day	39
6.14.2.4	daysInMonth	39
6.14.2.5	getDate	39

6.14.2.6	getString	39
6.14.2.7	getStringFormatted	40
6.14.2.8	isLeap	40
6.14.2.9	isSet	40
6.14.2.10	month	40
6.14.2.11	operator BString	40
6.14.2.12	operator!=	40
6.14.2.13	operator<	40
6.14.2.14	operator<=	40
6.14.2.15	operator==	40
6.14.2.16	operator>	40
6.14.2.17	operator>=	40
6.14.2.18	set	40
6.14.2.19	set	40
6.14.2.20	setFirst	40
6.14.2.21	setLast	40
6.14.2.22	setNow	40
6.14.2.23	setString	40
6.14.2.24	setYDay	40
6.14.2.25	yday	40
6.14.2.26	year	40
6.14.3	Member Data Documentation	41
6.14.3.1	oyday	41
6.14.3.2	oyear	41
6.15	BDebugBacktrace Class Reference	41
6.15.1	Constructor & Destructor Documentation	41
6.15.1.1	BDebugBacktrace	41
6.15.1.2	~BDebugBacktrace	41
6.15.2	Member Function Documentation	41
6.15.2.1	dumpBacktrace	41
6.15.2.2	dumpBacktraceFile	41
6.15.2.3	dumpBacktraceStdout	41
6.15.2.4	dumpBacktraceSyslog	41
6.16	BDict< Type > Class Template Reference	42
6.16.1	Member Typedef Documentation	43
6.16.1.1	iterator	43
6.16.2	Constructor & Destructor Documentation	43
6.16.2.1	BDict	43
6.16.2.2	BDict	43
6.16.3	Member Function Documentation	43

6.16.3.1	append	43
6.16.3.2	append	43
6.16.3.3	clear	43
6.16.3.4	del	43
6.16.3.5	del	43
6.16.3.6	find	43
6.16.3.7	hashAdd	43
6.16.3.8	hashDelete	43
6.16.3.9	hashFind	43
6.16.3.10	hashPrint	43
6.16.3.11	hasKey	43
6.16.3.12	insert	43
6.16.3.13	key	43
6.16.3.14	operator+	43
6.16.3.15	operator=	44
6.16.3.16	operator[]	44
6.16.3.17	operator[]	44
6.16.3.18	operator[]	44
6.16.4	Member Data Documentation	44
6.16.4.1	ohashLists	44
6.16.4.2	ohashSize	44
6.17	BDictItem< Type > Class Template Reference	44
6.17.1	Detailed Description	44
6.17.2	Constructor & Destructor Documentation	44
6.17.2.1	BDictItem	44
6.17.3	Member Data Documentation	44
6.17.3.1	key	44
6.17.3.2	value	44
6.18	BDictMap< Value > Class Template Reference	45
6.18.1	Detailed Description	45
6.18.2	Member Typedef Documentation	45
6.18.2.1	iterator	45
6.18.3	Member Function Documentation	45
6.18.3.1	clear	45
6.18.3.2	del	45
6.18.3.3	del	45
6.18.3.4	hasKey	45
6.18.3.5	isEnd	45
6.18.3.6	key	46
6.18.3.7	next	46

6.18.3.8	operator[]	46
6.18.3.9	operator[]	46
6.18.3.10	size	46
6.18.3.11	start	46
6.19	BDir Class Reference	46
6.19.1	Detailed Description	47
6.19.2	Constructor & Destructor Documentation	47
6.19.2.1	BDir	47
6.19.2.2	BDir	47
6.19.2.3	~BDir	47
6.19.3	Member Function Documentation	47
6.19.3.1	clear	47
6.19.3.2	entryName	47
6.19.3.3	entryStat	47
6.19.3.4	entryStat64	47
6.19.3.5	error	47
6.19.3.6	open	47
6.19.3.7	read	47
6.19.3.8	setSort	48
6.19.3.9	setWild	48
6.19.4	Member Data Documentation	48
6.19.4.1	odirname	48
6.19.4.2	oerror	48
6.19.4.3	osort	48
6.19.4.4	owild	48
6.20	BDuration Class Reference	48
6.20.1	Constructor & Destructor Documentation	49
6.20.1.1	BDuration	49
6.20.1.2	BDuration	49
6.20.1.3	~BDuration	49
6.20.2	Member Function Documentation	49
6.20.2.1	addMicroSeconds	49
6.20.2.2	addMilliSeconds	49
6.20.2.3	addSeconds	49
6.20.2.4	clear	49
6.20.2.5	getMicroSeconds	49
6.20.2.6	getSeconds	49
6.20.2.7	getString	49
6.20.2.8	hour	50
6.20.2.9	microSecond	50

6.20.2.10 minute	50
6.20.2.11 second	50
6.20.2.12 set	50
6.20.2.13 setString	50
6.20.3 Member Data Documentation	50
6.20.3.1 ohour	50
6.20.3.2 omicroSecond	50
6.20.3.3 ominute	50
6.20.3.4 osecond	50
6.20.3.5 ospare	50
6.21 BEntry Class Reference	50
6.21.1 Detailed Description	51
6.21.2 Constructor & Destructor Documentation	51
6.21.2.1 BEntry	51
6.21.2.2 BEntry	51
6.21.2.3 BEntry	51
6.21.3 Member Function Documentation	51
6.21.3.1 getName	51
6.21.3.2 getValue	51
6.21.3.3 line	52
6.21.3.4 print	52
6.21.3.5 setLine	52
6.21.3.6 setName	52
6.21.3.7 setValue	52
6.21.4 Member Data Documentation	52
6.21.4.1 oname	52
6.21.4.2 ovalue	52
6.22 BEntryFile Class Reference	52
6.22.1 Detailed Description	53
6.22.2 Constructor & Destructor Documentation	53
6.22.2.1 BEntryFile	53
6.22.2.2 BEntryFile	53
6.22.2.3 ~BEntryFile	53
6.22.3 Member Function Documentation	53
6.22.3.1 clear	53
6.22.3.2 filename	53
6.22.3.3 open	53
6.22.3.4 read	54
6.22.3.5 write	54
6.22.3.6 writeList	54

6.22.4	Member Data Documentation	54
6.22.4.1	ocomments	54
6.22.4.2	ofilename	54
6.23	BEntryList Class Reference	54
6.23.1	Detailed Description	55
6.23.2	Constructor & Destructor Documentation	55
6.23.2.1	BEntryList	55
6.23.3	Member Function Documentation	55
6.23.3.1	clear	55
6.23.3.2	del	55
6.23.3.3	deleteEntry	55
6.23.3.4	find	55
6.23.3.5	findValue	56
6.23.3.6	getString	56
6.23.3.7	insert	56
6.23.3.8	isSet	56
6.23.3.9	operator=	56
6.23.3.10	print	56
6.23.3.11	setValue	56
6.23.3.12	setValueRaw	56
6.23.4	Member Data Documentation	56
6.23.4.1	olastPos	56
6.24	BError Class Reference	56
6.24.1	Detailed Description	57
6.24.2	Constructor & Destructor Documentation	57
6.24.2.1	BError	57
6.24.2.2	BError	58
6.24.3	Member Function Documentation	58
6.24.3.1	clear	58
6.24.3.2	copy	58
6.24.3.3	getErrorNo	58
6.24.3.4	getNumber	58
6.24.3.5	getString	58
6.24.3.6	num	58
6.24.3.7	operator int	58
6.24.3.8	set	58
6.24.3.9	setError	58
6.24.3.10	str	58
6.24.4	Member Data Documentation	59
6.24.4.1	oerrNo	59

6.24.4.2	oerrStr	59
6.25	BErrorTime Class Reference	59
6.25.1	Detailed Description	59
6.25.2	Member Enumeration Documentation	60
6.25.2.1	Type	60
6.25.3	Constructor & Destructor Documentation	60
6.25.3.1	BErrorTime	60
6.25.4	Member Function Documentation	60
6.25.4.1	clear	60
6.25.4.2	copy	60
6.25.4.3	getErrorNo	60
6.25.4.4	getString	60
6.25.4.5	getTime	60
6.25.4.6	operator int	60
6.25.4.7	set	60
6.25.5	Member Data Documentation	60
6.25.5.1	oerrNo	60
6.25.5.2	oerrStr	60
6.25.5.3	oerrTime	61
6.26	BEvent Class Reference	61
6.26.1	Constructor & Destructor Documentation	61
6.26.1.1	BEvent	61
6.26.2	Member Function Documentation	61
6.26.2.1	arg	61
6.26.2.2	type	61
6.26.3	Member Data Documentation	61
6.26.3.1	oarg	61
6.26.3.2	otype	61
6.27	BEvent1 Class Reference	62
6.27.1	Detailed Description	62
6.27.2	Constructor & Destructor Documentation	62
6.27.2.1	BEvent1	62
6.27.2.2	~BEvent1	62
6.27.3	Member Function Documentation	62
6.27.3.1	getBinary	62
6.27.3.2	getType	62
6.27.3.3	setBinary	62
6.27.4	Member Data Documentation	63
6.27.4.1	otype	63
6.28	BEvent1Error Class Reference	63

6.28.1	Constructor & Destructor Documentation	63
6.28.1.1	BEvent1Error	63
6.28.2	Member Function Documentation	63
6.28.2.1	getBinary	63
6.28.2.2	setBinary	63
6.29	BEvent1Int Class Reference	63
6.29.1	Detailed Description	64
6.29.2	Constructor & Destructor Documentation	64
6.29.2.1	BEvent1Int	64
6.29.2.2	~BEvent1Int	64
6.29.3	Member Function Documentation	64
6.29.3.1	clear	64
6.29.3.2	getEvent	64
6.29.3.3	getFd	64
6.29.3.4	sendEvent	64
6.29.4	Member Data Documentation	64
6.29.4.1	ofds	64
6.30	BEvent1Pipe Class Reference	65
6.30.1	Detailed Description	65
6.30.2	Constructor & Destructor Documentation	65
6.30.2.1	BEvent1Pipe	65
6.30.2.2	~BEvent1Pipe	65
6.30.3	Member Function Documentation	65
6.30.3.1	clear	65
6.30.3.2	getEvent	65
6.30.3.3	getReceiveFd	66
6.30.3.4	sendEvent	66
6.30.4	Member Data Documentation	66
6.30.4.1	ofds	66
6.31	BEventPipe Class Reference	66
6.31.1	Detailed Description	66
6.31.2	Constructor & Destructor Documentation	67
6.31.2.1	BEventPipe	67
6.31.2.2	~BEventPipe	67
6.31.3	Member Function Documentation	67
6.31.3.1	clear	67
6.31.3.2	getFd	67
6.31.3.3	read	67
6.31.3.4	readAvailable	67
6.31.3.5	write	67

6.31.3.6	writeAvailable	67
6.31.4	Member Data Documentation	67
6.31.4.1	ofds	67
6.32	BFifo< Type > Class Template Reference	67
6.32.1	Constructor & Destructor Documentation	69
6.32.1.1	BFifo	69
6.32.1.2	~BFifo	69
6.32.2	Member Function Documentation	69
6.32.2.1	clear	69
6.32.2.2	operator[]	69
6.32.2.3	read	69
6.32.2.4	read	69
6.32.2.5	readAvailable	69
6.32.2.6	readAvailableChunk	69
6.32.2.7	readData	69
6.32.2.8	readData	69
6.32.2.9	readDone	69
6.32.2.10	readPos	69
6.32.2.11	resize	69
6.32.2.12	size	69
6.32.2.13	write	70
6.32.2.14	write	70
6.32.2.15	writeAvailable	70
6.32.2.16	writeAvailableChunk	70
6.32.2.17	writeBackup	70
6.32.2.18	writeData	70
6.32.2.19	writeData	70
6.32.2.20	writeDone	70
6.32.3	Member Data Documentation	70
6.32.3.1	odata	70
6.32.3.2	oclock	70
6.32.3.3	oreadPos	70
6.32.3.4	osize	70
6.32.3.5	owritePos	71
6.33	BFifoCirc< Type > Class Template Reference	71
6.33.1	Detailed Description	72
6.33.2	Member Enumeration Documentation	72
6.33.2.1	anonymous enum	72
6.33.3	Constructor & Destructor Documentation	72
6.33.3.1	BFifoCirc	72

6.33.3.2	<code>~BFifoCirc</code>	72
6.33.4	Member Function Documentation	72
6.33.4.1	<code>clear</code>	72
6.33.4.2	<code>mapCircularBuffer</code>	72
6.33.4.3	<code>operator[]</code>	72
6.33.4.4	<code>read</code>	72
6.33.4.5	<code>readAvailable</code>	72
6.33.4.6	<code>readData</code>	73
6.33.4.7	<code>readDone</code>	73
6.33.4.8	<code>readWaitAvailable</code>	73
6.33.4.9	<code>size</code>	73
6.33.4.10	<code>unmapCircularBuffer</code>	73
6.33.4.11	<code>write</code>	73
6.33.4.12	<code>writeAvailable</code>	73
6.33.4.13	<code>writeData</code>	73
6.33.4.14	<code>writeDone</code>	73
6.33.4.15	<code>writeWaitAvailable</code>	73
6.33.5	Member Data Documentation	73
6.33.5.1	<code>odata</code>	73
6.33.5.2	<code>oclock</code>	73
6.33.5.3	<code>oreadPos</code>	73
6.33.5.4	<code>osize</code>	74
6.33.5.5	<code>ovmSize</code>	74
6.33.5.6	<code>owriteNumFifoSamples</code>	74
6.33.5.7	<code>owritePos</code>	74
6.34	BFifoCircPos Class Reference	74
6.34.1	Detailed Description	74
6.34.2	Constructor & Destructor Documentation	75
6.34.2.1	<code>BFifoCircPos</code>	75
6.34.3	Member Function Documentation	75
6.34.3.1	<code>difference</code>	75
6.34.3.2	<code>increment</code>	75
6.34.3.3	<code>operator int</code>	75
6.34.3.4	<code>operator!=</code>	75
6.34.3.5	<code>operator+=</code>	75
6.34.3.6	<code>operator==</code>	75
6.34.3.7	<code>pos</code>	75
6.34.3.8	<code>set</code>	75
6.34.3.9	<code>setSize</code>	75
6.34.4	Member Data Documentation	75

6.34.4.1	opos	75
6.34.4.2	osize	75
6.35	BFile Class Reference	75
6.35.1	Detailed Description	77
6.35.2	Constructor & Destructor Documentation	77
6.35.2.1	BFile	77
6.35.2.2	BFile	77
6.35.2.3	~BFile	77
6.35.3	Member Function Documentation	77
6.35.3.1	close	77
6.35.3.2	fgets	77
6.35.3.3	fileName	77
6.35.3.4	flush	77
6.35.3.5	getFd	77
6.35.3.6	isEnd	77
6.35.3.7	isOpen	77
6.35.3.8	length	77
6.35.3.9	open	78
6.35.3.10	open	78
6.35.3.11	open	78
6.35.3.12	operator=	78
6.35.3.13	position	78
6.35.3.14	printf	78
6.35.3.15	read	78
6.35.3.16	readString	78
6.35.3.17	seek	78
6.35.3.18	setVBuf	78
6.35.3.19	truncate	78
6.35.3.20	write	78
6.35.3.21	writeString	78
6.35.4	Member Data Documentation	79
6.35.4.1	ofile	79
6.35.4.2	ofilename	79
6.35.4.3	omode	79
6.36	BFileCsv Class Reference	79
6.36.1	Constructor & Destructor Documentation	79
6.36.1.1	BFileCsv	79
6.36.2	Member Function Documentation	79
6.36.2.1	readCsv	79
6.36.2.2	writeCsv	79

6.36.3	Member Data Documentation	79
6.36.3.1	oseparator	79
6.37	BFileData Class Reference	80
6.37.1	Member Function Documentation	80
6.37.1.1	del	80
6.37.1.2	find	80
6.37.1.3	getNextId	80
6.37.1.4	open	80
6.37.1.5	read	80
6.37.1.6	write	80
6.37.1.7	write	80
6.37.2	Member Data Documentation	80
6.37.2.1	ofilename	80
6.38	BIter Class Reference	81
6.38.1	Detailed Description	81
6.38.2	Constructor & Destructor Documentation	81
6.38.2.1	BIter	81
6.38.3	Member Function Documentation	81
6.38.3.1	operator BNode *	81
6.38.3.2	operator==	81
6.38.3.3	valid	81
6.38.4	Member Data Documentation	81
6.38.4.1	oi	81
6.39	BList< T > Class Template Reference	81
6.39.1	Detailed Description	84
6.39.2	Member Typedef Documentation	84
6.39.2.1	SortFunc	84
6.39.3	Constructor & Destructor Documentation	84
6.39.3.1	BList	84
6.39.3.2	BList	84
6.39.3.3	~BList	84
6.39.4	Member Function Documentation	84
6.39.4.1	append	84
6.39.4.2	append	84
6.39.4.3	begin	84
6.39.4.4	clear	84
6.39.4.5	del	84
6.39.4.6	deleteFirst	85
6.39.4.7	deleteLast	85
6.39.4.8	end	85

6.39.4.9 end	85
6.39.4.10 front	85
6.39.4.11 get	85
6.39.4.12 get	85
6.39.4.13 goTo	85
6.39.4.14 has	85
6.39.4.15 insert	85
6.39.4.16 insertAfter	85
6.39.4.17 isEnd	85
6.39.4.18 next	86
6.39.4.19 nodeCreate	86
6.39.4.20 nodeCreate	86
6.39.4.21 nodeGet	86
6.39.4.22 nodeGet	86
6.39.4.23 number	86
6.39.4.24 operator+	86
6.39.4.25 operator=	86
6.39.4.26 operator[]	86
6.39.4.27 operator[]	86
6.39.4.28 operator[]	86
6.39.4.29 operator[]	86
6.39.4.30 pop	86
6.39.4.31 position	86
6.39.4.32 prev	86
6.39.4.33 push	86
6.39.4.34 queueAdd	86
6.39.4.35 queueGet	87
6.39.4.36 rear	87
6.39.4.37 size	87
6.39.4.38 sort	87
6.39.4.39 sort	87
6.39.4.40 start	87
6.39.4.41 swap	87
6.39.5 Member Data Documentation	87
6.39.5.1 olength	87
6.39.5.2 onodes	87
6.40 BMutex Class Reference	87
6.40.1 Detailed Description	88
6.40.2 Member Enumeration Documentation	88
6.40.2.1 Type	88

6.40.3	Constructor & Destructor Documentation	88
6.40.3.1	BMutex	88
6.40.3.2	BMutex	88
6.40.3.3	~BMutex	88
6.40.4	Member Function Documentation	88
6.40.4.1	lock	88
6.40.4.2	operator=	89
6.40.4.3	timedLock	89
6.40.4.4	tryLock	89
6.40.4.5	unlock	89
6.40.5	Member Data Documentation	89
6.40.5.1	omutex	89
6.41	BMutexLock Class Reference	89
6.41.1	Constructor & Destructor Documentation	89
6.41.1.1	BMutexLock	89
6.41.1.2	~BMutexLock	89
6.41.2	Member Function Documentation	89
6.41.2.1	lock	89
6.41.2.2	unlock	89
6.41.3	Member Data Documentation	90
6.41.3.1	olock	90
6.42	BMysql Class Reference	90
6.42.1	Constructor & Destructor Documentation	90
6.42.1.1	BMysql	90
6.42.1.2	~BMysql	90
6.42.2	Member Function Documentation	90
6.42.2.1	close	90
6.42.2.2	db	90
6.42.2.3	del	90
6.42.2.4	escapeString	91
6.42.2.5	flush	91
6.42.2.6	get	91
6.42.2.7	insert	91
6.42.2.8	open	91
6.42.2.9	query	91
6.42.2.10	setDebug	91
6.42.2.11	update	91
6.42.3	Member Data Documentation	91
6.42.3.1	odb	91
6.42.3.2	odebug	91

6.42.3.3	o_lock	91
6.42.3.4	o_opened	91
6.43	BNameValue< T > Class Template Reference	91
6.43.1	Constructor & Destructor Documentation	92
6.43.1.1	BNameValue	92
6.43.1.2	BNameValue	92
6.43.2	Member Function Documentation	92
6.43.2.1	getName	92
6.43.2.2	getValue	92
6.43.3	Member Data Documentation	92
6.43.3.1	o_name	92
6.43.3.2	o_value	92
6.44	BNameValueList< T > Class Template Reference	92
6.44.1	Member Function Documentation	92
6.44.1.1	find	92
6.44.1.2	findPos	92
6.45	BNode Class Reference	93
6.45.1	Constructor & Destructor Documentation	93
6.45.1.1	BNode	93
6.45.2	Member Data Documentation	93
6.45.2.1	next	93
6.45.2.2	prev	93
6.46	BoapClientObject Class Reference	93
6.46.1	Constructor & Destructor Documentation	95
6.46.1.1	BoapClientObject	95
6.46.1.2	~BoapClientObject	95
6.46.1.3	BoapClientObject	95
6.46.2	Member Function Documentation	95
6.46.2.1	checkApiVersion	95
6.46.2.2	connectService	95
6.46.2.3	connectService	95
6.46.2.4	disconnectService	95
6.46.2.5	getServiceName	95
6.46.2.6	handleReconnect	95
6.46.2.7	performCall	95
6.46.2.8	performCall	95
6.46.2.9	performRecv	95
6.46.2.10	performRecv	95
6.46.2.11	performSend	95
6.46.2.12	performSend	95

6.46.2.13	ping	96
6.46.2.14	pingLocked	96
6.46.2.15	setConnectionPriority	96
6.46.2.16	setMaxLength	96
6.46.2.17	setTimeout	96
6.46.3	Member Data Documentation	96
6.46.3.1	oapiVersion	96
6.46.3.2	oconnected	96
6.46.3.3	oclock	96
6.46.3.4	omaxLength	96
6.46.3.5	oname	96
6.46.3.6	opriority	96
6.46.3.7	oreconnect	96
6.46.3.8	orx	96
6.46.3.9	oservice	96
6.46.3.10	otimeout	96
6.46.3.11	otx	96
6.47	Boapns::BoapEntry Class Reference	97
6.47.1	Constructor & Destructor Documentation	97
6.47.1.1	BoapEntry	97
6.47.2	Member Data Documentation	97
6.47.2.1	addressList	97
6.47.2.2	hostName	97
6.47.2.3	name	97
6.47.2.4	port	97
6.47.2.5	service	97
6.48	BoapFuncEntry Class Reference	97
6.48.1	Constructor & Destructor Documentation	98
6.48.1.1	BoapFuncEntry	98
6.48.1.2	BoapFuncEntry	98
6.48.2	Member Data Documentation	98
6.48.2.1	ocmd	98
6.48.2.2	ocmd	98
6.48.2.3	ofunc	98
6.49	BoapMcClientObject Class Reference	98
6.49.1	Constructor & Destructor Documentation	99
6.49.1.1	BoapMcClientObject	99
6.49.1.2	~BoapMcClientObject	99
6.49.2	Member Function Documentation	99
6.49.2.1	getApiVersion	99

6.49.2.2	performCall	99
6.49.2.3	performRecv	99
6.49.2.4	performSend	99
6.49.2.5	setAddress	99
6.49.3	Member Data Documentation	99
6.49.3.1	oaddressFrom	99
6.49.3.2	oaddressTo	99
6.49.3.3	oapiVersion	99
6.49.3.4	ocomms	99
6.49.3.5	opacket	99
6.50	BoapMcComms Class Reference	100
6.50.1	Constructor & Destructor Documentation	101
6.50.1.1	BoapMcComms	101
6.50.1.2	~BoapMcComms	101
6.50.2	Member Function Documentation	101
6.50.2.1	getApiVersion	101
6.50.2.2	packetRecv	101
6.50.2.3	packetSend	101
6.50.2.4	performCall	101
6.50.2.5	performSend	102
6.50.2.6	processPacket	102
6.50.2.7	processRequest	102
6.50.2.8	processRequests	102
6.50.2.9	processRx	102
6.50.2.10	setAddress	102
6.50.2.11	setComms	102
6.50.2.12	setComms	102
6.50.2.13	setCommsMode	102
6.50.2.14	setTimeout	102
6.50.3	Member Data Documentation	102
6.50.3.1	oaddressFrom	102
6.50.3.2	oaddressTo	102
6.50.3.3	oapiVersion	102
6.50.3.4	ocomms	103
6.50.3.5	oclockCall	103
6.50.3.6	oclockTx	103
6.50.3.7	opacket	103
6.50.3.8	opacketReqQueue	103
6.50.3.9	opacketReqRx	103
6.50.3.10	opacketReqTx	103

6.50.3.11	opacketRx	103
6.50.3.12	opacketRxSema	103
6.50.3.13	opacketTx	103
6.50.3.14	opacketTxQueue	103
6.50.3.15	opacketTxQueueWriteNum	103
6.50.3.16	opacketTxSema	103
6.50.3.17	oslave	104
6.50.3.18	othreaded	104
6.50.3.19	otimeout	104
6.51	BoapMcPacket Class Reference	104
6.51.1	Member Data Documentation	104
6.51.1.1	data	104
6.51.1.2	head	104
6.52	BoapMcPacketHead Struct Reference	104
6.52.1	Member Data Documentation	105
6.52.1.1	addressFrom	105
6.52.1.2	addressTo	105
6.52.1.3	checksum	105
6.52.1.4	cmd	105
6.52.1.5	error	105
6.52.1.6	length	105
6.53	BoapMcServiceObject Class Reference	105
6.53.1	Constructor & Destructor Documentation	105
6.53.1.1	BoapMcServiceObject	105
6.53.1.2	~BoapMcServiceObject	105
6.53.2	Member Function Documentation	105
6.53.2.1	process	105
6.53.2.2	processEvent	105
6.53.2.3	sendEvent	105
6.53.3	Member Data Documentation	106
6.53.3.1	oapiVersion	106
6.54	BoapMcSignalObject Class Reference	106
6.54.1	Constructor & Destructor Documentation	106
6.54.1.1	BoapMcSignalObject	106
6.54.2	Member Function Documentation	106
6.54.2.1	performSend	106
6.54.3	Member Data Documentation	106
6.54.3.1	ocomms	106
6.55	Boapns::Boapns Class Reference	106
6.55.1	Constructor & Destructor Documentation	107

6.55.1.1	Boapns	107
6.55.2	Member Function Documentation	107
6.55.2.1	addEntry	107
6.55.2.2	delEntry	107
6.55.2.3	getEntry	107
6.55.2.4	getEntryList	107
6.55.2.5	getNewName	107
6.55.2.6	getVersion	107
6.56	BoapPacket Class Reference	107
6.56.1	Constructor & Destructor Documentation	109
6.56.1.1	BoapPacket	109
6.56.1.2	~BoapPacket	109
6.56.1.3	BoapPacket	109
6.56.1.4	~BoapPacket	109
6.56.2	Member Function Documentation	109
6.56.2.1	data	109
6.56.2.2	getCmd	109
6.56.2.3	nbytes	109
6.56.2.4	peekHead	109
6.56.2.5	pop	109
6.56.2.6	pop	109
6.56.2.7	pop	109
6.56.2.8	pop	109
6.56.2.9	pop	109
6.56.2.10	pop	109
6.56.2.11	pop	109
6.56.2.12	pop	109
6.56.2.13	pop	109
6.56.2.14	pop	109
6.56.2.15	popHead	109
6.56.2.16	popHead	109
6.56.2.17	push	109
6.56.2.18	push	110
6.56.2.19	push	110
6.56.2.20	push	110
6.56.2.21	push	110
6.56.2.22	push	110
6.56.2.23	push	110
6.56.2.24	push	110
6.56.2.25	push	110

6.56.2.26	push	110
6.56.2.27	pushHead	110
6.56.2.28	pushHead	110
6.56.2.29	resize	110
6.56.2.30	setData	110
6.56.2.31	updateHead	110
6.56.2.32	updateLen	110
6.56.3	Member Data Documentation	110
6.56.3.1	odata	110
6.56.3.2	onbytes	110
6.56.3.3	opos	110
6.56.3.4	osize	110
6.57	BoapPacketHead Struct Reference	110
6.57.1	Member Data Documentation	111
6.57.1.1	cmd	111
6.57.1.2	cmd	111
6.57.1.3	length	111
6.57.1.4	length	111
6.57.1.5	reserved	111
6.57.1.6	service	111
6.57.1.7	service	111
6.57.1.8	type	111
6.57.1.9	type	111
6.58	BoapServer Class Reference	111
6.58.1	Member Enumeration Documentation	113
6.58.1.1	anonymous enum	113
6.58.2	Constructor & Destructor Documentation	113
6.58.2.1	BoapServer	113
6.58.2.2	~BoapServer	113
6.58.2.3	BoapServer	113
6.58.3	Member Function Documentation	113
6.58.3.1	addObject	113
6.58.3.2	addObject	113
6.58.3.3	clientGone	113
6.58.3.4	function	113
6.58.3.5	getConnectionsNumber	113
6.58.3.6	getEventSocket	113
6.58.3.7	getEventSocket	113
6.58.3.8	getHostName	113
6.58.3.9	getHostName	113

6.58.3.10	getSocket	113
6.58.3.11	getSocket	113
6.58.3.12	init	113
6.58.3.13	init	113
6.58.3.14	newConnection	113
6.58.3.15	process	113
6.58.3.16	process	113
6.58.3.17	processEvent	114
6.58.3.18	processEvent	114
6.58.3.19	processEvent	114
6.58.3.20	processEvent	114
6.58.3.21	run	114
6.58.3.22	run	114
6.58.3.23	sendEvent	114
6.58.3.24	sendEvent	114
6.58.4	Member Data Documentation	114
6.58.4.1	oboapNs	114
6.58.4.2	oboapns	114
6.58.4.3	oclientGoneEvent	114
6.58.4.4	oclients	114
6.58.4.5	ohostName	114
6.58.4.6	oisBoapns	114
6.58.4.7	onet	114
6.58.4.8	onetEvent	114
6.58.4.9	onetEventAddress	114
6.58.4.10	onumOperations	114
6.58.4.11	opoll	114
6.58.4.12	orx	114
6.58.4.13	oservices	114
6.58.4.14	othreaded	114
6.58.4.15	otx	114
6.59	BoapServerConnection Class Reference	115
6.59.1	Constructor & Destructor Documentation	115
6.59.1.1	BoapServerConnection	115
6.59.1.2	~BoapServerConnection	115
6.59.2	Member Function Documentation	115
6.59.2.1	function	115
6.59.2.2	getHead	116
6.59.2.3	getSocket	116
6.59.2.4	init	116

6.59.2.5	process	116
6.59.2.6	setMaxLength	116
6.59.2.7	validate	116
6.59.3	Member Data Documentation	116
6.59.3.1	oboapServer	116
6.59.3.2	omaxLength	116
6.59.3.3	orx	116
6.59.3.4	osocket	116
6.59.3.5	otx	116
6.60	BoapServiceEntry Class Reference	116
6.60.1	Constructor & Destructor Documentation	116
6.60.1.1	BoapServiceEntry	116
6.60.1.2	BoapServiceEntry	117
6.60.2	Member Data Documentation	117
6.60.2.1	oobject	117
6.60.2.2	oservice	117
6.61	BoapServiceObject Class Reference	117
6.61.1	Constructor & Destructor Documentation	118
6.61.1.1	BoapServiceObject	118
6.61.1.2	~BoapServiceObject	118
6.61.1.3	BoapServiceObject	118
6.61.1.4	~BoapServiceObject	118
6.61.2	Member Function Documentation	118
6.61.2.1	doConnectionPriority	118
6.61.2.2	doPing	118
6.61.2.3	name	118
6.61.2.4	name	118
6.61.2.5	process	118
6.61.2.6	process	118
6.61.2.7	processEvent	118
6.61.2.8	processEvent	118
6.61.2.9	processEvent	118
6.61.2.10	processEvent	118
6.61.2.11	sendEvent	118
6.61.2.12	sendEvent	118
6.61.2.13	sendEvent	118
6.61.2.14	sendEvent	118
6.61.2.15	setName	118
6.61.3	Member Data Documentation	118
6.61.3.1	oapiVersion	118

6.61.3.2	ofuncList	118
6.61.3.3	oname	118
6.61.3.4	oserver	118
6.62	BoapSignalObject Class Reference	119
6.62.1	Constructor & Destructor Documentation	119
6.62.1.1	BoapSignalObject	119
6.62.1.2	BoapSignalObject	119
6.62.2	Member Function Documentation	119
6.62.2.1	performSend	119
6.62.2.2	performSend	119
6.62.3	Member Data Documentation	119
6.62.3.1	orx	119
6.62.3.2	otx	120
6.63	BObj Class Reference	120
6.63.1	Constructor & Destructor Documentation	120
6.63.1.1	BObj	120
6.63.1.2	~BObj	120
6.63.2	Member Function Documentation	120
6.63.2.1	getDebugString	120
6.63.2.2	getMember	120
6.63.2.3	getMembers	120
6.63.2.4	getMembers	120
6.63.2.5	getType	120
6.63.2.6	membersPrint	120
6.63.2.7	setMember	121
6.63.2.8	setMembers	121
6.64	BObjMember Struct Reference	121
6.64.1	Member Data Documentation	121
6.64.1.1	dataOffset	121
6.64.1.2	name	121
6.64.1.3	size	121
6.64.1.4	type	121
6.64.1.5	typeComp	121
6.64.1.6	typeName	121
6.65	BPoll Class Reference	121
6.65.1	Detailed Description	122
6.65.2	Member Typedef Documentation	122
6.65.2.1	PollFd	122
6.65.3	Constructor & Destructor Documentation	122
6.65.3.1	BPoll	122

6.65.3.2	<code>~BPoll</code>	122
6.65.4	Member Function Documentation	122
6.65.4.1	<code>append</code>	122
6.65.4.2	<code>clear</code>	123
6.65.4.3	<code>delFd</code>	123
6.65.4.4	<code>doPoll</code>	123
6.65.4.5	<code>doPollEvents</code>	123
6.65.4.6	<code>getPollFds</code>	123
6.65.4.7	<code>getPollFdsNum</code>	123
6.65.4.8	<code>nextFd</code>	123
6.65.5	Member Data Documentation	123
6.65.5.1	<code>ofds</code>	123
6.65.5.2	<code>ofdsNext</code>	123
6.65.5.3	<code>ofdsNum</code>	123
6.66	<code>BQueue< T ></code> Class Template Reference	123
6.66.1	Detailed Description	124
6.66.2	Constructor & Destructor Documentation	124
6.66.2.1	<code>BQueue</code>	124
6.66.2.2	<code>~BQueue</code>	124
6.66.3	Member Function Documentation	124
6.66.3.1	<code>clear</code>	124
6.66.3.2	<code>read</code>	124
6.66.3.3	<code>readAvailable</code>	124
6.66.3.4	<code>write</code>	124
6.66.3.5	<code>writeAvailable</code>	125
6.66.4	Member Data Documentation	125
6.66.4.1	<code>olock</code>	125
6.66.4.2	<code>onumber</code>	125
6.66.4.3	<code>osize</code>	125
6.67	<code>BRefData</code> Class Reference	125
6.67.1	Detailed Description	125
6.67.2	Constructor & Destructor Documentation	126
6.67.2.1	<code>BRefData</code>	126
6.67.2.2	<code>BRefData</code>	126
6.67.2.3	<code>BRefData</code>	126
6.67.2.4	<code>~BRefData</code>	126
6.67.3	Member Function Documentation	126
6.67.3.1	<code>addRef</code>	126
6.67.3.2	<code>copy</code>	126
6.67.3.3	<code>data</code>	126

6.67.3.4	deleteRef	126
6.67.3.5	len	126
6.67.3.6	operator=	126
6.67.3.7	setLen	126
6.67.4	Member Data Documentation	126
6.67.4.1	odata	126
6.67.4.2	olen	126
6.67.4.3	orefCount	127
6.68	BRtc Class Reference	127
6.68.1	Detailed Description	127
6.68.2	Constructor & Destructor Documentation	127
6.68.2.1	BRtc	127
6.68.2.2	~BRtc	127
6.68.3	Member Function Documentation	127
6.68.3.1	init	127
6.68.3.2	wait	127
6.68.4	Member Data Documentation	128
6.68.4.1	ofd	128
6.68.4.2	orate	128
6.69	BRtcThreaded Class Reference	128
6.69.1	Detailed Description	128
6.69.2	Constructor & Destructor Documentation	129
6.69.2.1	BRtcThreaded	129
6.69.2.2	~BRtcThreaded	129
6.69.3	Member Function Documentation	129
6.69.3.1	function	129
6.69.3.2	init	129
6.69.3.3	wait	129
6.69.4	Member Data Documentation	129
6.69.4.1	ocond	129
6.69.4.2	orate	129
6.69.4.3	ortc	129
6.70	BRWLock Class Reference	129
6.70.1	Detailed Description	130
6.70.2	Constructor & Destructor Documentation	130
6.70.2.1	BRWLock	130
6.70.2.2	BRWLock	130
6.70.2.3	~BRWLock	130
6.70.3	Member Function Documentation	130
6.70.3.1	operator=	130

6.70.3.2	rdLock	130
6.70.3.3	tryRdLock	130
6.70.3.4	tryWrLock	130
6.70.3.5	unlock	130
6.70.3.6	wrLock	130
6.70.4	Member Data Documentation	130
6.70.4.1	oLock	130
6.71	BSema Class Reference	131
6.71.1	Detailed Description	131
6.71.2	Constructor & Destructor Documentation	131
6.71.2.1	BSema	131
6.71.2.2	BSema	131
6.71.2.3	~BSema	131
6.71.3	Member Function Documentation	131
6.71.3.1	getValue	131
6.71.3.2	operator=	131
6.71.3.3	post	131
6.71.3.4	timedWait	132
6.71.3.5	tryWait	132
6.71.3.6	wait	132
6.71.4	Member Data Documentation	132
6.71.4.1	osema	132
6.72	BSemaphore Class Reference	132
6.72.1	Detailed Description	132
6.72.2	Constructor & Destructor Documentation	133
6.72.2.1	BSemaphore	133
6.72.2.2	BSemaphore	133
6.72.2.3	~BSemaphore	133
6.72.3	Member Function Documentation	133
6.72.3.1	getValue	133
6.72.3.2	operator=	133
6.72.3.3	set	133
6.72.3.4	wait	133
6.72.4	Member Data Documentation	133
6.72.4.1	osema	133
6.73	BSemaphoreCount Class Reference	133
6.73.1	Constructor & Destructor Documentation	134
6.73.1.1	BSemaphoreCount	134
6.73.1.2	BSemaphoreCount	134
6.73.1.3	~BSemaphoreCount	134

6.73.2	Member Function Documentation	134
6.73.2.1	add	134
6.73.2.2	operator=	134
6.73.2.3	setValue	134
6.73.2.4	take	134
6.73.2.5	value	134
6.73.2.6	wait	134
6.73.3	Member Data Documentation	134
6.73.3.1	olock	134
6.73.3.2	osema	134
6.73.3.3	ovalue	134
6.74	BSocket Class Reference	134
6.74.1	Member Enumeration Documentation	136
6.74.1.1	NType	136
6.74.1.2	Priority	136
6.74.2	Constructor & Destructor Documentation	136
6.74.2.1	BSocket	136
6.74.2.2	BSocket	136
6.74.2.3	BSocket	136
6.74.2.4	BSocket	136
6.74.2.5	~BSocket	136
6.74.3	Member Function Documentation	136
6.74.3.1	accept	136
6.74.3.2	accept	136
6.74.3.3	bind	136
6.74.3.4	close	136
6.74.3.5	connect	136
6.74.3.6	getAddress	136
6.74.3.7	getFd	136
6.74.3.8	getMTU	136
6.74.3.9	getSockOpt	136
6.74.3.10	init	136
6.74.3.11	init	136
6.74.3.12	listen	136
6.74.3.13	recv	137
6.74.3.14	recvFrom	137
6.74.3.15	recvFromWithTimeout	137
6.74.3.16	recvWithTimeout	137
6.74.3.17	send	137
6.74.3.18	sendTo	137

6.74.3.19	setBroadCast	137
6.74.3.20	setFd	137
6.74.3.21	setPriority	137
6.74.3.22	setReuseAddress	137
6.74.3.23	setSockOpt	137
6.74.3.24	shutdown	137
6.74.4	Member Data Documentation	137
6.74.4.1	osocket	137
6.75	BSocketAddress Class Reference	137
6.75.1	Detailed Description	138
6.75.2	Member Typedef Documentation	138
6.75.2.1	SockAddr	138
6.75.3	Constructor & Destructor Documentation	138
6.75.3.1	BSocketAddress	138
6.75.3.2	BSocketAddress	138
6.75.3.3	BSocketAddress	138
6.75.3.4	~BSocketAddress	138
6.75.4	Member Function Documentation	138
6.75.4.1	len	138
6.75.4.2	operator const SockAddr *	138
6.75.4.3	operator!=	138
6.75.4.4	operator=	138
6.75.4.5	operator==	138
6.75.4.6	raw	138
6.75.4.7	set	138
6.75.5	Member Data Documentation	139
6.75.5.1	oaddress	139
6.75.5.2	olen	139
6.76	BSocketAddressINET Class Reference	139
6.76.1	Detailed Description	140
6.76.2	Member Typedef Documentation	140
6.76.2.1	SockAddrIP	140
6.76.3	Member Function Documentation	140
6.76.3.1	address	140
6.76.3.2	getHostName	140
6.76.3.3	getIpAddresses	140
6.76.3.4	getIpAddressList	140
6.76.3.5	getIpAddressListAll	140
6.76.3.6	getString	140
6.76.3.7	port	140

6.76.3.8	set	140
6.76.3.9	set	140
6.76.3.10	set	140
6.76.3.11	setPort	140
6.77	BSpi Class Reference	141
6.77.1	Detailed Description	141
6.77.2	Member Enumeration Documentation	141
6.77.2.1	Mode	141
6.77.3	Constructor & Destructor Documentation	141
6.77.3.1	BSpi	141
6.77.4	Member Function Documentation	141
6.77.4.1	init	141
6.77.4.2	transact	141
6.77.5	Member Data Documentation	141
6.77.5.1	odev	141
6.77.5.2	odevName	141
6.78	BString Class Reference	142
6.78.1	Constructor & Destructor Documentation	145
6.78.1.1	BString	145
6.78.1.2	BString	145
6.78.1.3	BString	145
6.78.1.4	BString	145
6.78.1.5	BString	145
6.78.1.6	BString	145
6.78.1.7	BString	145
6.78.1.8	BString	145
6.78.1.9	BString	145
6.78.1.10	~BString	145
6.78.2	Member Function Documentation	145
6.78.2.1	add	145
6.78.2.2	append	145
6.78.2.3	base64Decode	145
6.78.2.4	base64Encode	145
6.78.2.5	basename	146
6.78.2.6	clear	146
6.78.2.7	compare	146
6.78.2.8	compareRegex	146
6.78.2.9	compareWild	146
6.78.2.10	compareWildExpression	146
6.78.2.11	convert	146

6.78.2.12 convert	146
6.78.2.13 convert	146
6.78.2.14 convert	146
6.78.2.15 convert	146
6.78.2.16 convertHex	146
6.78.2.17 convertHex	146
6.78.2.18 copy	147
6.78.2.19 csvDecode	147
6.78.2.20 csvEncode	147
6.78.2.21 del	147
6.78.2.22 dirname	147
6.78.2.23 extension	147
6.78.2.24 field	147
6.78.2.25 fields	147
6.78.2.26 find	147
6.78.2.27 find	147
6.78.2.28 findReverse	147
6.78.2.29 firstLine	147
6.78.2.30 fixedLen	147
6.78.2.31 get	147
6.78.2.32 get	147
6.78.2.33 getTokenList	147
6.78.2.34 getTokenList	148
6.78.2.35 hash	148
6.78.2.36 init	148
6.78.2.37 insert	148
6.78.2.38 inString	148
6.78.2.39 isSpace	148
6.78.2.40 justify	148
6.78.2.41 len	148
6.78.2.42 lowerFirst	148
6.78.2.43 operator const char *	148
6.78.2.44 operator!=	148
6.78.2.45 operator!=	148
6.78.2.46 operator+	148
6.78.2.47 operator+	148
6.78.2.48 operator+	148
6.78.2.49 operator+	148
6.78.2.50 operator+	148
6.78.2.51 operator+	148

6.78.2.52 operator+=	148
6.78.2.53 operator+=	148
6.78.2.54 operator<	148
6.78.2.55 operator<	148
6.78.2.56 operator<=	149
6.78.2.57 operator=	149
6.78.2.58 operator==	149
6.78.2.59 operator==	149
6.78.2.60 operator>	149
6.78.2.61 operator>	149
6.78.2.62 operator>=	149
6.78.2.63 operator[]	149
6.78.2.64 pad	149
6.78.2.65 printf	149
6.78.2.66 pullLine	149
6.78.2.67 pullSeparators	149
6.78.2.68 pullToken	149
6.78.2.69 pullWord	149
6.78.2.70 removeNL	149
6.78.2.71 removeSeparators	149
6.78.2.72 retDouble	149
6.78.2.73 retInt	150
6.78.2.74 retStr	150
6.78.2.75 retStrDup	150
6.78.2.76 retUInt	150
6.78.2.77 reverse	150
6.78.2.78 split	150
6.78.2.79 subString	150
6.78.2.80 toLower	150
6.78.2.81 toUpper	150
6.78.2.82 translateChar	150
6.78.2.83 truncate	150
6.78.3 Member Data Documentation	150
6.78.3.1 ostr	150
6.79 BStringLocked Class Reference	151
6.79.1 Constructor & Destructor Documentation	151
6.79.1.1 BStringLocked	151
6.79.1.2 BStringLocked	151
6.79.1.3 BStringLocked	151
6.79.2 Member Function Documentation	151

6.79.2.1	len	151
6.79.2.2	operator BString	151
6.79.2.3	operator+	151
6.79.2.4	operator=	151
6.79.3	Member Data Documentation	151
6.79.3.1	oLock	151
6.79.3.2	ostr	151
6.80	BStringMutex Class Reference	152
6.80.1	Constructor & Destructor Documentation	152
6.80.1.1	BStringMutex	152
6.81	BTable Class Reference	152
6.81.1	Constructor & Destructor Documentation	153
6.81.1.1	BTable	153
6.81.1.2	~BTable	153
6.81.2	Member Function Documentation	153
6.81.2.1	addRow	153
6.81.2.2	calculateWidths	153
6.81.2.3	clear	153
6.81.2.4	print	153
6.81.2.5	printLine	153
6.81.2.6	setTitle	153
6.81.3	Member Data Documentation	153
6.81.3.1	oColumnWidths	153
6.81.3.2	odata	153
6.81.3.3	otitle	153
6.82	BThread Class Reference	153
6.82.1	Constructor & Destructor Documentation	154
6.82.1.1	BThread	154
6.82.1.2	~BThread	154
6.82.2	Member Function Documentation	154
6.82.2.1	cancel	154
6.82.2.2	function	154
6.82.2.3	getThread	154
6.82.2.4	result	154
6.82.2.5	running	154
6.82.2.6	setInitPriority	154
6.82.2.7	setInitStackSize	154
6.82.2.8	setPriority	154
6.82.2.9	start	154
6.82.2.10	startFunc	154

6.82.2.11	waitForCompletion	154
6.82.3	Member Data Documentation	154
6.82.3.1	opolicy	155
6.82.3.2	opriority	155
6.82.3.3	oreult	155
6.82.3.4	orunning	155
6.82.3.5	ostackSize	155
6.82.3.6	othread	155
6.83	BTime Class Reference	155
6.83.1	Constructor & Destructor Documentation	156
6.83.1.1	BTime	156
6.83.2	Member Function Documentation	156
6.83.2.1	addSeconds	156
6.83.2.2	getDate	156
6.83.2.3	getSeconds	156
6.83.2.4	getString	156
6.83.2.5	getTime	156
6.83.2.6	isLeapYear	156
6.83.2.7	isSet	156
6.83.2.8	operator!=	156
6.83.2.9	operator+	156
6.83.2.10	operator+=	156
6.83.2.11	operator<	156
6.83.2.12	operator<=	156
6.83.2.13	operator==	156
6.83.2.14	operator>	157
6.83.2.15	operator>=	157
6.83.2.16	set	157
6.83.2.17	set	157
6.83.2.18	setString	157
6.83.2.19	setYearDay	157
6.83.3	Member Data Documentation	157
6.83.3.1	otime	157
6.84	BTimer Class Reference	157
6.84.1	Detailed Description	158
6.84.2	Constructor & Destructor Documentation	158
6.84.2.1	BTimer	158
6.84.2.2	~BTimer	158
6.84.3	Member Function Documentation	158
6.84.3.1	add	158

6.84.3.2	average	158
6.84.3.3	clear	158
6.84.3.4	getElapsedTime	158
6.84.3.5	getTime	158
6.84.3.6	peak	158
6.84.3.7	start	159
6.84.3.8	stop	159
6.84.4	Member Data Documentation	159
6.84.4.1	oaverage	159
6.84.4.2	oendTime	159
6.84.4.3	oclock	159
6.84.4.4	onum	159
6.84.4.5	opeak	159
6.84.4.6	ostartTime	159
6.85	BTimeStamp Class Reference	159
6.85.1	Constructor & Destructor Documentation	161
6.85.1.1	BTimeStamp	161
6.85.1.2	BTimeStamp	161
6.85.1.3	BTimeStamp	161
6.85.1.4	~BTimeStamp	161
6.85.2	Member Function Documentation	161
6.85.2.1	addMicroSeconds	161
6.85.2.2	addMilliSeconds	161
6.85.2.3	addSeconds	161
6.85.2.4	clear	161
6.85.2.5	compare	161
6.85.2.6	day	162
6.85.2.7	difference	162
6.85.2.8	getDate	162
6.85.2.9	getString	162
6.85.2.10	getStringFormatted	162
6.85.2.11	getStringNoMs	162
6.85.2.12	getYearMicroSeconds	162
6.85.2.13	getYearSeconds	162
6.85.2.14	hour	162
6.85.2.15	isLeap	162
6.85.2.16	isSet	162
6.85.2.17	microSecond	162
6.85.2.18	minute	162
6.85.2.19	month	162

6.85.2.20 operator BString	162
6.85.2.21 operator!=	162
6.85.2.22 operator<	162
6.85.2.23 operator<=	162
6.85.2.24 operator=	162
6.85.2.25 operator==	162
6.85.2.26 operator>	162
6.85.2.27 operator>=	162
6.85.2.28 second	163
6.85.2.29 set	163
6.85.2.30 set	163
6.85.2.31 set	163
6.85.2.32 setFirst	163
6.85.2.33 setLast	163
6.85.2.34 setNow	163
6.85.2.35 setString	163
6.85.2.36 setTime	163
6.85.2.37 setYDay	163
6.85.2.38 yday	163
6.85.2.39 year	163
6.85.3 Member Data Documentation	163
6.85.3.1 ohour	163
6.85.3.2 omicroSecond	163
6.85.3.3 ominute	163
6.85.3.4 osecond	164
6.85.3.5 ospare	164
6.85.3.6 oyday	164
6.85.3.7 oyear	164
6.86 BTimeStampMs Class Reference	164
6.86.1 Constructor & Destructor Documentation	165
6.86.1.1 BTimeStampMs	165
6.86.1.2 ~BTimeStampMs	165
6.86.2 Member Function Documentation	165
6.86.2.1 addMilliseconds	165
6.86.2.2 addSeconds	166
6.86.2.3 clear	166
6.86.2.4 compare	166
6.86.2.5 difference	166
6.86.2.6 getDate	166
6.86.2.7 getDurationString	166

6.86.2.8	getDurationStringNoMs	166
6.86.2.9	getString	166
6.86.2.10	getStringNoMs	166
6.86.2.11	getStringRaw	166
6.86.2.12	getYearMilliseconds	166
6.86.2.13	getYearSeconds	166
6.86.2.14	isLeap	166
6.86.2.15	operator<	166
6.86.2.16	operator<=	166
6.86.2.17	operator>	167
6.86.2.18	operator>=	167
6.86.2.19	setDurationString	167
6.86.2.20	setNow	167
6.86.2.21	setString	167
6.86.2.22	subMilliseconds	167
6.86.2.23	subSeconds	167
6.86.3	Member Data Documentation	167
6.86.3.1	hour	167
6.86.3.2	milliSecond	167
6.86.3.3	minute	167
6.86.3.4	sampleNumber	167
6.86.3.5	second	167
6.86.3.6	yday	167
6.86.3.7	year	168
6.87	BUrl Class Reference	168
6.87.1	Detailed Description	168
6.87.2	Constructor & Destructor Documentation	168
6.87.2.1	BUrl	168
6.87.2.2	~BUrl	168
6.87.3	Member Function Documentation	168
6.87.3.1	readString	168
6.87.3.2	writeData	169
6.87.4	Member Data Documentation	169
6.87.4.1	oinit	169
6.87.4.2	ores	169
6.88	BList< T >::Node Class Reference	169
6.88.1	Constructor & Destructor Documentation	169
6.88.1.1	Node	169
6.88.2	Member Data Documentation	169
6.88.2.1	item	169

7 File Documentation	171
7.1 BArray.h File Reference	171
7.1.1 Macro Definition Documentation	171
7.1.1.1 BArrayLoop	171
7.2 BAtomic.h File Reference	171
7.2.1 Typedef Documentation	172
7.2.1.1 BAtomicInt32	172
7.2.1.2 BAtomicInt64	172
7.2.1.3 BAtomicUInt32	172
7.2.1.4 BAtomicUInt64	172
7.3 BAtomicCount.h File Reference	172
7.4 BBuffer.cpp File Reference	172
7.4.1 Variable Documentation	172
7.4.1.1 roundSize	172
7.5 BBuffer.h File Reference	172
7.5.1 Macro Definition Documentation	173
7.5.1.1 BBigEndian	173
7.6 BComms.cpp File Reference	173
7.7 BComms.h File Reference	173
7.8 BComplex.h File Reference	173
7.8.1 Typedef Documentation	173
7.8.1.1 BComplex	173
7.8.1.2 BComplex32	173
7.8.1.3 BComplex64	173
7.9 BCond.cpp File Reference	174
7.10 BCond.h File Reference	174
7.11 BCondInt.cpp File Reference	174
7.11.1 Function Documentation	174
7.11.1.1 getTimeout	174
7.12 BCondInt.h File Reference	174
7.13 BConfig.cpp File Reference	175
7.14 BConfig.h File Reference	175
7.15 BCrc16.cpp File Reference	175
7.15.1 Function Documentation	175
7.15.1.1 bcrc16	175
7.15.2 Variable Documentation	175
7.15.2.1 table_crc_hi	175
7.15.2.2 table_crc_lo	176
7.16 BCrc16.h File Reference	176
7.16.1 Function Documentation	176

7.16.1.1	brcrc16	176
7.17	BDate.cpp File Reference	176
7.17.1	Function Documentation	177
7.17.1.1	fromBString	177
7.17.1.2	toBString	177
7.17.2	Variable Documentation	177
7.17.2.1	mon_yday	177
7.18	BDate.h File Reference	177
7.18.1	Function Documentation	177
7.18.1.1	fromBString	177
7.18.1.2	toBString	177
7.19	BDebug.cpp File Reference	178
7.19.1	Macro Definition Documentation	178
7.19.1.1	BTRACE_SIZE	178
7.19.2	Function Documentation	178
7.19.2.1	gettid	178
7.19.2.2	getTime	178
7.19.2.3	hd32	178
7.19.2.4	hd8	178
7.19.2.5	hd8a	178
7.19.2.6	hda32	179
7.19.2.7	hda8	179
7.19.2.8	setDebug	179
7.19.2.9	tprintf	179
7.19.3	Variable Documentation	179
7.19.3.1	bdebug	179
7.19.3.2	STRBUF_SIZE	179
7.20	BDebug.h File Reference	179
7.20.1	Macro Definition Documentation	180
7.20.1.1	BDebug_STD	180
7.20.1.2	dprintf	180
7.20.1.3	eprintf	180
7.20.1.4	nprintf	180
7.20.1.5	wprintf	180
7.20.2	Function Documentation	180
7.20.2.1	gettid	180
7.20.2.2	getTime	180
7.20.2.3	hd32	180
7.20.2.4	hd8	180
7.20.2.5	hd8a	180

7.20.2.6	hda8	180
7.20.2.7	hds32	180
7.20.2.8	setDebug	180
7.20.2.9	tprintf	180
7.20.3	Variable Documentation	180
7.20.3.1	bdebug	180
7.21	BDict.cpp File Reference	180
7.21.1	Function Documentation	181
7.21.1.1	bdictStringToString	181
7.21.1.2	fromBString	181
7.21.1.3	toBString	181
7.22	BDict.h File Reference	181
7.22.1	Typedef Documentation	181
7.22.1.1	BDictString	181
7.22.2	Function Documentation	181
7.22.2.1	bdictStringToString	181
7.22.2.2	fromBString	181
7.22.2.3	toBString	181
7.23	BDictMap.h File Reference	181
7.23.1	Typedef Documentation	182
7.23.1.1	BDictMapString	182
7.24	BDir.cpp File Reference	182
7.24.1	Function Documentation	182
7.24.1.1	wild	182
7.24.2	Variable Documentation	182
7.24.2.1	wildString	182
7.25	BDir.h File Reference	182
7.26	BDuration.cpp File Reference	182
7.27	BDuration.h File Reference	183
7.28	BEndian.cpp File Reference	183
7.28.1	Function Documentation	183
7.28.1.1	bswap_copy	183
7.29	BEndian.h File Reference	183
7.29.1	Macro Definition Documentation	184
7.29.1.1	be16toh	184
7.29.1.2	be32toh	184
7.29.1.3	be64toh	184
7.29.1.4	htobe16	184
7.29.1.5	htobe32	184
7.29.1.6	htobe64	184

7.29.1.7	htole16	184
7.29.1.8	htole32	184
7.29.1.9	htole64	184
7.29.1.10	le16toh	185
7.29.1.11	le32toh	185
7.29.1.12	le64toh	185
7.29.2	Function Documentation	185
7.29.2.1	betoh	185
7.29.2.2	betoh	185
7.29.2.3	betoh	185
7.29.2.4	betoh	185
7.29.2.5	betoh	185
7.29.2.6	betoh	185
7.29.2.7	betoh	185
7.29.2.8	betoh	185
7.29.2.9	bswap_copy	185
7.29.2.10	bswap_p16	185
7.29.2.11	bswap_p32	185
7.29.2.12	bswap_p64	185
7.29.2.13	bswap_p8	185
7.29.2.14	htobe	185
7.29.2.15	htobe	185
7.29.2.16	htobe	185
7.29.2.17	htobe	185
7.29.2.18	htobe	185
7.29.2.19	htobe	185
7.29.2.20	htobe	185
7.29.2.21	htobe	185
7.29.2.22	htole	185
7.29.2.23	htole	185
7.29.2.24	htole	185
7.29.2.25	htole	186
7.29.2.26	htole	186
7.29.2.27	htole	186
7.29.2.28	htole	186
7.29.2.29	htole	186
7.29.2.30	letoh	186
7.29.2.31	letoh	186
7.29.2.32	letoh	186
7.29.2.33	letoh	186

7.29.2.34 letoh	186
7.29.2.35 letoh	186
7.29.2.36 letoh	186
7.29.2.37 letoh	186
7.30 BEntry.cpp File Reference	186
7.31 BEntry.h File Reference	186
7.32 BError.cpp File Reference	187
7.33 BError.h File Reference	187
7.33.1 Enumeration Type Documentation	187
7.33.1.1 BErrorNum	187
7.34 BErrorTime.cpp File Reference	188
7.35 BErrorTime.h File Reference	188
7.36 BEvent.cpp File Reference	188
7.37 BEvent.h File Reference	188
7.37.1 Typedef Documentation	189
7.37.1.1 BEventQueue	189
7.37.2 Enumeration Type Documentation	189
7.37.2.1 BEventType	189
7.38 BEvent1.cpp File Reference	189
7.39 BEvent1.h File Reference	189
7.39.1 Enumeration Type Documentation	190
7.39.1.1 BEvent1Type	190
7.40 BFifo.h File Reference	190
7.41 BFifo.inc File Reference	190
7.42 BFifoCirc.cpp File Reference	190
7.42.1 Macro Definition Documentation	190
7.42.1.1 dprintf	190
7.43 BFifoCirc.h File Reference	190
7.44 BFifoCirc.inc File Reference	191
7.45 BFile.cpp File Reference	191
7.45.1 Macro Definition Documentation	191
7.45.1.1 STRBUF	191
7.46 BFile.h File Reference	191
7.47 BFileCsv.cpp File Reference	191
7.48 BFileCsv.h File Reference	192
7.49 BFileData.cpp File Reference	192
7.50 BFileData.h File Reference	192
7.51 BList.h File Reference	192
7.51.1 Macro Definition Documentation	193
7.51.1.1 BListLoop	193

7.52 BList_func.h File Reference	193
7.53 BMutex.cpp File Reference	193
7.53.1 Macro Definition Documentation	193
7.53.1.1 MDEBUG	193
7.54 BMutex.h File Reference	193
7.55 BMySQL.cpp File Reference	193
7.56 BMySQL.h File Reference	193
7.57 BNameValue.h File Reference	194
7.58 Boap.cpp File Reference	194
7.58.1 Macro Definition Documentation	194
7.58.1.1 APIVERSION_TEST	194
7.58.1.2 DEBUG	194
7.58.1.3 dprintf	195
7.58.1.4 IS_BIG_ENDIAN	195
7.58.2 Variable Documentation	195
7.58.2.1 boapPort	195
7.59 Boap.h File Reference	195
7.59.1 Typedef Documentation	196
7.59.1.1 BoapFunc	196
7.59.1.2 BoapService	196
7.59.2 Enumeration Type Documentation	196
7.59.2.1 BoapPriority	196
7.59.2.2 BoapType	196
7.59.3 Variable Documentation	196
7.59.3.1 BoapMagic	196
7.60 BoapMc.cpp File Reference	196
7.60.1 Macro Definition Documentation	197
7.60.1.1 DEBUG_LOCAL	197
7.60.1.2 DEBUG_LOCAL1	197
7.60.1.3 dl1printf	197
7.60.1.4 dlprintf	197
7.61 BoapMc.h File Reference	197
7.61.1 Enumeration Type Documentation	197
7.61.1.1 BoapMcType	197
7.61.2 Function Documentation	198
7.61.2.1 __attribute__	198
7.61.3 Variable Documentation	198
7.61.3.1 __attribute__	198
7.61.3.2 addressFrom	198
7.61.3.3 addressTo	198

7.61.3.4	checksum	198
7.61.3.5	cmd	198
7.61.3.6	error	198
7.61.3.7	length	198
7.62	BoapnsC.cpp File Reference	198
7.63	BoapnsC.h File Reference	198
7.64	BoapnsD.cpp File Reference	199
7.65	BoapnsD.h File Reference	199
7.66	BoapSimple.cc File Reference	199
7.66.1	Macro Definition Documentation	200
7.66.1.1	DEBUG	200
7.66.1.2	dprintf	200
7.66.2	Variable Documentation	200
7.66.2.1	roundSize	200
7.67	BoapSimple.h File Reference	200
7.67.1	Typedef Documentation	201
7.67.1.1	BoapFunc	201
7.67.1.2	BoapService	201
7.67.1.3	Double	201
7.67.1.4	Int16	201
7.67.1.5	Int32	201
7.67.1.6	Int8	201
7.67.1.7	UInt16	201
7.67.1.8	UInt32	201
7.67.1.9	UInt8	201
7.67.2	Enumeration Type Documentation	201
7.67.2.1	BoapType	201
7.68	BObj.cpp File Reference	201
7.69	BObj.h File Reference	201
7.70	BObjStringFormat.cpp File Reference	202
7.70.1	Function Documentation	203
7.70.1.1	toBDictStringFromJson	203
7.70.1.2	toBString	203
7.70.1.3	toBString	203
7.70.1.4	toBString	203
7.70.1.5	toBString	203
7.70.1.6	toBString	203
7.70.1.7	toBString	203
7.70.1.8	toBString	203
7.70.1.9	toBString	203

7.70.1.10 toBString	203
7.70.1.11 toBString	203
7.70.1.12 toBString	203
7.70.1.13 toBString	203
7.70.1.14 toBString	203
7.70.1.15 toBString	203
7.70.1.16 toBString	203
7.70.1.17 toBString	203
7.70.1.18 toBString	203
7.70.1.19 toBString	203
7.70.1.20 toBStringJson	203
7.70.1.21 toBStringJson	203
7.70.1.22 toBStringJson	203
7.70.1.23 toBStringJson	203
7.70.1.24 toBStringJson	203
7.70.1.25 toBStringJson	203
7.70.1.26 toBStringJson	203
7.70.1.27 toBStringJson	203
7.70.1.28 toBStringJson	204
7.70.1.29 toBStringJson	204
7.70.1.30 toBStringJson	204
7.70.1.31 toBStringJson	204
7.70.1.32 toBStringJson	204
7.70.1.33 toBStringJson	204
7.70.1.34 toBStringJson	204
7.70.1.35 toBStringJson	204
7.70.1.36 toBStringJson	204
7.70.1.37 toBStringJson	204
7.71 BObjStringFormat.h File Reference	204
7.71.1 Function Documentation	205
7.71.1.1 base64_decode	205
7.71.1.2 base64_encode	205
7.71.1.3 toBDictStringFromJson	205
7.71.1.4 toBString	205
7.71.1.5 toBString	205
7.71.1.6 toBString	205
7.71.1.7 toBString	205
7.71.1.8 toBString	205
7.71.1.9 toBString	205
7.71.1.10 toBString	205

7.71.1.11 toBString	205
7.71.1.12 toBString	205
7.71.1.13 toBString	205
7.71.1.14 toBString	205
7.71.1.15 toBString	205
7.71.1.16 toBString	205
7.71.1.17 toBString	205
7.71.1.18 toBString	205
7.71.1.19 toBString	205
7.71.1.20 toBStringJson	205
7.71.1.21 toBStringJson	206
7.71.1.22 toBStringJson	206
7.71.1.23 toBStringJson	206
7.71.1.24 toBStringJson	206
7.71.1.25 toBStringJson	206
7.71.1.26 toBStringJson	206
7.71.1.27 toBStringJson	206
7.71.1.28 toBStringJson	206
7.71.1.29 toBStringJson	206
7.71.1.30 toBStringJson	206
7.71.1.31 toBStringJson	206
7.71.1.32 toBStringJson	206
7.71.1.33 toBStringJson	206
7.71.1.34 toBStringJson	206
7.71.1.35 toBStringJson	206
7.72 BPoll.cpp File Reference	206
7.73 BPoll.h File Reference	206
7.74 BQueue.h File Reference	207
7.74.1 Typedef Documentation	207
7.74.1.1 BQueueInt	207
7.75 BRefData.cpp File Reference	207
7.75.1 Macro Definition Documentation	207
7.75.1.1 CHUNK	207
7.76 BRefData.h File Reference	207
7.77 BRtc.cpp File Reference	208
7.78 BRtc.h File Reference	208
7.79 BRWLock.cpp File Reference	208
7.80 BRWLock.h File Reference	208
7.81 BSema.cpp File Reference	208
7.82 BSema.h File Reference	209

7.83	BSemaphore.cpp File Reference	209
7.84	BSemaphore.h File Reference	209
7.85	BSocket.cpp File Reference	209
7.85.1	Macro Definition Documentation	210
7.85.1.1	IP_MTU	210
7.86	BSocket.h File Reference	210
7.87	BSpi.cpp File Reference	210
7.88	BSpi.h File Reference	210
7.89	BString.cpp File Reference	211
7.89.1	Macro Definition Documentation	211
7.89.1.1	MINUS	211
7.89.1.2	STRIP	211
7.89.2	Function Documentation	212
7.89.2.1	barrayToString	212
7.89.2.2	blistToString	212
7.89.2.3	bstringListinList	212
7.89.2.4	bstringToArray	212
7.89.2.5	bstringToList	212
7.89.2.6	charToArray	212
7.89.2.7	charToList	212
7.89.2.8	fromBString	212
7.89.2.9	fromBString	212
7.89.2.10	fromBString	212
7.89.2.11	fromBString	212
7.89.2.12	fromBString	212
7.89.2.13	fromBString	212
7.89.2.14	gmatch	212
7.89.2.15	operator<<	212
7.89.2.16	operator>>	212
7.89.2.17	toBString	212
7.89.2.18	toBString	212
7.89.2.19	toBString	212
7.89.2.20	toBString	212
7.89.2.21	toBString	212
7.89.2.22	toBString	212
7.89.3	Variable Documentation	212
7.89.3.1	base64_decode_table	212
7.90	BString.h File Reference	213
7.90.1	Function Documentation	213
7.90.1.1	fromBString	213

7.90.1.2	fromBString	213
7.90.1.3	fromBString	213
7.90.1.4	fromBString	213
7.90.1.5	fromBString	213
7.90.1.6	fromBString	213
7.90.1.7	operator<<	213
7.90.1.8	operator>>	213
7.90.1.9	toBString	214
7.90.1.10	toBString	214
7.90.1.11	toBString	214
7.90.1.12	toBString	214
7.90.1.13	toBString	214
7.90.1.14	toBString	214
7.91	BStringLocked.h File Reference	214
7.92	BTable.cpp File Reference	214
7.93	BTable.h File Reference	214
7.94	BThread.cpp File Reference	214
7.95	BThread.h File Reference	215
7.96	BTime.cpp File Reference	215
7.96.1	Function Documentation	215
7.96.1.1	yearDays	215
7.96.1.2	yearIsLeap	215
7.96.2	Variable Documentation	215
7.96.2.1	monDays	215
7.97	BTime.h File Reference	215
7.98	BTimer.cpp File Reference	216
7.99	BTimer.h File Reference	216
7.100	BTimeStamp.cpp File Reference	216
7.100.1	Function Documentation	216
7.100.1.1	fromBString	216
7.100.1.2	toBString	216
7.100.2	Variable Documentation	216
7.100.2.1	mon_yday	216
7.101	BTimeStamp.h File Reference	217
7.101.1	Function Documentation	217
7.101.1.1	fromBString	217
7.101.1.2	toBString	217
7.102	BTimeStampMs.cpp File Reference	217
7.102.1	Variable Documentation	217
7.102.1.1	mon_yday	217

7.103BTimeStampMs.h File Reference	217
7.104BTypes.h File Reference	218
7.104.1 Typedef Documentation	219
7.104.1.1 BArrayDouble	219
7.104.1.2 BArrayFloat	219
7.104.1.3 BChar	219
7.104.1.4 BDouble	219
7.104.1.5 BFloat	219
7.104.1.6 BFloat32	219
7.104.1.7 BFloat64	219
7.104.1.8 BInt	219
7.104.1.9 BInt16	219
7.104.1.10BInt32	219
7.104.1.11BInt64	219
7.104.1.12BInt8	219
7.104.1.13BBool	219
7.104.1.14BSize	219
7.104.1.15BTimeout	219
7.104.1.16BUInt	219
7.104.1.17BUInt16	219
7.104.1.18BUInt32	219
7.104.1.19BUInt64	219
7.104.1.20BUInt8	219
7.104.2 Enumeration Type Documentation	220
7.104.2.1 BType	220
7.104.2.2 BTypeComp	220
7.104.3 Function Documentation	220
7.104.3.1 byteSwap16	220
7.104.3.2 byteSwap32	220
7.104.3.3 byteSwap64	220
7.104.3.4 byteSwap8	220
7.104.3.5 timeoutTicks	220
7.104.4 Variable Documentation	220
7.104.4.1 BTimeoutForever	220
7.105BUrl.cpp File Reference	221
7.106BUrl.h File Reference	221

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Boapns	15
----------------------------------	----

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

BAtomic< Type >	18
BAtomicCount	19
BBuffer	20
BBufferStore	22
BoapPacket	107
BComms	25
BCond	27
BCondBool	27
BCondInt	29
BCondResource	31
BCondValue	32
BCondWrap	35
BDate	38
BDebugBacktrace	41
BDictItem< Type >	44
BDuration	48
BEntry	50
BError	56
BEvent1Error	63
BErrorTime	59
BEvent	61
BEvent1	62
BEvent1Error	63
BEvent1Int	63
BEvent1Pipe	65
BEventPipe	66
BFifo< Type >	67
BFifo< BoapMcPacket >	67
BFifoCirc< Type >	71
BFifoCircPos	74
BFile	75
BFileCsv	79
BIter	81
BList< T >	81
BQueue< T >	123
BList< BArray< BString > >	81

BList< BDictItem< Type > >	81
BDict< Type >	42
BConfig	37
BList< BEntry >	81
BEntryList	54
BEntryFile	52
BList< BNameValue< T > >	81
BNameValueList< T >	92
BList< BoapFuncEntry >	81
BList< BoapMcPacket >	81
BQueue< BoapMcPacket >	123
BList< BoapServerConnection * >	81
BList< BoapServiceEntry >	81
BList< BString >	81
BList< BStringList >	81
BFileData	80
BList< struct dirent * >	81
BDir	46
BMutex	87
BStringMutex	152
BMutexLock	89
BMysql	90
BNameValue< T >	91
BNode	93
BList< T >::Node	169
Boapns::BoapEntry	97
BoapFuncEntry	97
BoapMcClientObject	98
BoapMcComms	100
BoapMcPacket	104
BoapMcPacketHead	104
BoapMcServiceObject	105
BoapMcSignalObject	106
BoapPacketHead	110
BoapServiceEntry	116
BoapServiceObject	117
BObj	120
BObjMember	121
BPoll	121
BRefData	125
BRtc	127
BRWLock	129
BSema	131
BSemaphore	132
BSemaphoreCount	133
BSocket	134
BoapClientObject	93
Boapns::Boapns	106
BoapClientObject	93
BoapSignalObject	119
BoapSignalObject	119
BSocketAddress	137
BSocketAddressINET	139
BSpi	141
BString	142

BStringLocked	151
BTable	152
BThread	153
BoapServer	111
BoapServerConnection	115
BRtcThreaded	128
BTime	155
BTimer	157
BTimeStamp	159
BTimeStampMs	164
BUrl	168
map	
BDictMap< Value >	45
vector	
BArray< T >	17
BArray< BList< BIter > >	17
BArray< BString >	17
BArray< int >	17

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

BArray< T >	17
BAtomic< Type >	
BAtomic class	18
BAtomicCount	
BAtomicCount class	19
BBuffer	20
BBufferStore	22
BComms	25
BCond	27
BCondBool	
Thread conditional boolean	27
BCondInt	
Thread conditional value	29
BCondResource	
Resource lock	31
BCondValue	
Thread conditional value	32
BCondWrap	35
BConfig	
This class implements the configuration file access	37
BDate	38
BDebugBacktrace	41
BDict< Type >	42
BDictItem< Type >	
Template based Dictionary class	44
BDictMap< Value >	45
BDir	
File system directory class	46
BDuration	48
BEntry	
Manipulate a name value pair	50
BEntryFile	
File of Entries	52
BEntryList	
List of Entries. Where an entry is a name value pair	54
BError	
Error return class	56

BErrorTime	
Error return class	59
BEvent	61
BEvent1	
This class provides a base class for all event objects that can be sent over the events interface	62
BEvent1Error	63
BEvent1Int	
This class provides an interface for sending simple integer events via a file descriptor. This allows threads to send events that can be picked up by the poll system call	63
BEvent1Pipe	
This class provides a base interface for sending events via a pipe. This allows threads to send events that can be picked up by the poll system call	65
BEventPipe	
This class provides an interface for sending simple integer events via a pipe file descriptor	66
BFifo< Type >	67
BFifoCirc< Type >	
This class implements a thread safe FIFO buffer	71
BFifoCircPos	
This class implements a pointer into the Fifo's circular buffer	74
BFile	
File operations class	75
BFileCsv	79
BFileData	80
BIter	
Iterator for BList	81
BList< T >	
Template based list class	81
BMutex	
Mutex class	87
BMutexLock	89
BMySQL	90
BNameValue< T >	91
BNameValueList< T >	92
BNode	93
BoapClientObject	93
Boapns::BoapEntry	97
BoapFuncEntry	97
BoapMcClientObject	98
BoapMcComms	100
BoapMcPacket	104
BoapMcPacketHead	104
BoapMcServiceObject	105
BoapMcSignalObject	106
Boapns::Boapns	106
BoapPacket	107
BoapPacketHead	110
BoapServer	111
BoapServerConnection	115
BoapServiceEntry	116
BoapServiceObject	117
BoapSignalObject	119
BObj	120
BObjMember	121
BPoll	
This class provides an interface for polling a number of file descriptors. It uses round robin polling	121
BQueue< T >	
Queue class	123
BRefData	125

BRtc		
	Realtime clock	127
BRtcThreaded		
	Threaded real time clock	128
BRWLock		
	Thread read-write locks	129
BSema		
	Sempahore class	131
BSemaphore		
	Semaphore class	132
BSemaphoreCount		133
BSocket		134
BSocketAddress		
	Socket Address	137
BSocketAddressINET		
	IP aware socket address	139
BSpi		
	BSpi class	141
BString		142
BStringLocked		151
BStringMutex		152
BTable		152
BThread		153
BTime		155
BTimer		
	Stopwatch style timer	157
BTimeStamp		159
BTimeStampMs		164
BUrl		
	Basic access to a Url	168
BList< T >::Node		169

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

BArray.h	171
BAtomic.h	171
BAtomicCount.h	172
BBuffer.cpp	172
BBuffer.h	172
BComms.cpp	173
BComms.h	173
BComplex.h	173
BCond.cpp	174
BCond.h	174
BCondInt.cpp	174
BCondInt.h	174
BConfig.cpp	175
BConfig.h	175
BCrc16.cpp	175
BCrc16.h	176
BDate.cpp	176
BDate.h	177
BDebug.cpp	178
BDebug.h	179
BDict.cpp	180
BDict.h	181
BDictMap.h	181
BDir.cpp	182
BDir.h	182
BDuration.cpp	182
BDuration.h	183
BEndian.cpp	183
BEndian.h	183
BEntry.cpp	186
BEntry.h	186
BError.cpp	187
BError.h	187
BErrorTime.cpp	188
BErrorTime.h	188
BEvent.cpp	188
BEvent.h	188
BEvent1.cpp	189

BEvent1.h	189
BFifo.h	190
BFifo.inc	190
BFifoCirc.cpp	190
BFifoCirc.h	190
BFifoCirc.inc	191
BFile.cpp	191
BFile.h	191
BFileCsv.cpp	191
BFileCsv.h	192
BFileData.cpp	192
BFileData.h	192
BList.h	192
BList_func.h	193
BMutex.cpp	193
BMutex.h	193
BMySQL.cpp	193
BMySQL.h	193
BNameValue.h	194
Boap.cpp	194
Boap.h	195
BoapMc.cpp	196
BoapMc.h	197
BoapnsC.cpp	198
BoapnsC.h	198
BoapnsD.cpp	199
BoapnsD.h	199
BoapSimple.cc	199
BoapSimple.h	200
BObj.cpp	201
BObj.h	201
BObjStringFormat.cpp	202
BObjStringFormat.h	204
BPoll.cpp	206
BPoll.h	206
BQueue.h	207
BRefData.cpp	207
BRefData.h	207
BRtc.cpp	208
BRtc.h	208
BRWLock.cpp	208
BRWLock.h	208
BSema.cpp	208
BSema.h	209
BSemaphore.cpp	209
BSemaphore.h	209
BSocket.cpp	209
BSocket.h	210
BSpi.cpp	210
BSpi.h	210
BString.cpp	211
BString.h	213
BStringLocked.h	214
BTable.cpp	214
BTable.h	214
BThread.cpp	214
BThread.h	215
BTime.cpp	215

BTime.h	215
BTimer.cpp	216
BTimer.h	216
BTimeStamp.cpp	216
BTimeStamp.h	217
BTimeStampMs.cpp	217
BTimeStampMs.h	217
BTypes.h	218
BUrl.cpp	221
BUrl.h	221

Chapter 5

Namespace Documentation

5.1 Boapns Namespace Reference

Classes

- class [Boapns](#)
- class [BoapEntry](#)

Variables

- const [BUInt32](#) [apiVersion](#) = 0

5.1.1 Variable Documentation

5.1.1.1 const BUInt32 Boapns::apiVersion = 0

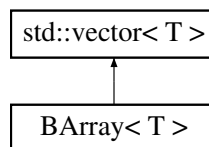
Chapter 6

Class Documentation

6.1 BArray< T > Class Template Reference

```
#include <BArray.h>
```

Inheritance diagram for BArray< T >:



Public Types

- typedef int(* SortFunc)(T &a, T &b)
Prototype for sorting function.

Public Member Functions

- BArray ()
- BArray (BSize size, T value=T())
- BArray (const BArray &array)
- BUInt number () const
- void append (const T &value)
- void append (const BArray< T > &array)
- void insert (BUInt pos, const T &value)
- void del (BUInt pos, BUInt num=1)
- T & rear ()
- void sort ()

6.1.1 Detailed Description

```
template<class T>class BArray< T >
```

Template based Array class. This is based on the Standard C++ library vector class and has all of the functionality of that class.

6.1.2 Member Typedef Documentation

6.1.2.1 `template<class T> typedef int(* BArray< T >::SortFunc)(T &a, T &b)`

Prototype for sorting function.

6.1.3 Constructor & Destructor Documentation

6.1.3.1 `template<class T> BArray< T >::BArray () [inline]`

6.1.3.2 `template<class T> BArray< T >::BArray (BSize size, T value = T()) [inline]`

6.1.3.3 `template<class T> BArray< T >::BArray (const BArray< T > & array) [inline]`

6.1.4 Member Function Documentation

6.1.4.1 `template<class T> void BArray< T >::append (const T & value) [inline]`

6.1.4.2 `template<class T> void BArray< T >::append (const BArray< T > & array)`

6.1.4.3 `template<class T> void BArray< T >::del (BUInt pos, BUInt num = 1) [inline]`

6.1.4.4 `template<class T> void BArray< T >::insert (BUInt pos, const T & value) [inline]`

6.1.4.5 `template<class T> BUInt BArray< T >::number () const [inline]`

6.1.4.6 `template<class T> T& BArray< T >::rear () [inline]`

6.1.4.7 `template<class T> void BArray< T >::sort () [inline]`

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

- [BArray.h](#)

6.2 BAtomic< Type > Class Template Reference

[BAtomic](#) class.

```
#include <BAtomic.h>
```

Public Member Functions

- [BAtomic](#) (Type value=0)
- Type [getValue](#) () const
- Type [add](#) (long value)
- Type [operator++](#) (int)
- Type [operator++](#) ()
- Type [operator--](#) (int)
- Type [operator--](#) ()
- [operator Type](#) () const

Private Attributes

- Type [ovalue](#)

6.2.1 Detailed Description

template<class Type>class BAtomic< Type >

[BAtomic](#) class.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 template<class Type > BAtomic< Type >::BAtomic (Type *value* = 0) [inline]

6.2.3 Member Function Documentation

6.2.3.1 template<class Type > Type BAtomic< Type >::add (long *value*) [inline]

6.2.3.2 template<class Type > Type BAtomic< Type >::getValue () const [inline]

6.2.3.3 template<class Type > BAtomic< Type >::operator Type () const [inline]

6.2.3.4 template<class Type > Type BAtomic< Type >::operator++ (int) [inline]

6.2.3.5 template<class Type > Type BAtomic< Type >::operator++ () [inline]

6.2.3.6 template<class Type > Type BAtomic< Type >::operator-- (int) [inline]

6.2.3.7 template<class Type > Type BAtomic< Type >::operator-- () [inline]

6.2.4 Member Data Documentation

6.2.4.1 template<class Type > Type BAtomic< Type >::ovalue [mutable],[private]

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

- [BAtomic.h](#)

6.3 BAtomicCount Class Reference

[BAtomicCount](#) class.

```
#include <BAtomicCount.h>
```

Public Member Functions

- [BAtomicCount](#) (long *value*=0)
- long [getValue](#) () const
- long [add](#) (long *value*)
- long [operator++](#) (int)
- long [operator++](#) ()
- long [operator--](#) (int)
- long [operator--](#) ()
- [operator long](#) () const

Private Attributes

- [_Atomic_word](#) [ovalue](#)

6.3.1 Detailed Description

[BAtomicCount](#) class.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 `BAtomicCount::BAtomicCount (long value = 0)` `[inline]`

6.3.3 Member Function Documentation

6.3.3.1 `long BAtomicCount::add (long value)` `[inline]`

6.3.3.2 `long BAtomicCount::getValue () const` `[inline]`

6.3.3.3 `BAtomicCount::operator long () const` `[inline]`

6.3.3.4 `long BAtomicCount::operator++ (int)` `[inline]`

6.3.3.5 `long BAtomicCount::operator++ ()` `[inline]`

6.3.3.6 `long BAtomicCount::operator-- (int)` `[inline]`

6.3.3.7 `long BAtomicCount::operator-- ()` `[inline]`

6.3.4 Member Data Documentation

6.3.4.1 `_Atomic_word BAtomicCount::ovalue` `[mutable], [private]`

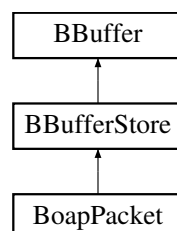
The documentation for this class was generated from the following file:

- [BAtomicCount.h](#)

6.4 BBuffer Class Reference

```
#include <BBuffer.h>
```

Inheritance diagram for BBuffer:



Public Member Functions

- [BBuffer](#) ([BUInt](#) *size*=0)
Create and manipulate a data buffer. On creation the buffer size defaults to 1024 bytes.
- [~BBuffer](#) ()
- `int` [setSize](#) ([BUInt32](#) *size*)

Sets the bufer size.

- int [setData](#) (const void **data*, [BUInt32 size](#))

Sets buffer data resized to contain the data.

- int [writeData](#) ([BUInt32 pos](#), const void **data*, [BUInt32 size](#))

Writes data into buffer from offset pos.

- char * [data](#) ()

The data.

- [BUInt32 size](#) ()

Size of the buffer in bytes.

- int [resize](#) ([BUInt32 size](#))

Alternative to [setSize\(\)](#)

Protected Attributes

- [BUInt32 odataSize](#)
- char * [odata](#)
- [BUInt32 osize](#)

6.4.1 Constructor & Destructor Documentation

6.4.1.1 BBuffer::BBuffer ([BUInt size](#) = 0)

Create and manipulate a data buffer. On creation the buffer size defaults to 1024 bytes.

6.4.1.2 BBuffer::~~BBuffer ()

6.4.2 Member Function Documentation

6.4.2.1 char * BBuffer::data ()

The data.

6.4.2.2 int BBuffer::resize ([BUInt32 size](#)) `[inline]`

Alternative to [setSize\(\)](#)

6.4.2.3 int BBuffer::setData (const void * *data*, [BUInt32 size](#))

Sets buffer data resized to contain the data.

6.4.2.4 int BBuffer::setSize ([BUInt32 size](#))

Sets the bufer size.

6.4.2.5 [BUInt32](#) BBuffer::size ()

Size of the buffer in bytes.

6.4.2.6 `int BBuffer::writeData (BUInt32 pos, const void * data, BUInt32 size)`

Writes data into buffer from offset pos.

6.4.3 Member Data Documentation

6.4.3.1 `char* BBuffer::odata` [protected]

6.4.3.2 `BUInt32 BBuffer::odataSize` [protected]

6.4.3.3 `BUInt32 BBuffer::osize` [protected]

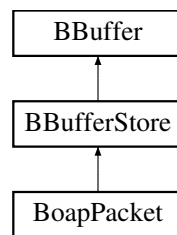
The documentation for this class was generated from the following files:

- [BBuffer.h](#)
- [BBuffer.cpp](#)

6.5 BBufferStore Class Reference

```
#include <BBuffer.h>
```

Inheritance diagram for BBufferStore:



Public Member Functions

- `BBufferStore` (`BUInt size=0`, `int swapBytes=BBigEndian`)
- `~BBufferStore` ()
- `BUInt32 getPos` ()
- `void setPos` (`BUInt32 pos`)
- `BString getHexString` ()
- `void setHexString` (`BString s`)
- `int push` (`BInt8 v`)
- `int push` (`BUInt8 v`)
- `int push` (`BInt16 v`)
- `int push` (`BUInt16 v`)
- `int push` (`BInt32 v`)
- `int push` (`BUInt32 v`)
- `int push` (`BInt64 v`)
- `int push` (`BUInt64 v`)
- `int push` (`BFloat32 v`)
- `int push` (`BFloat64 v`)
- `int push` (`const BString &v`)
- `int push` (`const BError &v`)
- `int push` (`const BTimeStamp &v`)

- int [push](#) (const [BComplex](#) &v)
- int [push](#) ([BUInt32](#) nBytes, const void *[data](#), const char *swapType="1")
- int [pop](#) ([BInt8](#) &v)
- int [pop](#) ([BUInt8](#) &v)
- int [pop](#) ([BInt16](#) &v)
- int [pop](#) ([BUInt16](#) &v)
- int [pop](#) ([BInt32](#) &v)
- int [pop](#) ([BUInt32](#) &v)
- int [pop](#) ([BInt64](#) &v)
- int [pop](#) ([BUInt64](#) &v)
- int [pop](#) ([BFloat32](#) &v)
- int [pop](#) ([BFloat64](#) &v)
- int [pop](#) ([BString](#) &v)
- int [pop](#) ([BError](#) &v)
- int [pop](#) ([BTimeStamp](#) &v)
- int [pop](#) ([BComplex](#) &v)
- int [pop](#) ([BUInt32](#) nBytes, void *[data](#), const char *swapType="1")

Protected Attributes

- [BUInt32](#) opos
- int [oswapBytes](#)

6.5.1 Constructor & Destructor Documentation

6.5.1.1 [BBufferStore::BBufferStore](#) ([BUInt](#) size = 0, int [swapBytes](#) = [BBigEndian](#))

6.5.1.2 [BBufferStore::~~BBufferStore](#) ()

6.5.2 Member Function Documentation

6.5.2.1 [BString](#) [BBufferStore::getHexString](#) ()

6.5.2.2 [BUInt32](#) [BBufferStore::getPos](#) ()

6.5.2.3 int [BBufferStore::pop](#) ([BInt8](#) & v)

6.5.2.4 int [BBufferStore::pop](#) ([BUInt8](#) & v)

6.5.2.5 int [BBufferStore::pop](#) ([BInt16](#) & v)

6.5.2.6 int [BBufferStore::pop](#) ([BUInt16](#) & v)

6.5.2.7 int [BBufferStore::pop](#) ([BInt32](#) & v)

6.5.2.8 int [BBufferStore::pop](#) ([BUInt32](#) & v)

6.5.2.9 int [BBufferStore::pop](#) ([BInt64](#) & v)

6.5.2.10 int [BBufferStore::pop](#) ([BUInt64](#) & v)

6.5.2.11 int [BBufferStore::pop](#) ([BFloat32](#) & v)

6.5.2.12 int [BBufferStore::pop](#) ([BFloat64](#) & v)

- 6.5.2.13 `int BBufferStore::pop (BString & v)`
- 6.5.2.14 `int BBufferStore::pop (BError & v)`
- 6.5.2.15 `int BBufferStore::pop (BTimeStamp & v)`
- 6.5.2.16 `int BBufferStore::pop (BComplex & v)`
- 6.5.2.17 `int BBufferStore::pop (BUInt32 nBytes, void * data, const char * swapType = "1")`
- 6.5.2.18 `int BBufferStore::push (BInt8 v)`
- 6.5.2.19 `int BBufferStore::push (BUInt8 v)`
- 6.5.2.20 `int BBufferStore::push (BInt16 v)`
- 6.5.2.21 `int BBufferStore::push (BUInt16 v)`
- 6.5.2.22 `int BBufferStore::push (BInt32 v)`
- 6.5.2.23 `int BBufferStore::push (BUInt32 v)`
- 6.5.2.24 `int BBufferStore::push (BInt64 v)`
- 6.5.2.25 `int BBufferStore::push (BUInt64 v)`
- 6.5.2.26 `int BBufferStore::push (BFloat32 v)`
- 6.5.2.27 `int BBufferStore::push (BFloat64 v)`
- 6.5.2.28 `int BBufferStore::push (const BString & v)`
- 6.5.2.29 `int BBufferStore::push (const BError & v)`
- 6.5.2.30 `int BBufferStore::push (const BTimeStamp & v)`
- 6.5.2.31 `int BBufferStore::push (const BComplex & v)`
- 6.5.2.32 `int BBufferStore::push (BUInt32 nBytes, const void * data, const char * swapType = "1")`
- 6.5.2.33 `void BBufferStore::setHexString (BString s)`
- 6.5.2.34 `void BBufferStore::setPos (BUInt32 pos)`

6.5.3 Member Data Documentation

- 6.5.3.1 `BUInt32 BBufferStore::opos` [protected]
- 6.5.3.2 `int BBufferStore::oswapBytes` [protected]

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

- [BBuffer.h](#)
- [BBuffer.cpp](#)

6.6 BComms Class Reference

```
#include <BComms.h>
```

Public Types

- enum [Wait](#) { [WaitNone](#) = 0x00, [WaitRead](#) = 0x01, [WaitWrite](#) = 0x02, [WaitError](#) = 0x04 }

Public Member Functions

- [BComms](#) ()
- virtual [~BComms](#) ()
- virtual [BError](#) [init](#) ()
- virtual [BError](#) [setPacketMode](#) ([Bool](#) packetMode)
Set packet mode.
- virtual [Bool](#) [packetMode](#) ()
Device is in packet mode.
- virtual [BError](#) [setTimeout](#) ([BInt](#) timeoutMs)
Set communication timeout.
- virtual [BUInt](#) [writeAvailable](#) ()
- virtual [BError](#) [write](#) (const void *data, [BUInt32](#) nBytes, [BUInt32](#) &nTrans)=0
- virtual [BUInt](#) [readAvailable](#) ()
- virtual [BError](#) [read](#) (void *data, [BUInt32](#) num, [BUInt32](#) &nTrans)=0
- virtual [BError](#) [wait](#) ([BUInt8](#) events, [BInt](#) timeout=-1, [BUInt32](#) num=1)
- virtual void [eventQueue](#) ([BEventQueue](#) *eventQueue, [BInt32](#) event, [BUInt](#) num=1)

Protected Attributes

- [Bool](#) [opacketMode](#)
- [BInt32](#) [otimeout](#)
- [BEventQueue](#) * [oeventQueue](#)
- [BInt32](#) [oevent](#)
- [BUInt](#) [oeventNum](#)

6.6.1 Member Enumeration Documentation

6.6.1.1 enum BComms::Wait

Enumerator

WaitNone

WaitRead

WaitWrite

WaitError

6.6.2 Constructor & Destructor Documentation

6.6.2.1 **BComms::BComms** ()

6.6.2.2 **BComms::~~BComms** () [virtual]

6.6.3 Member Function Documentation

6.6.3.1 **void BComms::eventQueue** (**BEventQueue** * *eventQueue*, **BInt32** *event*, **BUInt** *num* = 1) [virtual]

6.6.3.2 **BError BComms::init** () [virtual]

6.6.3.3 **Bool BComms::packetMode** () [virtual]

Device is in packet mode.

6.6.3.4 **virtual BError BComms::read** (**void** * *data*, **BUInt32** *num*, **BUInt32** & *nTrans*) [pure virtual]

6.6.3.5 **BUInt BComms::readAvailable** () [virtual]

6.6.3.6 **BError BComms::setPacketMode** (**Bool** *packetMode*) [virtual]

Set packet mode.

6.6.3.7 **BError BComms::setTimeout** (**BInt** *timeoutMs*) [virtual]

Set communication timeout.

6.6.3.8 **BError BComms::wait** (**BUInt8** *events*, **BInt** *timeout* = -1, **BUInt32** *num* = 1) [virtual]

6.6.3.9 **virtual BError BComms::write** (**const void** * *data*, **BUInt32** *nBytes*, **BUInt32** & *nTrans*) [pure virtual]

6.6.3.10 **BUInt BComms::writeAvailable** () [virtual]

6.6.4 Member Data Documentation

6.6.4.1 **BInt32 BComms::oevent** [protected]

6.6.4.2 **BUInt BComms::oeventNum** [protected]

6.6.4.3 **BEventQueue* BComms::oeventQueue** [protected]

6.6.4.4 **Bool BComms::opacketMode** [protected]

6.6.4.5 **BInt32 BComms::otimeout** [protected]

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

- [BComms.h](#)
- [BComms.cpp](#)

6.7 BCond Class Reference

```
#include <BCond.h>
```

Public Member Functions

- [BCond](#) ()
Thread conditional variable.
- [~BCond](#) ()
- int [signal](#) ()
- int [wait](#) ()
- int [timedWait](#) (int timeOutUs)

Private Attributes

- pthread_mutex_t [omutex](#)
- pthread_cond_t [ocond](#)

6.7.1 Constructor & Destructor Documentation

6.7.1.1 BCond::BCond ()

Thread conditional variable.

6.7.1.2 BCond::~~BCond ()

6.7.2 Member Function Documentation

6.7.2.1 int BCond::signal ()

6.7.2.2 int BCond::timedWait (int *timeOutUs*)

6.7.2.3 int BCond::wait ()

6.7.3 Member Data Documentation

6.7.3.1 pthread_cond_t BCond::ocond [private]

6.7.3.2 pthread_mutex_t BCond::omutex [private]

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

- [BCond.h](#)
- [BCond.cpp](#)

6.8 BCondBool Class Reference

Thread conditional boolean.

```
#include <BCondInt.h>
```

Public Member Functions

- [BCondBool](#) ()
- [~BCondBool](#) ()
- [int set](#) ()
Set value. Wakes waiting.
- [int clear](#) ()
Clear Value.
- [int value](#) ()
Current value.
- [int wait](#) ()
Wait until value is true.
- [int timedWait](#) (int *timeOutUs*)
Wait until set, with timeout.
- [operator int](#) ()

Private Attributes

- `pthread_mutex_t` [omutex](#)
- `pthread_cond_t` [ocond](#)
- `int` [ovalue](#)

6.8.1 Detailed Description

Thread conditional boolean.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 `BCondBool::BCondBool ()`

6.8.2.2 `BCondBool::~~BCondBool ()`

6.8.3 Member Function Documentation

6.8.3.1 `int BCondBool::clear ()`

Clear Value.

6.8.3.2 `BCondBool::operator int ()` `[inline]`

6.8.3.3 `int BCondBool::set ()`

Set value. Wakes waiting.

6.8.3.4 `int BCondBool::timedWait (int timeOutUs)`

Wait until set, with timeout.

6.8.3.5 `int BCondBool::value ()`

Current value.

6.8.3.6 `int BCondBool::wait ()`

Wait until value is true.

6.8.4 Member Data Documentation

6.8.4.1 `pthread_cond_t BCondBool::ocond` [private]6.8.4.2 `pthread_mutex_t BCondBool::omutex` [private]6.8.4.3 `int BCondBool::ovalue` [private]

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

- [BCondInt.h](#)
- [BCondInt.cpp](#)

6.9 BCondInt Class Reference

Thread conditional value.

```
#include <BCondInt.h>
```

Public Member Functions

- [BCondInt](#) ()
- [~BCondInt](#) ()
- void [setValue](#) (BInt value)
Set the value. Wakes waiting.
- [BInt value](#) () const
Current value.
- [BInt increment](#) (BInt v=1)
Increment. Wakes waiting.
- [BInt decrement](#) (BInt v=1)
Decrement. Wakes waiting.
- [Bool waitMoreThanOrEqual](#) (BInt v, [Bool](#) decrement=0, [BTimeout](#) timeoutUs=[BTimeoutForever](#))
Wait until value is at least the value given.
- [Bool waitLessThanOrEqual](#) (BInt v, [Bool](#) increment=0, [BTimeout](#) timeoutUs=[BTimeoutForever](#))
Wait until value is equal to or below the value given.
- [Bool waitLessThan](#) (BInt v, [BTimeout](#) timeoutUs=[BTimeoutForever](#))
Wait until value is equal to or below the value given.
- void [operator+=](#) (int v)
Add to value. Wakes waiting.
- void [operator-=](#) (int v)
Subtract from value. Wakes waiting.
- void [operator++](#) (int)
Increment value. Wakes waiting.
- void [operator--](#) (int)
Decrement value. Wakes waiting.

Private Attributes

- pthread_mutex_t [omutex](#)
- pthread_cond_t [ocond](#)
- BInt [ovalue](#)

6.9.1 Detailed Description

Thread conditional value.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 BCondInt::BCondInt ()

6.9.2.2 BCondInt::~~BCondInt ()

6.9.3 Member Function Documentation

6.9.3.1 BInt BCondInt::decrement (BInt v = 1)

Decrement. Wakes waiting.

6.9.3.2 BInt BCondInt::increment (BInt v = 1)

Increment. Wakes waiting.

6.9.3.3 void BCondInt::operator++ (int) [inline]

Increment value. Wakes waiting.

6.9.3.4 void BCondInt::operator+= (int v) [inline]

Add to value. Wakes waiting.

6.9.3.5 void BCondInt::operator-- (int) [inline]

Decrement value. Wakes waiting.

6.9.3.6 void BCondInt::operator-= (int v) [inline]

Subtract from value. Wakes waiting.

6.9.3.7 void BCondInt::setValue (BInt value)

Set the value. Wakes waiting.

6.9.3.8 BInt BCondInt::value () const

Current value.

6.9.3.9 **Bool** BCondInt::waitLessThan (**BInt** *v*, **BTimeout** *timeoutUs* = **BTimeoutForever**)

Wait until value is equal to or below the value given.

6.9.3.10 **Bool** BCondInt::waitLessThanOrEqual (**BInt** *v*, **Bool** *increment* = 0, **BTimeout** *timeoutUs* = **BTimeoutForever**)

Wait until value is equal to or below the value given.

6.9.3.11 **Bool** BCondInt::waitMoreThanOrEqual (**BInt** *v*, **Bool** *decrement* = 0, **BTimeout** *timeoutUs* = **BTimeoutForever**)

Wait until value is at least the value given.

6.9.4 Member Data Documentation

6.9.4.1 **pthread_cond_t** BCondInt::ocond [private]

6.9.4.2 **pthread_mutex_t** BCondInt::omutex [private]

6.9.4.3 **BInt** BCondInt::ovalue [private]

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

- [BCondInt.h](#)
- [BCondInt.cpp](#)

6.10 BCondResource Class Reference

Resource lock.

```
#include <BCondInt.h>
```

Public Member Functions

- [BCondResource](#) ()
- [~BCondResource](#) ()
- [int lock](#) (uint32_t timeOutUs=0)
Lock the resource, will wait for all usage to be 0.
- [int unlock](#) ()
Unlock the resource.
- [int start](#) (uint32_t timeOutUs=0)
Start using the resource.
- [int end](#) ()
Finish using the resource.
- [int locked](#) ()
- [int inUse](#) ()

Private Attributes

- **pthread_mutex_t** [omutex](#)
- **pthread_cond_t** [ocond](#)
- **int** [olock](#)
- **int** [ouse](#)

6.10.1 Detailed Description

Resource lock.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 `BCondResource::BCondResource ()`

6.10.2.2 `BCondResource::~~BCondResource ()`

6.10.3 Member Function Documentation

6.10.3.1 `int BCondResource::end ()`

Finish using the resource.

6.10.3.2 `int BCondResource::inUse ()`

6.10.3.3 `int BCondResource::lock (uint32_t timeOutUs = 0)`

Lock the resource, will wait for all usage to be 0.

6.10.3.4 `int BCondResource::locked ()`

6.10.3.5 `int BCondResource::start (uint32_t timeOutUs = 0)`

Start using the resource.

6.10.3.6 `int BCondResource::unlock ()`

Unlock the resource.

6.10.4 Member Data Documentation

6.10.4.1 `pthread_cond_t BCondResource::ocond [private]`

6.10.4.2 `int BCondResource::olock [private]`

6.10.4.3 `pthread_mutex_t BCondResource::omutex [private]`

6.10.4.4 `int BCondResource::ouse [private]`

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

- [BCondInt.h](#)
- [BCondInt.cpp](#)

6.11 BCondValue Class Reference

Thread conditional value.

```
#include <BCondInt.h>
```


Public Member Functions

- [BCondValue](#) ()
- [~BCondValue](#) ()
- void [setValue](#) (int [value](#))
Set the value. Wakes waiting.
- int [value](#) ()
Current value.
- int [increment](#) (int [v](#)=1)
Increment. Wakes waiting.
- int [decrement](#) (int [v](#)=1)
Decrement. Wakes waiting.
- int [waitMoreThanOrEqual](#) (int [v](#), int [decrement](#)=0, int [timeOutUs](#)=0)
Wait until value is at least the value given.
- int [waitLessThanOrEqual](#) (int [v](#), int [increment](#)=0, int [timeOutUs](#)=0)
Wait until value is equal to or below the value given.
- int [waitLessThan](#) (int [v](#), int [timeOutUs](#)=0)
Wait until value is equal to or below the value given.
- void [operator+=](#) (int [v](#))
Add to value. Wakes waiting.
- void [operator-=](#) (int [v](#))
Subtract from value. Wakes waiting.
- void [operator++](#) ()
Increment value. Wakes waiting.
- void [operator--](#) ()
Decrement value. Wakes waiting.

Private Attributes

- pthread_mutex_t [omutex](#)
- pthread_cond_t [ocond](#)
- int [ovalue](#)

6.11.1 Detailed Description

Thread conditional value.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 [BCondValue::BCondValue](#) ()

6.11.2.2 [BCondValue::~~BCondValue](#) ()

6.11.3 Member Function Documentation

6.11.3.1 [int BCondValue::decrement](#) (int [v](#) = 1)

Decrement. Wakes waiting.

6.11.3.2 [int BCondValue::increment](#) (int [v](#) = 1)

Increment. Wakes waiting.

6.11.3.3 `void BCondValue::operator++ (int) [inline]`

Increment value. Wakes waiting.

6.11.3.4 `void BCondValue::operator+=(int v) [inline]`

Add to value. Wakes waiting.

6.11.3.5 `void BCondValue::operator-- (int) [inline]`

Decrement value. Wakes waiting.

6.11.3.6 `void BCondValue::operator-= (int v) [inline]`

Subtract from value. Wakes waiting.

6.11.3.7 `void BCondValue::setValue (int value)`

Set the value. Wakes waiting.

6.11.3.8 `int BCondValue::value ()`

Current value.

6.11.3.9 `int BCondValue::waitLessThan (int v, int timeOutUs = 0)`

Wait until value is equal to or below the value given.

6.11.3.10 `int BCondValue::waitLessThanOrEqualTo (int v, int increment = 0, int timeOutUs = 0)`

Wait until value is equal to or below the value given.

6.11.3.11 `int BCondValue::waitMoreThanOrEqualTo (int v, int decrement = 0, int timeOutUs = 0)`

Wait until value is at least the value given.

6.11.4 Member Data Documentation

6.11.4.1 `pthread_cond_t BCondValue::ocond [private]`

6.11.4.2 `pthread_mutex_t BCondValue::omutex [private]`

6.11.4.3 `int BCondValue::ovalue [private]`

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

- [BCondInt.h](#)
- [BCondInt.cpp](#)

6.12 BCondWrap Class Reference

```
#include <BCondInt.h>
```

Public Member Functions

- [BCondWrap](#) ()
- [~BCondWrap](#) ()
- void [setValue](#) (uint32_t value)
Set the value. Wakes waiting.
- uint32_t [value](#) ()
Current value.
- uint32_t [increment](#) (uint32_t v=1)
Increment. Wakes waiting.
- uint32_t [decrement](#) (uint32_t v=1)
Decrement. Wakes waiting.
- int [waitMoreThanOrEqual](#) (uint32_t v, uint32_t [decrement](#)=0, uint32_t timeOutUs=0)
Wait until value is at least the value given.
- int [waitLessThanOrEqual](#) (uint32_t v, uint32_t [increment](#)=0, uint32_t timeOutUs=0)
Wait until value is equal to or below the value given.
- int [waitLessThan](#) (uint32_t v, uint32_t timeOutUs=0)
Wait until value is equal to or below the value given.
- void [operator+=](#) (int v)
Add to value. Wakes waiting.
- void [operator-=](#) (int v)
Subtract from value. Wakes waiting.
- void [operator++](#) (int)
Increment value. Wakes waiting.
- void [operator--](#) (int)
Decrement value. Wakes waiting.

Private Member Functions

- int [diff](#) (uint32_t v)

Private Attributes

- pthread_mutex_t [omutex](#)
- pthread_cond_t [ocond](#)
- uint32_t [ovalue](#)

6.12.1 Constructor & Destructor Documentation

6.12.1.1 [BCondWrap::BCondWrap](#) ()

6.12.1.2 [BCondWrap::~~BCondWrap](#) ()

6.12.2 Member Function Documentation

6.12.2.1 [uint32_t BCondWrap::decrement](#) (uint32_t v = 1)

Decrement. Wakes waiting.

6.12.2.2 `int BCondWrap::diff (uint32_t v)` `[private]`

6.12.2.3 `uint32_t BCondWrap::increment (uint32_t v = 1)`

Increment. Wakes waiting.

6.12.2.4 `void BCondWrap::operator++ (int)` `[inline]`

Increment value. Wakes waiting.

6.12.2.5 `void BCondWrap::operator+= (int v)` `[inline]`

Add to value. Wakes waiting.

6.12.2.6 `void BCondWrap::operator-- (int)` `[inline]`

Decrement value. Wakes waiting.

6.12.2.7 `void BCondWrap::operator-= (int v)` `[inline]`

Subtract from value. Wakes waiting.

6.12.2.8 `void BCondWrap::setValue (uint32_t value)`

Set the value. Wakes waiting.

6.12.2.9 `uint32_t BCondWrap::value ()`

Current value.

6.12.2.10 `int BCondWrap::waitLessThan (uint32_t v, uint32_t timeoutUs = 0)`

Wait until value is equal to or below the value given.

6.12.2.11 `int BCondWrap::waitLessThanOrEqualTo (uint32_t v, uint32_t increment = 0, uint32_t timeoutUs = 0)`

Wait until value is equal to or below the value given.

6.12.2.12 `int BCondWrap::waitMoreThanOrEqualTo (uint32_t v, uint32_t decrement = 0, uint32_t timeoutUs = 0)`

Wait until value is at least the value given.

6.12.3 Member Data Documentation

6.12.3.1 `pthread_cond_t BCondWrap::ocond` `[private]`

6.12.3.2 `pthread_mutex_t BCondWrap::omutex` `[private]`

6.12.3.3 `uint32_t BCondWrap::ovalue` `[private]`

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

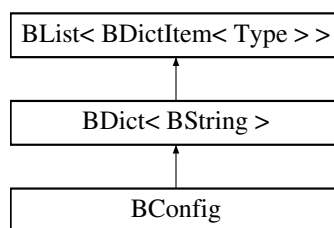
- [BCondInt.h](#)
- [BCondInt.cpp](#)

6.13 BConfig Class Reference

This class implements the configuration file access.

```
#include <BConfig.h>
```

Inheritance diagram for BConfig:



Public Member Functions

- [BError open](#) ([BString fileName](#), [BString mode](#)="r")
- `void close ()`
- [BError read](#) ()
- [BError write](#) ()
- [BString findValue](#) ([BString name](#))
- [BString fileName](#) ()

Private Attributes

- [BMutex olock](#)
- [BString ofileName](#)
- [BFile ofile](#)

Additional Inherited Members

6.13.1 Detailed Description

This class implements the configuration file access.

6.13.2 Member Function Documentation

6.13.2.1 `void BConfig::close ()`

6.13.2.2 `BString BConfig::fileName ()`

6.13.2.3 `BString BConfig::findValue (BString name)`

6.13.2.4 **BError** BConfig::open (**BString** *fileName*, **BString** *mode* = "r")

6.13.2.5 **BError** BConfig::read ()

6.13.2.6 **BError** BConfig::write ()

6.13.3 Member Data Documentation

6.13.3.1 **BFile** BConfig::ofile [private]

6.13.3.2 **BString** BConfig::ofilename [private]

6.13.3.3 **BMutex** BConfig::olock [private]

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

- [BConfig.h](#)
- [BConfig.cpp](#)

6.14 BDate Class Reference

```
#include <BDate.h>
```

Public Member Functions

- **BDate** (int *year*=0, int *month*=1, int *day*=1)
- **BDate** (**BString** *str*)
- **~BDate** ()
- void **clear** ()
Clear the date/time.
- void **setFirst** ()
Set the first date available.
- void **setLast** ()
Set the last date available.
- void **set** (time_t *time*)
Set time using Unix time (seconds from 1970-01-01)
- void **set** (int *year*=0, int *month*=1, int *day*=1)
- void **setYDay** (int *year*=0, int *yday*=0)
- void **setNow** ()
Set the timeStamp to now.
- int **year** ()
- int **yday** ()
- int **month** ()
- int **day** ()
- void **getDate** (int &*year*, int &*mon*, int &*day*)
- **BString** **getString** ()
Get the time as an ISO date/time string.
- **BString** **getStringFormatted** (**BString** *format*)
Gets the time in a string form as per the format. Format syntax as per strftime()
- **BError** **setString** (**BString** *str*)
Set the time from an ISO date/time.
- int **isSet** ()

- int `compare` (const `BDate` &date) const
Compare two dates.
- `operator BString` ()
- int `operator==` (const `BDate` &date) const
- int `operator!=` (const `BDate` &date) const
- int `operator>` (const `BDate` &date) const
- int `operator>=` (const `BDate` &date) const
- int `operator<` (const `BDate` &date) const
- int `operator<=` (const `BDate` &date) const

Static Public Member Functions

- static int `isLeap` (int `year`)
- static int `daysInMonth` (int `year`, int `month`)

Public Attributes

- uint16_t `oyear`
Year (0 .. 65535)
- uint16_t `oyday`
Day in year (0 .. 365)

6.14.1 Constructor & Destructor Documentation

6.14.1.1 `BDate::BDate (int year = 0, int month = 1, int day = 1)`

6.14.1.2 `BDate::BDate (BString str)`

6.14.1.3 `BDate::~~BDate ()`

6.14.2 Member Function Documentation

6.14.2.1 `void BDate::clear ()`

Clear the date/time.

6.14.2.2 `int BDate::compare (const BDate & date) const`

Compare two dates.

6.14.2.3 `int BDate::day ()`

6.14.2.4 `int BDate::daysInMonth (int year, int month)` `[static]`

6.14.2.5 `void BDate::getDate (int & year, int & mon, int & day)`

6.14.2.6 `BString BDate::getString ()`

Get the time as an ISO date/time string.

6.14.2.7 BString BDate::getStringFormatted (BString *format*)

Gets the time in a string form as per the format. Format syntax as per strftime()

6.14.2.8 int BDate::isLeap (int *year*) [static]

6.14.2.9 int BDate::isSet () [inline]

6.14.2.10 int BDate::month ()

6.14.2.11 BDate::operator BString () [inline]

6.14.2.12 int BDate::operator!= (const BDate & *date*) const [inline]

6.14.2.13 int BDate::operator< (const BDate & *date*) const [inline]

6.14.2.14 int BDate::operator<= (const BDate & *date*) const [inline]

6.14.2.15 int BDate::operator== (const BDate & *date*) const [inline]

6.14.2.16 int BDate::operator> (const BDate & *date*) const [inline]

6.14.2.17 int BDate::operator>= (const BDate & *date*) const [inline]

6.14.2.18 void BDate::set (time_t *time*)

Set time using Unix time (seconds from 1970-01-01)

6.14.2.19 void BDate::set (int *year* = 0, int *month* = 1, int *day* = 1)

6.14.2.20 void BDate::setFirst ()

Set the first date available.

6.14.2.21 void BDate::setLast ()

Set the last date available.

6.14.2.22 void BDate::setNow ()

Set the timeStamp to now.

6.14.2.23 BError BDate::setString (BString *str*)

Set the time from an ISO date/time.

6.14.2.24 void BDate::setYDay (int *year* = 0, int *yday* = 0)

6.14.2.25 int BDate::yday ()

6.14.2.26 int BDate::year ()

6.14.3 Member Data Documentation

6.14.3.1 uint16_t BDate::oyday

Day in year (0 .. 365)

6.14.3.2 uint16_t BDate::oyear

Year (0 .. 65535)

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

- [BDate.h](#)
- [BDate.cpp](#)

6.15 BDebugBacktrace Class Reference

```
#include <BDebug.h>
```

Public Member Functions

- [BDebugBacktrace](#) ()
- [~BDebugBacktrace](#) ()
- void [dumpBacktraceStdout](#) (char *comment)
- int [dumpBacktraceFile](#) (char *fileName, char *comment)
- void [dumpBacktraceSyslog](#) (char *comment)
- void [dumpBacktrace](#) (char *strBuf, int strBufLen, char *comment)

6.15.1 Constructor & Destructor Documentation

6.15.1.1 BDebugBacktrace::BDebugBacktrace ()

6.15.1.2 BDebugBacktrace::~~BDebugBacktrace ()

6.15.2 Member Function Documentation

6.15.2.1 void BDebugBacktrace::dumpBacktrace (char * *strBuf*, int *strBufLen*, char * *comment*)

6.15.2.2 int BDebugBacktrace::dumpBacktraceFile (char * *fileName*, char * *comment*)

6.15.2.3 void BDebugBacktrace::dumpBacktraceStdout (char * *comment*)

6.15.2.4 void BDebugBacktrace::dumpBacktraceSyslog (char * *comment*)

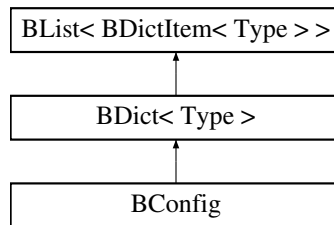
The documentation for this class was generated from the following files:

- [BDebug.h](#)
- [BDebug.cpp](#)

6.16 BDict< Type > Class Template Reference

```
#include <BDict.h>
```

Inheritance diagram for BDict< Type >:



Public Types

- typedef [Blter](#) iterator

Public Member Functions

- [BDict](#) (int hashSize=100)
- [BDict](#) (const [BDict](#)< Type > &dict)
- int [hasKey](#) (const [BString](#) &k) const
- [BString](#) [key](#) (const [Blter](#) &i) const
- void [clear](#) ()
Clear the list.
- void [insert](#) ([Blter](#) &i, const [BDictItem](#)< Type > &item)
Insert item before item.
- void [append](#) (const [BDictItem](#)< Type > &item)
- void [append](#) (const [BDict](#)< Type > &dict)
- void [del](#) (const [BString](#) &k)
- void [del](#) ([Blter](#) &i)
Delete specified item.
- [Blter](#) [find](#) (const [BString](#) &k) const
- Type & [operator\[\]](#) (const [BString](#) &i)
- Type & [operator\[\]](#) (const [Blter](#) &i)
- const Type & [operator\[\]](#) (const [Blter](#) &i) const
- [BDict](#)< Type > [operator+](#) (const [BDict](#)< Type > &dict) const
- [BDict](#)< Type > & [operator=](#) (const [BDict](#)< Type > &dict)
- void [hashPrint](#) ()

Private Member Functions

- void [hashAdd](#) (const [BString](#) &k, [Blter](#) iter)
- void [hashDelete](#) (const [BString](#) &k, [Blter](#) iter)
- int [hashFind](#) (const [BString](#) &k, [Blter](#) &iter) const

Private Attributes

- int [ohashSize](#)
- [BArray](#)< [BList](#)< [Blter](#) > > [ohashLists](#)

Additional Inherited Members

6.16.1 Member Typedef Documentation

6.16.1.1 `template<class Type > typedef BIter BDict< Type >::iterator`

6.16.2 Constructor & Destructor Documentation

6.16.2.1 `template<class Type > BDict< Type >::BDict (int hashSize = 100)`

6.16.2.2 `template<class Type > BDict< Type >::BDict (const BDict< Type > & dict)`

6.16.3 Member Function Documentation

6.16.3.1 `template<class Type > void BDict< Type >::append (const BDictItem< Type > & item)`

6.16.3.2 `template<class Type > void BDict< Type >::append (const BDict< Type > & dict)`

6.16.3.3 `template<class Type > void BDict< Type >::clear () [virtual]`

Clear the list.

Reimplemented from [BList< BDictItem< Type > >](#).

6.16.3.4 `template<class Type > void BDict< Type >::del (const BString & k)`

6.16.3.5 `template<class Type > void BDict< Type >::del (BIter & i) [virtual]`

Delete specified item.

Reimplemented from [BList< BDictItem< Type > >](#).

6.16.3.6 `template<class Type > BIter BDict< Type >::find (const BString & k) const`

6.16.3.7 `template<class Type > void BDict< Type >::hashAdd (const BString & k, BIter iter) [private]`

6.16.3.8 `template<class Type > void BDict< Type >::hashDelete (const BString & k, BIter iter) [private]`

6.16.3.9 `template<class Type > int BDict< Type >::hashFind (const BString & k, BIter & iter) const [private]`

6.16.3.10 `template<class Type > void BDict< Type >::hashPrint ()`

6.16.3.11 `template<class Type > int BDict< Type >::hasKey (const BString & k) const`

6.16.3.12 `template<class Type > void BDict< Type >::insert (BIter & i, const BDictItem< Type > & item) [virtual]`

Insert item before item.

Reimplemented from [BList< BDictItem< Type > >](#).

6.16.3.13 `template<class Type > BString BDict< Type >::key (const BIter & i) const`

6.16.3.14 `template<class Type > BDict< Type > BDict< Type >::operator+ (const BDict< Type > & dict) const`

6.16.3.15 `template<class Type> BDict< Type> & BDict< Type>::operator=(const BDict< Type> & dict)`

6.16.3.16 `template<class Type> Type & BDict< Type>::operator[] (const BString & i)`

6.16.3.17 `template<class Type> Type & BDict< Type>::operator[] (const Blter & i)`

6.16.3.18 `template<class Type> const Type & BDict< Type>::operator[] (const Blter & i) const`

6.16.4 Member Data Documentation

6.16.4.1 `template<class Type> BArray<BList<Blter>> BDict< Type>::ohashLists [private]`

6.16.4.2 `template<class Type> int BDict< Type>::ohashSize [private]`

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

- [BDict.h](#)

6.17 BDictItem< Type> Class Template Reference

Template based Dictionary class.

```
#include <BDict.h>
```

Public Member Functions

- [BDictItem](#) (BString k="", Type v=Type())

Public Attributes

- BString [key](#)
- Type [value](#)

6.17.1 Detailed Description

```
template<class Type> class BDictItem< Type>
```

Template based Dictionary class.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 `template<class Type> BDictItem< Type>::BDictItem (BString k = " ", Type v = Type()) [inline]`

6.17.3 Member Data Documentation

6.17.3.1 `template<class Type> BString BDictItem< Type>::key`

6.17.3.2 `template<class Type> Type BDictItem< Type>::value`

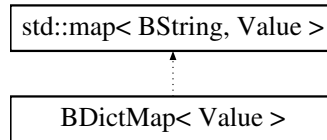
The documentation for this class was generated from the following file:

- [BDict.h](#)

6.18 BDictMap< Value > Class Template Reference

```
#include <BDictMap.h>
```

Inheritance diagram for BDictMap< Value >:



Public Types

- typedef [BDictMap< Value >::iterator](#) [iterator](#)

Public Member Functions

- void [clear](#) ()
- int [hasKey](#) (const [BString](#) &k)
- [BString](#) [key](#) ([iterator](#) &i)
- unsigned int [size](#) ()
- void [start](#) ([iterator](#) &i)
- int [isEnd](#) ([iterator](#) &i)
- void [next](#) ([iterator](#) &i)
- void [del](#) (const [iterator](#) &i)
- void [del](#) (const [BString](#) &k)
- Value & [operator\[\]](#) ([iterator](#) &i)
- Value & [operator\[\]](#) (const [BString](#) &i)

6.18.1 Detailed Description

```
template<typename Value>class BDictMap< Value >
```

Template based Array class. This is based on the Standard C++ library map class and has all of the functionality of that class.

6.18.2 Member Typedef Documentation

6.18.2.1 `template<typename Value > typedef BDictMap<Value>::iterator BDictMap< Value >::iterator`

6.18.3 Member Function Documentation

6.18.3.1 `template<typename Value > void BDictMap< Value >::clear () [inline]`

6.18.3.2 `template<typename Value > void BDictMap< Value >::del (const iterator & i) [inline]`

6.18.3.3 `template<typename Value > void BDictMap< Value >::del (const BString & k) [inline]`

6.18.3.4 `template<typename Value > int BDictMap< Value >::hasKey (const BString & k) [inline]`

6.18.3.5 `template<typename Value > int BDictMap< Value >::isEnd (iterator & i) [inline]`

6.18.3.6 `template<typename Value > BString BDictMap< Value >::key (iterator & i)` `[inline]`

6.18.3.7 `template<typename Value > void BDictMap< Value >::next (iterator & i)` `[inline]`

6.18.3.8 `template<typename Value > Value& BDictMap< Value >::operator[] (iterator & i)` `[inline]`

6.18.3.9 `template<typename Value > Value& BDictMap< Value >::operator[] (const BString & i)` `[inline]`

6.18.3.10 `template<typename Value > unsigned int BDictMap< Value >::size ()` `[inline]`

6.18.3.11 `template<typename Value > void BDictMap< Value >::start (iterator & i)` `[inline]`

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

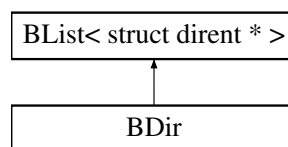
- [BDictMap.h](#)

6.19 BDir Class Reference

File system directory class.

```
#include <BDir.h>
```

Inheritance diagram for BDir:



Public Member Functions

- [BDir](#) ()
- [BDir](#) (BString name)
- [~BDir](#) ()
- [BError open](#) (BString name)
Reads named directory.
- [BError error](#) ()
Current value of error.
- [BError read](#) ()
read/re-reads directory
- void [clear](#) ()
Clears list.
- void [setWild](#) (BString wild)
Set wildcard filter string used on read.
- void [setSort](#) (int on)
Set alpha sort on/off.
- [BString entryName](#) (BIter i)
Get filename.
- struct stat [entryStat](#) (BIter i)
Get file stats.
- struct stat64 [entryStat64](#) (BIter i)
Get file stats 64.

Private Attributes

- [BError oerror](#)
- [BString odirname](#)
- [BString owild](#)
- [int osort](#)

Additional Inherited Members

6.19.1 Detailed Description

File system directory class.

6.19.2 Constructor & Destructor Documentation

6.19.2.1 `BDir::BDir ()`

6.19.2.2 `BDir::BDir (BString name)`

6.19.2.3 `BDir::~~BDir ()`

6.19.3 Member Function Documentation

6.19.3.1 `void BDir::clear ()` [virtual]

Clears list.

Reimplemented from [BList< struct dirent * >](#).

6.19.3.2 `BString BDir::entryName (BIter i)`

Get filename.

6.19.3.3 `struct stat BDir::entryStat (BIter i)`

Get file stats.

6.19.3.4 `struct stat64 BDir::entryStat64 (BIter i)`

Get file stats 64.

6.19.3.5 `BError BDir::error ()`

Current value of error.

6.19.3.6 `BError BDir::open (BString name)`

Reads named directory.

6.19.3.7 `BError BDir::read ()`

read/re-reads directory

6.19.3.8 void BDir::setSort (int on)

Set alpha sort on/off.

6.19.3.9 void BDir::setWild (BString wild)

Set wildcard filter string used on read.

6.19.4 Member Data Documentation

6.19.4.1 BString BDir::odirname [private]

6.19.4.2 BError BDir::oerror [private]

6.19.4.3 int BDir::osort [private]

6.19.4.4 BString BDir::owild [private]

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

- [BDir.h](#)
- [BDir.cpp](#)

6.20 BDuration Class Reference

```
#include <BDuration.h>
```

Public Member Functions

- [BDuration](#) (int hour=0, int minute=0, int second=0, int microsecond=0)
- [BDuration](#) (BString str)
- [~BDuration](#) ()
- void [clear](#) ()
Clear the duration.
- void [set](#) (int hour=0, int minute=0, int second=0, int microsecond=0)
- void [addMilliSeconds](#) (int64_t milliSeconds)
Add the given number of milli seconds.
- void [addMicroSeconds](#) (int64_t microSeconds)
Add the given number of micro seconds.
- void [addSeconds](#) (int seconds)
Add the given number of seconds.
- uint32_t [getSeconds](#) ()
Get number of seconds.
- uint64_t [getMicroSeconds](#) ()
Get number of micro seconds.
- int [hour](#) ()
- int [minute](#) ()
- int [second](#) ()
- int [microSecond](#) ()
- BString [getString](#) ()
Get the time as an ISO date/time string.
- BError [setString](#) (BString time)
Set the time from an ISO date/time.

Private Attributes

- `uint8_t ohour`
Hour (0 .. 23)
- `uint8_t ominute`
Minute (0 .. 59)
- `uint8_t osecond`
Second (0 .. 59)
- `uint8_t ospare`
Padding.
- `uint32_t omicroSecond`
MicroSecond (0 .. 999999)

6.20.1 Constructor & Destructor Documentation

6.20.1.1 `BDuration::BDuration (int hour = 0, int minute = 0, int second = 0, int microsecond = 0)`

6.20.1.2 `BDuration::BDuration (BString str)`

6.20.1.3 `BDuration::~~BDuration ()`

6.20.2 Member Function Documentation

6.20.2.1 `void BDuration::addMicroSeconds (int64_t microSeconds)`

Add the given number of micro seconds.

6.20.2.2 `void BDuration::addMilliSeconds (int64_t milliSeconds)`

Add the given number of milli seconds.

6.20.2.3 `void BDuration::addSeconds (int seconds)`

Add the given number of seconds.

6.20.2.4 `void BDuration::clear ()`

Clear the duration.

6.20.2.5 `uint64_t BDuration::getMicroSeconds ()`

Get number of micro seconds.

6.20.2.6 `uint32_t BDuration::getSeconds ()`

Get number of seconds.

6.20.2.7 `BString BDuration::getString ()`

Get the time as an ISO date/time string.

6.20.2.8 `int BDuration::hour ()`

6.20.2.9 `int BDuration::microSecond ()`

6.20.2.10 `int BDuration::minute ()`

6.20.2.11 `int BDuration::second ()`

6.20.2.12 `void BDuration::set (int hour = 0, int minute = 0, int second = 0, int microsecond = 0)`

6.20.2.13 `BError BDuration::setString (BString time)`

Set the time from an ISO date/time.

6.20.3 Member Data Documentation

6.20.3.1 `uint8_t BDuration::ohour [private]`

Hour (0 .. 23)

6.20.3.2 `uint32_t BDuration::omicroSecond [private]`

MicroSecond (0 .. 999999)

6.20.3.3 `uint8_t BDuration::ominate [private]`

Minute (0 .. 59)

6.20.3.4 `uint8_t BDuration::osecond [private]`

Second (0 .. 59)

6.20.3.5 `uint8_t BDuration::ospare [private]`

Padding.

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

- [BDuration.h](#)
- [BDuration.cpp](#)

6.21 BEntry Class Reference

Manipulate a name value pair.

```
#include <BEntry.h>
```

Public Member Functions

- [BEntry \(\)](#)
- [BEntry \(BString name, BString value\)](#)
Set name and value.

- [BEntry \(BString line\)](#)
Set name and value from white space delimited string.
- [BString getName \(\)](#)
Get the name.
- [BString getValue \(\)](#)
Get the value.
- void [setLine \(BString line\)](#)
Set name and value from white space delimited string.
- void [setName \(BString name\)](#)
Set the name.
- void [setValue \(BString value\)](#)
Set the value.
- [BString line \(\)](#)
Return name and value as padded single string.
- void [print \(\)](#)
Print name and value.

Private Attributes

- [BString oname](#)
- [BString ovalue](#)

6.21.1 Detailed Description

Manipulate a name value pair.

6.21.2 Constructor & Destructor Documentation

6.21.2.1 BEntry::BEntry ()

6.21.2.2 BEntry::BEntry (BString name, BString value)

Set name and value.

6.21.2.3 BEntry::BEntry (BString line)

Set name and value from white space delimited string.

6.21.3 Member Function Documentation

6.21.3.1 BString BEntry::getName ()

Get the name.

6.21.3.2 BString BEntry::getValue ()

Get the value.

6.21.3.3 BString BEntry::line ()

Return name and value as padded single string.

6.21.3.4 void BEntry::print ()

Print name and value.

6.21.3.5 void BEntry::setLine (BString line)

Set name and value from white space delimited string.

6.21.3.6 void BEntry::setName (BString name)

Set the name.

6.21.3.7 void BEntry::setValue (BString value)

Set the value.

6.21.4 Member Data Documentation**6.21.4.1 BString BEntry::oname [private]****6.21.4.2 BString BEntry::ovalue [private]**

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

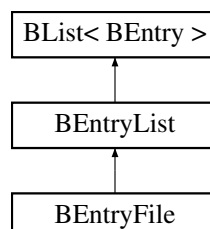
- [BEntry.h](#)
- [BEntry.cpp](#)

6.22 BEntryFile Class Reference

File of Entries.

```
#include <BEntry.h>
```

Inheritance diagram for BEntryFile:

**Public Member Functions**

- [BEntryFile \(\)](#)
- [BEntryFile \(BString filename\)](#)

- *Opens entryfile.*
- [~BEntryFile](#) ()
- [int open](#) (BString filename)
- *Opens entryfile.*
- [int read](#) ()
- *Reads entry file and builds list.*
- [int write](#) ()
- *Writes list to entryfile.*
- [int writeList](#) (BEntryList &l)
- *Writes specified list to file.*
- [void clear](#) ()
- *Clears current list.*
- [BString filename](#) ()
- *Returns the filename.*

Private Attributes

- [BString ofilename](#)
- [BString ocomments](#)

Additional Inherited Members

6.22.1 Detailed Description

File of Entries.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 [BEntryFile::BEntryFile](#) ()

6.22.2.2 [BEntryFile::BEntryFile](#) (BString filename)

Opens entryfile.

6.22.2.3 [BEntryFile::~~BEntryFile](#) ()

6.22.3 Member Function Documentation

6.22.3.1 [void BEntryFile::clear](#) () [virtual]

Clears current list.

Reimplemented from [BEntryList](#).

6.22.3.2 [BString BEntryFile::filename](#) ()

Returns the filename.

6.22.3.3 [int BEntryFile::open](#) (BString filename)

Opens entryfile.

6.22.3.4 `int BEntryFile::read ()`

Reads entry file and builds list.

6.22.3.5 `int BEntryFile::write ()`

Writes list to entryfile.

6.22.3.6 `int BEntryFile::writeList (BEntryList & l)`

Writes specified list to file.

6.22.4 Member Data Documentation

6.22.4.1 `BString BEntryFile::ocomments` [private]

6.22.4.2 `BString BEntryFile::ofilename` [private]

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

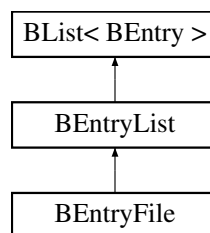
- [BEntry.h](#)
- [BEntry.cpp](#)

6.23 BEntryList Class Reference

List of Entries. Where an entry is a name value pair.

```
#include <BEntry.h>
```

Inheritance diagram for BEntryList:



Public Member Functions

- [BEntryList \(\)](#)
- `int isSet (BString name)`
1 if name is in list and value is set
- `BEntry * find (BString name)`
Returns entry if name is found otherwise NULL.
- `BString findValue (BString name)`
Returns value of name. Returns "" if name not found.
- `int setValue (BString name, BString value)`
Set the value of name. Returns 0 if name not found.
- `int setValueRaw (BString name, BString value)`

- *Raw setting of value without looking up existing entry.*
- void [deleteEntry](#) (BString name)
- *Deletes the entry.*
- void [print](#) ()
- *Print list.*
- BString [getString](#) ()
- *Return list as string. Each Entry padded and on a new line.*
- void [insert](#) (BIter &i, const BEntry &item)
- *Insert item before item.*
- void [del](#) (BIter &i)
- *Delete specified item.*
- void [clear](#) ()
- *Clear the list.*
- BEntryList & [operator=](#) (const BEntryList &l)

Private Attributes

- [BIter](#) [olastPos](#)

Additional Inherited Members

6.23.1 Detailed Description

List of Entries. Where an entry is a name value pair.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 BEntryList::BEntryList ()

6.23.3 Member Function Documentation

6.23.3.1 void BEntryList::clear () [virtual]

Clear the list.

Reimplemented from [BList< BEntry >](#).

Reimplemented in [BEntryFile](#).

6.23.3.2 void BEntryList::del (BIter & i) [virtual]

Delete specified item.

Reimplemented from [BList< BEntry >](#).

6.23.3.3 void BEntryList::deleteEntry (BString name)

Deletes the entry.

6.23.3.4 BEntry * BEntryList::find (BString name)

Returns entry if name is found otherwise NULL.

6.23.3.5 BString BEntryList::findValue (BString *name*)

Returns value of name. Returns "" if name not found.

6.23.3.6 BString BEntryList::getString ()

Return list as string. Each Entry padded and on a new line.

6.23.3.7 void BEntryList::insert (BIter & *i*, const BEntry & *item*) [virtual]

Insert item before item.

Reimplemented from [BList< BEntry >](#).

6.23.3.8 int BEntryList::isSet (BString *name*)

1 if name is in list and value is set

6.23.3.9 BEntryList & BEntryList::operator= (const BEntryList & *I*)

6.23.3.10 void BEntryList::print ()

Print list.

6.23.3.11 int BEntryList::setValue (BString *name*, BString *value*)

Set the value of name. Returns 0 if name not found.

6.23.3.12 int BEntryList::setValueRaw (BString *name*, BString *value*)

Raw setting of value without looking up existing entry.

6.23.4 Member Data Documentation

6.23.4.1 BIter BEntryList::olastPos [private]

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

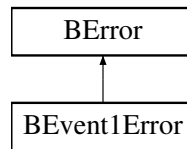
- [BEntry.h](#)
- [BEntry.cpp](#)

6.24 BError Class Reference

Error return class.

```
#include <BError.h>
```

Inheritance diagram for BError:



Public Member Functions

- **BError** (int errNo=**ErrorOk**, **BString** errStr="")
Create object.
- **BError** (**BString** errStr)
Create with error set and error string.
- **BError** copy ()
Return an independant copy.
- **BError** & **set** (int errNo, **BString** errStr="")
Set error number and message.
- **BError** & **clear** ()
Clear the error.
- **BError** & **setError** (**BString** errStr="")
Set error type ERROR with optional message.
- **BString** **getString** () const
Get error message.
- int **getNumber** () const
Get The error number.
- int **num** () const
Get The error number.
- const char * **str** () const
Return a char string.*
- int **getErrorNo** () const
Get The error number.
- **operator int** () const
Return error number.

Private Attributes

- int **oerrNo**
- **BString** **oerrStr**

6.24.1 Detailed Description

Error return class.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 BError::BError (int errNo = ErrorOk, BString errStr = " ")

Create object.

6.24.2.2 `BError::BError (BString errStr)`

Create with error set and error string.

6.24.3 Member Function Documentation

6.24.3.1 `BError & BError::clear ()`

Clear the error.

6.24.3.2 `BError BError::copy ()`

Return an independant copy.

6.24.3.3 `int BError::getErrorNo () const`

Get The error number.

6.24.3.4 `int BError::getNumber () const`

Get The error number.

6.24.3.5 `BString BError::getString () const`

Get error message.

6.24.3.6 `int BError::num () const`

Get The error number.

6.24.3.7 `BError::operator int () const` `[inline]`

Return error number.

6.24.3.8 `BError & BError::set (int errNo, BString errStr = " ")`

Set error number and message.

6.24.3.9 `BError & BError::setError (BString errStr = " ")`

Set error type ERROR with optional message.

6.24.3.10 `const char * BError::str () const`

Return a char* string.

6.24.4 Member Data Documentation

6.24.4.1 `int BError::oerrNo` `[private]`

6.24.4.2 `BString BError::oerrStr` `[private]`

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

- [BError.h](#)
- [BError.cpp](#)

6.25 BErrorTime Class Reference

Error return class.

```
#include <BErrorTime.h>
```

Public Types

- enum `Type` { `None` = 0, `Error` = 1 }

Public Member Functions

- `BErrorTime` (int errNo=`None`, `BTimeStamp` errTime=`BTimeStamp`(), `BString` errStr="")
Create object.
- `BErrorTime` & `set` (int errNo, `BTimeStamp` errTime=`BTimeStamp`(), `BString` errStr="")
Set error number and message.
- `BErrorTime` & `clear` ()
Clear the error.
- int `getErrorNo` () const
Get The error number.
- `BTimeStamp` `getTime` () const
Get time.
- `BString` `getString` () const
Get error message.
- `BErrorTime` `copy` ()
Return an independant copy.
- `operator int` () const
Return error number.

Private Attributes

- int `oerrNo`
- `BTimeStamp` `oerrTime`
- `BString` `oerrStr`

6.25.1 Detailed Description

Error return class.

6.25.2 Member Enumeration Documentation

6.25.2.1 enum BErrorTime::Type

Enumerator

None

Error

6.25.3 Constructor & Destructor Documentation

6.25.3.1 BErrorTime::BErrorTime (int *errNo* = None, BTimeStamp *errTime* = BTimeStamp (), BString *errStr* = " ")

Create object.

6.25.4 Member Function Documentation

6.25.4.1 BErrorTime & BErrorTime::clear ()

Clear the error.

6.25.4.2 BErrorTime BErrorTime::copy ()

Return an independant copy.

6.25.4.3 int BErrorTime::getErrorNo () const

Get The error number.

6.25.4.4 BString BErrorTime::getString () const

Get error message.

6.25.4.5 BTimeStamp BErrorTime::getTime () const

Get time.

6.25.4.6 BErrorTime::operator int () const

Return error number.

6.25.4.7 BErrorTime & BErrorTime::set (int *errNo*, BTimeStamp *errTime* = BTimeStamp (), BString *errStr* = " ")

Set error number and message.

6.25.5 Member Data Documentation

6.25.5.1 int BErrorTime::oerrNo [private]

6.25.5.2 BString BErrorTime::oerrStr [private]

6.25.5.3 BTimeStamp BErrorTime::oerrTime [private]

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

- [BErrorTime.h](#)
- [BErrorTime.cpp](#)

6.26 BEvent Class Reference

```
#include <BEvent.h>
```

Public Member Functions

- [BEvent](#) (BUInt32 type=BEventTypeNone, BUInt32 arg=0)
- [BUInt32 type](#) ()
- [BUInt32 arg](#) ()

Private Attributes

- [BUInt32 otype](#)
The events type.
- [BUInt32 oarg](#)
The events argument.

6.26.1 Constructor & Destructor Documentation

6.26.1.1 [BEvent::BEvent](#) (BUInt32 type = BEventTypeNone, BUInt32 arg = 0)

6.26.2 Member Function Documentation

6.26.2.1 [BUInt32 BEvent::arg](#) ()

6.26.2.2 [BUInt32 BEvent::type](#) ()

6.26.3 Member Data Documentation

6.26.3.1 [BUInt32 BEvent::oarg](#) [private]

The events argument.

6.26.3.2 [BUInt32 BEvent::otype](#) [private]

The events type.

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

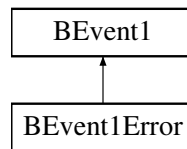
- [BEvent.h](#)
- [BEvent.cpp](#)

6.27 BEvent1 Class Reference

This class provides a base class for all event objects that can be sent over the events interface.

```
#include <BEvent1.h>
```

Inheritance diagram for BEvent1:



Public Member Functions

- [BEvent1](#) (uint32_t type)
- virtual [~BEvent1](#) ()
- uint32_t [getType](#) ()
- virtual [BError getBinary](#) (void *data, uint32_t &size)
- virtual [BError setBinary](#) (void *data, uint32_t &size)

Private Attributes

- uint32_t [otype](#)
The event type.

6.27.1 Detailed Description

This class provides a base class for all event objects that can be sent over the events interface.

6.27.2 Constructor & Destructor Documentation

6.27.2.1 [BEvent1::BEvent1](#) (uint32_t type)

6.27.2.2 [BEvent1::~~BEvent1](#) () [virtual]

6.27.3 Member Function Documentation

6.27.3.1 [BError BEvent1::getBinary](#) (void * data, uint32_t & size) [virtual]

Reimplemented in [BEvent1Error](#).

6.27.3.2 [uint32_t BEvent1::getType](#) ()

6.27.3.3 [BError BEvent1::setBinary](#) (void * data, uint32_t & size) [virtual]

Reimplemented in [BEvent1Error](#).

6.27.4 Member Data Documentation

6.27.4.1 uint32_t BEvent1::otype [private]

The event type.

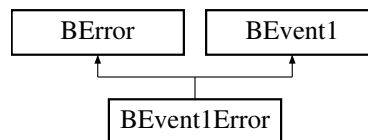
The documentation for this class was generated from the following files:

- [BEvent1.h](#)
- [BEvent1.cpp](#)

6.28 BEvent1Error Class Reference

```
#include <BEvent1.h>
```

Inheritance diagram for BEvent1Error:



Public Member Functions

- [BEvent1Error](#) (int errNo=[ErrorOk](#), [BString](#) errStr="")
- [BError](#) [getBinary](#) (void *data, uint32_t &size)
- [BError](#) [setBinary](#) (void *data, uint32_t &size)

6.28.1 Constructor & Destructor Documentation

6.28.1.1 BEvent1Error::BEvent1Error (int errNo = ErrorOk, BString errStr = " ")

6.28.2 Member Function Documentation

6.28.2.1 BError BEvent1Error::getBinary (void * data, uint32_t & size) [virtual]

Reimplemented from [BEvent1](#).

6.28.2.2 BError BEvent1Error::setBinary (void * data, uint32_t & size) [virtual]

Reimplemented from [BEvent1](#).

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

- [BEvent1.h](#)
- [BEvent1.cpp](#)

6.29 BEvent1Int Class Reference

This class provides an interface for sending simple integer events via a file descriptor. This allows threads to send events that can be picked up by the poll system call.

```
#include <BEvent1.h>
```

Public Member Functions

- [BEvent1Int](#) ()
- [~BEvent1Int](#) ()
- void [clear](#) ()
Clear events pending.
- [BError sendEvent](#) (int event)
Send an event.
- [BError getEvent](#) (int &event, int timeOutUs=-1)
Receive the event.
- int [getFd](#) ()

Private Attributes

- int [ofds](#) [2]
File descriptors for pipe.

6.29.1 Detailed Description

This class provides an interface for sending simple integer events via a file descriptor. This allows threads to send events that can be picked up by the poll system call.

6.29.2 Constructor & Destructor Documentation

6.29.2.1 [BEvent1Int::BEvent1Int](#) ()

6.29.2.2 [BEvent1Int::~~BEvent1Int](#) ()

6.29.3 Member Function Documentation

6.29.3.1 void [BEvent1Int::clear](#) ()

Clear events pending.

6.29.3.2 [BError BEvent1Int::getEvent](#) (int & *event*, int *timeOutUs* = -1)

Receive the event.

6.29.3.3 int [BEvent1Int::getFd](#) ()

6.29.3.4 [BError BEvent1Int::sendEvent](#) (int *event*)

Send an event.

6.29.4 Member Data Documentation

6.29.4.1 int [BEvent1Int::ofds](#)[2] [private]

File descriptors for pipe.

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

- [BEvent1.h](#)
- [BEvent1.cpp](#)

6.30 BEvent1Pipe Class Reference

This class provides a base interface for sending events via a pipe. This allows threads to send events that can be picked up by the poll system call.

```
#include <BEvent1.h>
```

Public Member Functions

- [BEvent1Pipe](#) ()
- [~BEvent1Pipe](#) ()
- void [clear](#) ()
Clear events pending.
- [BError](#) [sendEvent](#) ([BEvent1](#) *event)
Send an event.
- [BError](#) [getEvent](#) ([BEvent1](#) *event, int timeoutUs=-1)
Receive the event.
- int [getReceiveFd](#) ()
returns the receive file descriptor for the poll system call

Private Attributes

- int [ofds](#) [2]
File descriptors for pipe.

6.30.1 Detailed Description

This class provides a base interface for sending events via a pipe. This allows threads to send events that can be picked up by the poll system call.

6.30.2 Constructor & Destructor Documentation

6.30.2.1 [BEvent1Pipe::BEvent1Pipe](#) ()

6.30.2.2 [BEvent1Pipe::~~BEvent1Pipe](#) ()

6.30.3 Member Function Documentation

6.30.3.1 void [BEvent1Pipe::clear](#) ()

Clear events pending.

6.30.3.2 [BError](#) [BEvent1Pipe::getEvent](#) ([BEvent1](#) * event, int timeoutUs = -1)

Receive the event.

6.30.3.3 int BEvent1Pipe::getReceiveFd ()

returns the receive file descriptor for the poll system call

6.30.3.4 BError BEvent1Pipe::sendEvent (BEvent1 * event)

Send an event.

6.30.4 Member Data Documentation

6.30.4.1 int BEvent1Pipe::ofds[2] [private]

File descriptors for pipe.

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

- [BEvent1.h](#)
- [BEvent1.cpp](#)

6.31 BEventPipe Class Reference

This class provides an interface for sending simple integer events via a pipe file descriptor.

```
#include <BEvent.h>
```

Public Member Functions

- [BEventPipe](#) ()
- [~BEventPipe](#) ()
- void [clear](#) ()
Clear events pending.
- int [getFd](#) ()
- [BUInt writeAvailable](#) () const
- [BError write](#) (const [BEvent](#) &event, [BTimeout](#) timeout=[BTimeoutForever](#))
Append an item onto the queue.
- [BUInt readAvailable](#) () const
- [BError read](#) ([BEvent](#) &event, [BTimeout](#) timeout=[BTimeoutForever](#))
Get an item from the queue.

Private Attributes

- int [ofds](#) [2]
File descriptors for pipe.

6.31.1 Detailed Description

This class provides an interface for sending simple integer events via a pipe file descriptor.

6.31.2 Constructor & Destructor Documentation

6.31.2.1 `BEventPipe::BEventPipe ()`

6.31.2.2 `BEventPipe::~~BEventPipe ()`

6.31.3 Member Function Documentation

6.31.3.1 `void BEventPipe::clear ()`

Clear events pending.

6.31.3.2 `int BEventPipe::getFd ()`

6.31.3.3 `BError BEventPipe::read (BEvent & event, BTimeout timeout = BTimeoutForever)`

Get an item from the queue.

6.31.3.4 `BUInt BEventPipe::readAvailable () const`

6.31.3.5 `BError BEventPipe::write (const BEvent & event, BTimeout timeout = BTimeoutForever)`

Append an item onto the queue.

6.31.3.6 `BUInt BEventPipe::writeAvailable () const`

6.31.4 Member Data Documentation

6.31.4.1 `int BEventPipe::ofds[2] [private]`

File descriptors for pipe.

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

- [BEvent.h](#)
- [BEvent.cpp](#)

6.32 BFifo< Type > Class Template Reference

```
#include <BFifo.h>
```

Public Member Functions

- [BFifo \(BUInt size\)](#)
- [~BFifo \(\)](#)
- `void clear ()`
- [BUInt size \(\)](#)
Returns fifo size.
- [BError resize \(BUInt size\)](#)
Resize FIFO, clears it as well.
- [BUInt writeAvailable \(\)](#)
How many items that can be written.

- [BUInt writeAvailableChunk \(\)](#)
How many items that can be written in a chunk.
- [BError write \(const Type v\)](#)
Write a single item.
- [BError write \(const Type *data, BUInt num\)](#)
Write a set of items. Can only write a maximum of [writeAvailableChunk\(\)](#) to save going beyond end of FIFO buffer.
- [Type * writeData \(\)](#)
Returns a pointer to the data.
- [Type * writeData \(BUInt &num\)](#)
Returns a pointer to the data and how many can be written in a chunk.
- [void writeDone \(BUInt num\)](#)
Indicates when write is complete.
- [void writeBackup \(BUInt num\)](#)
Backup, remove num items at end of fifo. Careful, make sure read is not already happening.
- [BUInt readAvailable \(\)](#)
How many items are available to read.
- [BUInt readAvailableChunk \(\)](#)
How many items are available to read in a chunk.
- [Type read \(\)](#)
Read one item.
- [BError read \(Type *data, BUInt num\)](#)
Read a set of items.
- [Type readPos \(BUInt pos\)](#)
Read item at given offset from current read position.
- [Type * readData \(\)](#)
Returns a pointer to the data.
- [Type * readData \(BUInt &num\)](#)
Returns a pointer to the data and how many can be read in a chunk.
- [void readDone \(BUInt num\)](#)
- [Type & operator\[\] \(int pos\)](#)
Direct access to read samples in buffer.

Protected Attributes

- [BMutex olock](#)
- [BUInt osize](#)
The size of the FIFO.
- [Type * odata](#)
FIFO memory buffer.
- [BUInt owritePos](#)
The write pointer.
- [BUInt oreadPos](#)
The read pointer.

6.32.1 Constructor & Destructor Documentation

6.32.1.1 `template<class Type> BFifo< Type >::BFifo (BUInt size)`

6.32.1.2 `template<class Type> BFifo< Type >::~~BFifo ()`

6.32.2 Member Function Documentation

6.32.2.1 `template<class Type> void BFifo< Type >::clear ()`

6.32.2.2 `template<class Type> Type& BFifo< Type >::operator[] (int pos)`

Direct access to read samples in buffer.

6.32.2.3 `template<class Type> Type BFifo< Type >::read ()`

Read one item.

6.32.2.4 `template<class Type> BError BFifo< Type >::read (Type * data, BUInt num)`

Read a set of items.

6.32.2.5 `template<class Type> BUInt BFifo< Type >::readAvailable ()`

How many items are available to read.

6.32.2.6 `template<class Type> BUInt BFifo< Type >::readAvailableChunk ()`

How many items are available to read in a chunk.

6.32.2.7 `template<class Type> Type* BFifo< Type >::readData ()`

Returns a pointer to the data.

6.32.2.8 `template<class Type> Type* BFifo< Type >::readData (BUInt & num)`

Returns a pointer to the data and how many can be read in a chunk.

6.32.2.9 `template<class Type> void BFifo< Type >::readDone (BUInt num)`

6.32.2.10 `template<class Type> Type BFifo< Type >::readPos (BUInt pos)`

Read item at given offset from current read position.

6.32.2.11 `template<class Type> BError BFifo< Type >::resize (BUInt size)`

Resize FIFO, clears it as well.

6.32.2.12 `template<class Type> BUInt BFifo< Type >::size ()`

Returns fifo size.

6.32.2.13 `template<class Type> BError BFifo< Type >::write (const Type v)`

Write a single item.

6.32.2.14 `template<class Type> BError BFifo< Type >::write (const Type * data, BUInt num)`

Write a set of items. Can only write a maximum of `writeAvailableChunk()` to save going beyond end of FIFO buffer.

6.32.2.15 `template<class Type> BUInt BFifo< Type >::writeAvailable ()`

How many items that can be written.

6.32.2.16 `template<class Type> BUInt BFifo< Type >::writeAvailableChunk ()`

How many items that can be written in a chunk.

6.32.2.17 `template<class Type> void BFifo< Type >::writeBackup (BUInt num)`

Backup, remove num items at end of fifo. Careful, make sure read is not already happening.

6.32.2.18 `template<class Type> Type* BFifo< Type >::writeData ()`

Returns a pointer to the data.

6.32.2.19 `template<class Type> Type* BFifo< Type >::writeData (BUInt & num)`

Returns a pointer to the data and how many can be written in a chunk.

6.32.2.20 `template<class Type> void BFifo< Type >::writeDone (BUInt num)`

Indicates when write is complete.

6.32.3 Member Data Documentation

6.32.3.1 `template<class Type> Type* BFifo< Type >::odata [protected]`

FIFO memory buffer.

6.32.3.2 `template<class Type> BMutex BFifo< Type >::olock [protected]`

6.32.3.3 `template<class Type> BUInt BFifo< Type >::oreadPos [protected]`

The read pointer.

6.32.3.4 `template<class Type> BUInt BFifo< Type >::osize [protected]`

The size of the FIFO.

6.32.3.5 `template<class Type> BUInt BFifo< Type >::owritePos` `[protected]`

The write pointer.

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

- [BFifo.h](#)

6.33 BFifoCirc< Type > Class Template Reference

This class implements a thread safe FIFO buffer.

```
#include <BFifoCirc.h>
```

Public Types

- enum { `defaultSize` = 1024 }

Public Member Functions

- `BFifoCirc` (`uint32_t size=defaultSize`)
- `~BFifoCirc` ()
- `uint32_t size` ()
Return the buffers actual size.
- void `clear` ()
Clear all of the data in the buffer.
- `uint32_t writeAvailable` ()
Returns the space available to write.
- `BError writeWaitAvailable` (`uint32_t numFifoSamples`)
Wait for the given number of samples.
- `BError write` (`const Type *data, uint32_t numFifoSamples`)
Writes the data to the buffer. Blocks until complete.
- `Type * writeData` ()
Return a pointer to the current start of the buffer.
- void `writeDone` (`uint32_t numFifoSamples`)
Update the write pointer.
- `uint32_t readAvailable` ()
Returns the number of bytes of data available.
- `BError readWaitAvailable` (`uint32_t numFifoSamples`)
Wait for given number of samples.
- `BError read` (`Type *data, uint32_t numFifoSamples`)
- `Type * readData` ()
Pointer to raw data.
- `BError readDone` (`uint32_t numFifoSamples`)
Updates read pointer.
- `Type & operator[]` (`int pos`)
Direct access to read samples in buffer.

Protected Member Functions

- `BError mapCircularBuffer` (`uint32_t size`)
- void `unmapCircularBuffer` ()

Protected Attributes

- [BMutex olock](#)
- [uint32_t ovmSize](#)
- [uint32_t osize](#)
- [Type * odata](#)
- [BFifoCircPos owritePos](#)
Current write position.
- [BCondValue owriteNumFifoSamples](#)
The number of samples in the FIFO.
- [BFifoCircPos oreadPos](#)
Current read position.

6.33.1 Detailed Description

```
template<class Type>class BFifoCirc< Type >
```

This class implements a thread safe FIFO buffer.

6.33.2 Member Enumeration Documentation

6.33.2.1 `template<class Type > anonymous enum`

Enumerator

defaultSize

6.33.3 Constructor & Destructor Documentation

6.33.3.1 `template<class Type > BFifoCirc< Type >::BFifoCirc (uint32_t size = defaultSize)`

6.33.3.2 `template<class Type > BFifoCirc< Type >::~~BFifoCirc ()`

6.33.4 Member Function Documentation

6.33.4.1 `template<class Type > void BFifoCirc< Type >::clear ()`

Clear all of the data in the buffer.

6.33.4.2 `template<class Type > BError BFifoCirc< Type >::mapCircularBuffer (uint32_t size)` `[protected]`

6.33.4.3 `template<class Type > Type& BFifoCirc< Type >::operator[] (int pos)`

Direct access to read samples in buffer.

6.33.4.4 `template<class Type > BError BFifoCirc< Type >::read (Type * data, uint32_t numFifoSamples)`

6.33.4.5 `template<class Type > uint32_t BFifoCirc< Type >::readAvailable ()`

Returns the number of bytes of data available.

6.33.4.6 `template<class Type > Type* BFifoCirc< Type >::readData ()`

Pointer to raw data.

6.33.4.7 `template<class Type > BError BFifoCirc< Type >::readDone (uint32_t numFifoSamples)`

Updates read pointer.

6.33.4.8 `template<class Type > BError BFifoCirc< Type >::readWaitAvailable (uint32_t numFifoSamples)`

Wait for given number of samples.

6.33.4.9 `template<class Type > uint32_t BFifoCirc< Type >::size ()`

Return the buffers actual size.

6.33.4.10 `template<class Type > void BFifoCirc< Type >::unmapCircularBuffer ()` [protected]

6.33.4.11 `template<class Type > BError BFifoCirc< Type >::write (const Type * data, uint32_t numFifoSamples)`

Writes the data to the buffer. Blocks until complete.

6.33.4.12 `template<class Type > uint32_t BFifoCirc< Type >::writeAvailable ()`

Returns the space available to write.

6.33.4.13 `template<class Type > Type* BFifoCirc< Type >::writeData ()`

Return a pointer to the current start of the buffer.

6.33.4.14 `template<class Type > void BFifoCirc< Type >::writeDone (uint32_t numFifoSamples)`

Update the write pointer.

6.33.4.15 `template<class Type > BError BFifoCirc< Type >::writeWaitAvailable (uint32_t numFifoSamples)`

Wait for the given number of samples.

6.33.5 Member Data Documentation

6.33.5.1 `template<class Type > Type* BFifoCirc< Type >::odata` [protected]

6.33.5.2 `template<class Type > BMutex BFifoCirc< Type >::olock` [protected]

6.33.5.3 `template<class Type > BFifoCircPos BFifoCirc< Type >::oreadPos` [protected]

Current read position.

6.33.5.4 `template<class Type > uint32_t BFifoCirc< Type >::osize` [protected]

6.33.5.5 `template<class Type > uint32_t BFifoCirc< Type >::ovmSize` [protected]

6.33.5.6 `template<class Type > BCondValue BFifoCirc< Type >::owriteNumFifoSamples` [protected]

The number of samples in the FIFO.

6.33.5.7 `template<class Type > BFifoCircPos BFifoCirc< Type >::owritePos` [protected]

Current write position.

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

- [BFifoCirc.h](#)

6.34 BFifoCircPos Class Reference

This class implements a pointer into the Fifo's circular buffer.

```
#include <BFifoCirc.h>
```

Public Member Functions

- [BFifoCircPos](#) (uint32_t size)
- void [setSize](#) (uint32_t size)
- void [set](#) (uint32_t [pos](#))
Sets the position.
- uint32_t [pos](#) ()
The current position.
- void [increment](#) (uint32_t numFifoSamples)
Increment the pointer by the given value.
- uint32_t [difference](#) (const [BFifoCircPos](#) &[pos](#))
Return the difference between the two pointers.
- [operator int](#) ()
- void [operator+=](#) (uint32_t numFifoSamples)
- int [operator==](#) (const [BFifoCircPos](#) &[pos](#))
- int [operator!=](#) (const [BFifoCircPos](#) &[pos](#))

Private Attributes

- uint32_t [osize](#)
- uint32_t [opos](#)

6.34.1 Detailed Description

This class implements a pointer into the Fifo's circular buffer.

6.34.2 Constructor & Destructor Documentation

6.34.2.1 `BFifoCircPos::BFifoCircPos (uint32_t size)`

6.34.3 Member Function Documentation

6.34.3.1 `uint32_t BFifoCircPos::difference (const BFifoCircPos & pos)`

Return the difference between the two pointers.

6.34.3.2 `void BFifoCircPos::increment (uint32_t numFifoSamples)`

Increment the pointer by the given value.

6.34.3.3 `BFifoCircPos::operator int ()`

6.34.3.4 `int BFifoCircPos::operator!= (const BFifoCircPos & pos)`

6.34.3.5 `void BFifoCircPos::operator+= (uint32_t numFifoSamples)`

6.34.3.6 `int BFifoCircPos::operator== (const BFifoCircPos & pos)`

6.34.3.7 `uint32_t BFifoCircPos::pos ()`

The current position.

6.34.3.8 `void BFifoCircPos::set (uint32_t pos)`

Sets the position.

6.34.3.9 `void BFifoCircPos::setSize (uint32_t size)`

6.34.4 Member Data Documentation

6.34.4.1 `uint32_t BFifoCircPos::opos [private]`

6.34.4.2 `uint32_t BFifoCircPos::osize [private]`

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

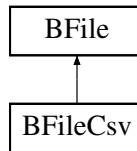
- [BFifoCirc.h](#)
- [BFifoCirc.cpp](#)

6.35 BFile Class Reference

File operations class.

```
#include <BFile.h>
```

Inheritance diagram for BFile:



Public Member Functions

- [BFile](#) ()
- [BFile](#) (const [BFile](#) &file)
Create opened specified file.
- [~BFile](#) ()
- [BError open](#) ([BString](#) name, [BString](#) mode)
Open file.
- [BError open](#) (FILE *file)
Assign object to opened file handle.
- [BError open](#) (int fd, [BString](#) mode)
Assign object to opened file descriptor.
- [BError close](#) ()
Close file.
- int [isOpen](#) ()
Returns 1 if the file is open.
- int [isEnd](#) ()
Returns 1 if at the end of the file, 0 otherwise.
- FILE * [getFd](#) ()
File descriptor.
- [BUInt64 length](#) ()
File size in bytes.
- int [setVBuf](#) (char *buf, int mode, size_t size)
Set stream buffering options.
- int [read](#) (void *buf, int nbytes)
Read from file.
- int [readString](#) ([BString](#) &str)
Read string. (ref fgets)
- char * [fgets](#) (char *buf, size_t size)
Read string. (ref fgets)
- int [write](#) (const void *buf, int nbytes)
Write to file.
- int [writeString](#) (const [BString](#) &str)
Write string to file.
- int [seek](#) ([BUInt64](#) pos)
Set seek position.
- [BUInt64 position](#) ()
The files position.
- int [printf](#) (const char *fmt,...)
Formatted print into the file.
- [BError truncate](#) ()
Truncate the file.
- [BError flush](#) ()
Flush the file.
- [BString fileName](#) ()
Return file name.
- [BFile](#) & [operator=](#) (const [BFile](#) &file)

Private Attributes

- FILE * [ofile](#)
- BString [ofilename](#)
- BString [omode](#)

6.35.1 Detailed Description

File operations class.

6.35.2 Constructor & Destructor Documentation

6.35.2.1 BFile::BFile ()

6.35.2.2 BFile::BFile (const BFile & *file*)

Create opened specified file.

6.35.2.3 BFile::~~BFile ()

6.35.3 Member Function Documentation

6.35.3.1 BError BFile::close ()

Close file.

6.35.3.2 char * BFile::fgets (char * *buf*, size_t *size*)

6.35.3.3 BString BFile::fileName ()

Return file name.

6.35.3.4 BError BFile::flush ()

Flush the file.

6.35.3.5 FILE * BFile::getFd ()

File descriptor.

6.35.3.6 int BFile::isEnd ()

Returns 1 if at the end of the file, 0 otherwise.

6.35.3.7 int BFile::isOpen ()

Returns 1 if the file is open.

6.35.3.8 BUInt64 BFile::length ()

File size in bytes.

6.35.3.9 BError BFile::open (BString name, BString mode)

Open file.

6.35.3.10 BError BFile::open (FILE * file)

Assign object to opened file handle.

6.35.3.11 BError BFile::open (int fd, BString mode)

Assign object to opened file descriptor.

6.35.3.12 BFile & BFile::operator= (const BFile & file)**6.35.3.13 BUInt64 BFile::position ()**

The files position.

6.35.3.14 int BFile::printf (const char * fmt, ...)

Formatted print into the file.

6.35.3.15 int BFile::read (void * buf, int nbytes)

Read from file.

6.35.3.16 int BFile::readString (BString & str)

Read string. (ref fgets)

6.35.3.17 int BFile::seek (BUInt64 pos)

Set seek position.

6.35.3.18 int BFile::setVBuf (char * buf, int mode, size_t size)

Set stream buffering options.

6.35.3.19 BError BFile::truncate ()

Truncate the file.

6.35.3.20 int BFile::write (const void * buf, int nbytes)

Write to file.

6.35.3.21 int BFile::writeString (const BString & str)

Write string to file.

6.35.4 Member Data Documentation

6.35.4.1 `FILE* BFile::ofile` [private]

6.35.4.2 `BString BFile::ofilename` [private]

6.35.4.3 `BString BFile::omode` [private]

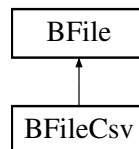
The documentation for this class was generated from the following files:

- [BFile.h](#)
- [BFile.cpp](#)

6.36 BFileCsv Class Reference

```
#include <BFileCsv.h>
```

Inheritance diagram for BFileCsv:



Public Member Functions

- [BFileCsv](#) (char separator= ';')
- [BError readCsv](#) (BStringList &csvList)
- [BError writeCsv](#) (BStringList &csvList)

Private Attributes

- char [oseparator](#)

6.36.1 Constructor & Destructor Documentation

6.36.1.1 `BFileCsv::BFileCsv (char separator = ' ; ')`

6.36.2 Member Function Documentation

6.36.2.1 `BError BFileCsv::readCsv (BStringList & csvList)`

6.36.2.2 `BError BFileCsv::writeCsv (BStringList & csvList)`

6.36.3 Member Data Documentation

6.36.3.1 `char BFileCsv::oseparator` [private]

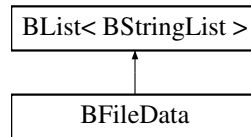
The documentation for this class was generated from the following files:

- [BFileCsv.h](#)
- [BFileCsv.cpp](#)

6.37 BFileData Class Reference

```
#include <BFileData.h>
```

Inheritance diagram for BFileData:



Public Member Functions

- [BError open](#) ([BString](#) filename)
- [BError getNextId](#) (int &id)
- [BError find](#) (int id, [BStringList](#) &csvList)
- [BError write](#) (int id, [BStringList](#) &csvList)
- [BError del](#) (int id)

Private Member Functions

- [BError read](#) ()
- [BError write](#) ()

Private Attributes

- [BString ofilename](#)

Additional Inherited Members

6.37.1 Member Function Documentation

6.37.1.1 **BError** BFileData::del (int *id*)

6.37.1.2 **BError** BFileData::find (int *id*, [BStringList](#) & *csvList*)

6.37.1.3 **BError** BFileData::getNextId (int & *id*)

6.37.1.4 **BError** BFileData::open ([BString](#) *filename*)

6.37.1.5 **BError** BFileData::read () [private]

6.37.1.6 **BError** BFileData::write (int *id*, [BStringList](#) & *csvList*)

6.37.1.7 **BError** BFileData::write () [private]

6.37.2 Member Data Documentation

6.37.2.1 **BString** BFileData::ofilename [private]

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

- [BFileData.h](#)
- [BFileData.cpp](#)

6.38 Blter Class Reference

Iterator for [BList](#).

```
#include <BList.h>
```

Public Member Functions

- [Blter](#) ([BNode](#) *i=0)
- [operator BNode *](#) ()
- [int operator==](#) (const [Blter](#) &i)
- [int valid](#) ()

Private Attributes

- [BNode](#) * oi

6.38.1 Detailed Description

Iterator for [BList](#).

6.38.2 Constructor & Destructor Documentation

6.38.2.1 [Blter::Blter](#) ([BNode](#) * i = 0) `[inline]`

6.38.3 Member Function Documentation

6.38.3.1 [Blter::operator BNode *](#) () `[inline]`

6.38.3.2 [int Blter::operator==](#) (const [Blter](#) & i) `[inline]`

6.38.3.3 [int Blter::valid](#) () `[inline]`

6.38.4 Member Data Documentation

6.38.4.1 [BNode*](#) [Blter::oi](#) `[private]`

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

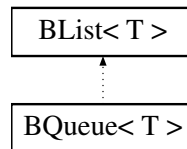
- [BList.h](#)

6.39 BList< T > Class Template Reference

Template based list class.

```
#include <BList.h>
```

Inheritance diagram for BList< T >:



Classes

- class [Node](#)

Public Types

- typedef int(* [SortFunc](#))(T &a, T &b)
Prototype for sorting function.

Public Member Functions

- [BList](#) ()
- [BList](#) (const [BList](#)< T > &l)
- virtual [~BList](#) ()
- void [start](#) ([Blter](#) &i) const
Iterator to start of list.
- [Blter begin](#) () const
Iterator for start of list.
- [Blter end](#) () const
Iterator for end of list.
- [Blter end](#) ([Blter](#) &i) const
Iterator for end of list.
- void [next](#) ([Blter](#) &i) const
Iterator for next item in list.
- void [prev](#) ([Blter](#) &i)
Iterator for previous item in list.
- [Blter goTo](#) (int pos) const
Iterator for pos item in list.
- int [position](#) ([Blter](#) i)
Postition in list item with iterator i.
- unsigned int [number](#) () const
Number of items in list.
- unsigned int [size](#) () const
Number of items in list.
- int [isEnd](#) ([Blter](#) &i) const
True if iterator refers to last item.
- T & [front](#) ()
Get first item in list.
- T & [rear](#) ()
Get last item in list.
- T & [get](#) ([Blter](#) i)
Get item specified by iterator in list.
- const T & [get](#) ([Blter](#) i) const
Get item specified by iterator in list.

- void [append](#) (const T &item)
Append item to list.
- virtual void [insert](#) ([BIter](#) &i, const T &item)
Insert item before item.
- void [insertAfter](#) ([BIter](#) &i, const T &item)
Insert item after item.
- virtual void [clear](#) ()
Clear the list.
- virtual void [del](#) ([BIter](#) &i)
Delete specified item.
- void [deleteLast](#) ()
Delete last item.
- void [deleteFirst](#) ()
Delete first item.
- void [push](#) (const T &i)
Push item onto list.
- T [pop](#) ()
Pop item from list deleting item.
- void [queueAdd](#) (const T &i)
Add item to end of list.
- T [queueGet](#) ()
Get item from front of list deleting item.
- void [append](#) (const [BList](#)< T > &l)
Append list to list.
- int [has](#) (const T &i) const
Checks if the item is in the list.
- void [swap](#) ([BIter](#) i1, [BIter](#) i2)
Swap two items in list.
- void [sort](#) ()
Sort list based on get(i) values.
- void [sort](#) ([SortFunc](#) func)
Sort list based on Sort func.
- [BList](#)< T > & [operator=](#) (const [BList](#)< T > &l)
- T & [operator\[\]](#) (int i)
- const T & [operator\[\]](#) (int i) const
- T & [operator\[\]](#) ([BIter](#) i)
- const T & [operator\[\]](#) (const [BIter](#) &i) const
- [BList](#)< T > [operator+](#) (const [BList](#)< T > &l) const

Protected Member Functions

- virtual [Node](#) * [nodeGet](#) ([BIter](#) i)
- virtual const [Node](#) * [nodeGet](#) ([BIter](#) i) const
- virtual [Node](#) * [nodeCreate](#) (const T &item)

Protected Attributes

- [Node](#) * [onodes](#)
- unsigned int [olength](#)

Private Member Functions

- virtual [Node](#) * [nodeCreate](#) ()

6.39.1 Detailed Description

`template<class T>class BList< T >`

Template based list class.

6.39.2 Member Typedef Documentation

6.39.2.1 `template<class T> typedef int(* BList< T >::SortFunc)(T &a, T &b)`

Prototype for sorting function.

6.39.3 Constructor & Destructor Documentation

6.39.3.1 `template<class T > BList< T >::BList ()`

6.39.3.2 `template<class T> BList< T >::BList (const BList< T > & l)`

6.39.3.3 `template<class T > BList< T >::~~BList ()` [virtual]

6.39.4 Member Function Documentation

6.39.4.1 `template<class T> void BList< T >::append (const T & item)`

Append item to list.

6.39.4.2 `template<class T> void BList< T >::append (const BList< T > & l)`

Append list to list.

6.39.4.3 `template<class T > BIter BList< T >::begin () const`

Iterator for start of list.

6.39.4.4 `template<class T > void BList< T >::clear ()` [virtual]

Clear the list.

Reimplemented in [BEntryFile](#), [BDir](#), [BEntryList](#), [BDict< Type >](#), [BQueue< T >](#), and [BQueue< BoapMcPacket >](#).

6.39.4.5 `template<class T > void BList< T >::del (BIter & i)` [virtual]

Delete specified item.

Reimplemented in [BEntryList](#), and [BDict< Type >](#).

6.39.4.6 `template<class T> void BList< T >::deleteFirst ()`

Delete first item.

6.39.4.7 `template<class T> void BList< T >::deleteLast ()`

Delete last item.

6.39.4.8 `template<class T> Blter BList< T >::end () const`

Iterator for end of list.

6.39.4.9 `template<class T> Blter BList< T >::end (Blter & i) const`

Iterator for end of list.

6.39.4.10 `template<class T> T & BList< T >::front ()`

Get first item in list.

6.39.4.11 `template<class T> T & BList< T >::get (Blter i)`

Get item specified by iterator in list.

6.39.4.12 `template<class T> const T & BList< T >::get (Blter i) const`

Get item specified by iterator in list.

6.39.4.13 `template<class T> Blter BList< T >::goTo (int pos) const`

Iterator for pos item in list.

6.39.4.14 `template<class T> int BList< T >::has (const T & i) const`

Checks if the item is in the list.

6.39.4.15 `template<class T> void BList< T >::insert (Blter & i, const T & item) [virtual]`

Insert item before item.

Reimplemented in [BEntryList](#), and [BDict< Type >](#).

6.39.4.16 `template<class T> void BList< T >::insertAfter (Blter & i, const T & item)`

Insert item after item.

6.39.4.17 `template<class T> int BList< T >::isEnd (Blter & i) const`

True if iterator refers to last item.

6.39.4.18 `template<class T> void BList< T>::next (BIter & i) const`

Iterator for next item in list.

6.39.4.19 `template<class T> BList< T>::Node * BList< T>::nodeCreate (const T & item) [protected], [virtual]`

6.39.4.20 `template<class T> BList< T>::Node * BList< T>::nodeCreate () [private], [virtual]`

6.39.4.21 `template<class T> BList< T>::Node * BList< T>::nodeGet (BIter i) [protected], [virtual]`

6.39.4.22 `template<class T> const BList< T>::Node * BList< T>::nodeGet (BIter i) const [protected], [virtual]`

6.39.4.23 `template<class T> unsigned int BList< T>::number () const`

Number of items in list.

6.39.4.24 `template<class T> BList< T> BList< T>::operator+ (const BList< T> & l) const`

6.39.4.25 `template<class T> BList< T> & BList< T>::operator= (const BList< T> & l)`

6.39.4.26 `template<class T> T & BList< T>::operator[] (int i)`

6.39.4.27 `template<class T> const T & BList< T>::operator[] (int i) const`

6.39.4.28 `template<class T> T & BList< T>::operator[] (BIter i)`

6.39.4.29 `template<class T> const T & BList< T>::operator[] (const BIter & i) const`

6.39.4.30 `template<class T> T BList< T>::pop ()`

Pop item from list deleteing item.

6.39.4.31 `template<class T> int BList< T>::position (BIter i)`

Postition in list item with iterator i.

6.39.4.32 `template<class T> void BList< T>::prev (BIter & i)`

Iterator for previous item in list.

6.39.4.33 `template<class T> void BList< T>::push (const T & i)`

Push item onto list.

6.39.4.34 `template<class T> void BList< T>::queueAdd (const T & i)`

Add item to end of list.

6.39.4.35 `template<class T> T BList< T >::queueGet ()`

Get item from front of list deleteing item.

6.39.4.36 `template<class T> T & BList< T >::rear ()`

Get last item in list.

6.39.4.37 `template<class T> unsigned int BList< T >::size () const`

Number of items in list.

6.39.4.38 `template<class T> void BList< T >::sort ()`

Sort list based on get(i) values.

6.39.4.39 `template<class T> void BList< T >::sort (SortFunc func)`

Sort list based on Sort func.

6.39.4.40 `template<class T> void BList< T >::start (BIter & i) const`

Iterator to start of list.

6.39.4.41 `template<class T> void BList< T >::swap (BIter i1, BIter i2)`

Swap two items in list.

6.39.5 Member Data Documentation

6.39.5.1 `template<class T> unsigned int BList< T >::olength [protected]`

6.39.5.2 `template<class T> Node* BList< T >::onodes [protected]`

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

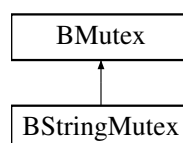
- [BList.h](#)
- [BList_func.h](#)

6.40 BMutex Class Reference

Mutex class.

```
#include <BMutex.h>
```

Inheritance diagram for BMutex:



Public Types

- enum `Type` { `Normal`, `Recursive` }

Public Member Functions

- `BMutex` (`Type` type=`Normal`)
- `BMutex` (`const BMutex` &mutex)
- `~BMutex` ()
- `int lock` ()
Set lock, wait as necessary.
- `int timedLock` (`int` timeoutUs)
Set lock, wait as necessary but timeout after given time.
- `int unlock` ()
Unlock the lock.
- `int tryLock` ()
Test the lock.
- `BMutex` & `operator=` (`const BMutex` &mutex)

Private Attributes

- `pthread_mutex_t omutex`

6.40.1 Detailed Description

Mutex class.

6.40.2 Member Enumeration Documentation

6.40.2.1 enum `BMutex::Type`

Enumerator

Normal

Recursive

6.40.3 Constructor & Destructor Documentation

6.40.3.1 `BMutex::BMutex` (`Type` type = `Normal`)

6.40.3.2 `BMutex::BMutex` (`const BMutex` & mutex)

6.40.3.3 `BMutex::~~BMutex` ()

6.40.4 Member Function Documentation

6.40.4.1 `int BMutex::lock` ()

Set lock, wait as necessary.

6.40.4.2 **BMutex** & **BMutex::operator=** (const **BMutex** & *mutex*)

6.40.4.3 int **BMutex::timedLock** (int *timeoutUs*)

Set lock, wait as necessary but timeout after given time.

6.40.4.4 int **BMutex::tryLock** ()

Test the lock.

6.40.4.5 int **BMutex::unlock** ()

Unlock the lock.

6.40.5 Member Data Documentation

6.40.5.1 pthread_mutex_t **BMutex::omutex** [private]

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

- [BMutex.h](#)
- [BMutex.cpp](#)

6.41 BMutexLock Class Reference

```
#include <BMutex.h>
```

Public Member Functions

- [BMutexLock](#) ([BMutex](#) &[lock](#), int doLock=0)
- [~BMutexLock](#) ()
- int [lock](#) ()
- int [unlock](#) ()

Private Attributes

- [BMutex](#) & [olock](#)

6.41.1 Constructor & Destructor Documentation

6.41.1.1 **BMutexLock::BMutexLock** (**BMutex** & *lock*, int *doLock* = 0) [inline]

6.41.1.2 **BMutexLock::~BMutexLock** () [inline]

6.41.2 Member Function Documentation

6.41.2.1 int **BMutexLock::lock** () [inline]

6.41.2.2 int **BMutexLock::unlock** () [inline]

6.41.3 Member Data Documentation

6.41.3.1 BMutex& BMutexLock::olock [private]

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

- [BMutex.h](#)

6.42 BMySQL Class Reference

```
#include <BMySQL.h>
```

Public Member Functions

- [BMySQL](#) ()
- [~BMySQL](#) ()
- [BError open](#) ([BString](#) hostName, [BString](#) dataBase, [BString](#) userName, [BString](#) password)
- void [close](#) ()
- [BError get](#) ([BString](#) table, [BString](#) where, [BDictString](#) &fields)
- [BError insert](#) ([BString](#) table, [BDictString](#) fields, [BUInt32](#) *id=0)
- [BError update](#) ([BString](#) table, [BUInt32](#) id, [BDictString](#) fields)
- [BError del](#) ([BString](#) table, [BUInt32](#) id)
Delete record from table.
- [BError flush](#) ()
Flush all data to disk.
- [BString escapeString](#) ([BString](#) str)
Escapes special characters in the string.
- [BError query](#) ([BString](#) cmd, [BList](#)< [BDictString](#) > &result)
- [MYSQL & db](#) ()
- void [setDebug](#) (int debug)

Private Attributes

- [MYSQL odb](#)
- int [oopened](#)
- int [odebug](#)
- [BMutex olock](#)

6.42.1 Constructor & Destructor Documentation

6.42.1.1 BMySQL::BMySQL ()

6.42.1.2 BMySQL::~~BMySQL ()

6.42.2 Member Function Documentation

6.42.2.1 void BMySQL::close ()

6.42.2.2 MYSQL & BMySQL::db ()

6.42.2.3 BError BMySQL::del ([BString](#) table, [BUInt32](#) id)

Delete record from table.

6.42.2.4 BString BMySQL::escapeString (BString *str*)

Escapes special characters in the string.

6.42.2.5 BError BMySQL::flush ()

Flush all data to disk.

6.42.2.6 BError BMySQL::get (BString *table*, BString *where*, BDictString & *fields*)6.42.2.7 BError BMySQL::insert (BString *table*, BDictString *fields*, BUInt32 * *id* = 0)6.42.2.8 BError BMySQL::open (BString *hostName*, BString *dataBase*, BString *userName*, BString *password*)6.42.2.9 BError BMySQL::query (BString *cmd*, BList< BDictString > & *result*)6.42.2.10 void BMySQL::setDebug (int *debug*)6.42.2.11 BError BMySQL::update (BString *table*, BUInt32 *id*, BDictString *fields*)

6.42.3 Member Data Documentation

6.42.3.1 MYSQL BMySQL::odb [private]

6.42.3.2 int BMySQL::odebug [private]

6.42.3.3 BMutex BMySQL::olock [private]

6.42.3.4 int BMySQL::oopened [private]

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

- [BMySQL.h](#)
- [BMySQL.cpp](#)

6.43 BNameValue< T > Class Template Reference

```
#include <BNameValue.h>
```

Public Member Functions

- [BNameValue](#) ()
- [BNameValue](#) (BString *name*, const T &*value*)
- [BString](#) [getName](#) ()
- T & [getValue](#) ()

Private Attributes

- [BString](#) *oname*
- T *ovalue*

6.43.1 Constructor & Destructor Documentation

6.43.1.1 `template<class T> BNameValue<T>::BNameValue () [inline]`

6.43.1.2 `template<class T> BNameValue<T>::BNameValue (BString name, const T & value) [inline]`

6.43.2 Member Function Documentation

6.43.2.1 `template<class T> BString BNameValue<T>::getName () [inline]`

6.43.2.2 `template<class T> T& BNameValue<T>::getValue () [inline]`

6.43.3 Member Data Documentation

6.43.3.1 `template<class T> BString BNameValue<T>::oname [private]`

6.43.3.2 `template<class T> T BNameValue<T>::ovalue [private]`

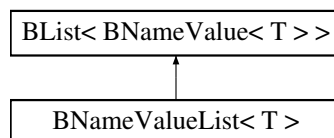
The documentation for this class was generated from the following file:

- [BNameValue.h](#)

6.44 BNameValueList< T > Class Template Reference

```
#include <BNameValue.h>
```

Inheritance diagram for BNameValueList< T >:



Public Member Functions

- `T * find (BString name)`
- `BIter findPos (BString name)`

Additional Inherited Members

6.44.1 Member Function Documentation

6.44.1.1 `template<class T> T* BNameValueList<T>::find (BString name) [inline]`

6.44.1.2 `template<class T> BIter BNameValueList<T>::findPos (BString name) [inline]`

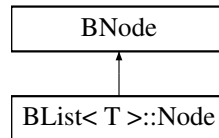
The documentation for this class was generated from the following file:

- [BNameValue.h](#)

6.45 BNode Class Reference

```
#include <BList.h>
```

Inheritance diagram for BNode:



Public Member Functions

- [BNode\(\)](#)

Public Attributes

- [BNode * next](#)
- [BNode * prev](#)

6.45.1 Constructor & Destructor Documentation

6.45.1.1 [BNode::BNode\(\)](#) `[inline]`

6.45.2 Member Data Documentation

6.45.2.1 [BNode*](#) [BNode::next](#)

6.45.2.2 [BNode*](#) [BNode::prev](#)

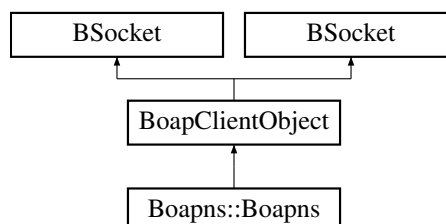
The documentation for this class was generated from the following file:

- [BList.h](#)

6.46 BoapClientObject Class Reference

```
#include <BoapSimple.h>
```

Inheritance diagram for BoapClientObject:



Public Member Functions

- [BoapClientObject](#) (BString name=(""))
- virtual [~BoapClientObject](#) ()
- [BError connectService](#) (BString name)
Connects to the named service.
- [BError disconnectService](#) ()
Disconnects from the named service.
- [BString getServiceName](#) ()
Get the name of the service.
- [BError ping](#) (BUInt32 &apiVersion)
Pings the connection and finds the remotes version number.
- [BError setConnectionPriority](#) ([BoapPriority](#) priority)
Sets the connection priority.
- void [setMaxLength](#) (BUInt32 maxLength)
Sets the maximum packet length.
- void [setTimeout](#) (int timeout)
Sets the timeout in micro seconds. -1 is wait indefinitely.
- [BoapClientObject](#) (BString name)
- [BError connectService](#) (BString name)

Protected Member Functions

- [BError pingLocked](#) (BUInt32 &apiVersion)
- [BError checkApiVersion](#) ()
- [BError performCall](#) ([BoapPacket](#) &tx, [BoapPacket](#) &rx)
Performs a RPC call to the named service.
- [BError performSend](#) ([BoapPacket](#) &tx)
Performs a send to the named service.
- [BError performRecv](#) ([BoapPacket](#) &rx)
Performs a receive.
- virtual [BError handleReconnect](#) ([BError](#) err)
Handle a reconnect performing autorisaztion if required.
- [BError performSend](#) ([BoapPacket](#) &tx)
- [BError performRecv](#) ([BoapPacket](#) &rx)
- [BError performCall](#) ([BoapPacket](#) &tx, [BoapPacket](#) &rx)

Protected Attributes

- [BString](#) oname
- [BUInt32](#) oapiVersion
- [BoapPriority](#) opriority
- [BoapService](#) oservice
- int oconnected
- [BUInt32](#) omaxLength
- [BoapPacket](#) otx
- [BoapPacket](#) orx
- [BMutex](#) olock
- int otimeout
- int oreconnect
Handle an automatic reconnect on timeout.

Additional Inherited Members

6.46.1 Constructor & Destructor Documentation

6.46.1.1 **BoapClientObject::BoapClientObject** (**BString** *name* = " ")

6.46.1.2 **BoapClientObject::~~BoapClientObject** () [virtual]

6.46.1.3 **BoapClientObject::BoapClientObject** (**BString** *name*)

6.46.2 Member Function Documentation

6.46.2.1 **BError** **BoapClientObject::checkApiVersion** () [protected]

6.46.2.2 **BError** **BoapClientObject::connectService** (**BString** *name*)

Connects to the named service.

6.46.2.3 **BError** **BoapClientObject::connectService** (**BString** *name*)

6.46.2.4 **BError** **BoapClientObject::disconnectService** ()

Disconnects from the named service.

6.46.2.5 **BString** **BoapClientObject::getServiceName** ()

Get the name of the service.

6.46.2.6 **BError** **BoapClientObject::handleReconnect** (**BError** *err*) [protected],[virtual]

Handle a reconnect performing autorisazion if required.

6.46.2.7 **BError** **BoapClientObject::performCall** (**BoapPacket** & *tx*, **BoapPacket** & *rx*) [protected]

Performs a RPC call to the named service.

6.46.2.8 **BError** **BoapClientObject::performCall** (**BoapPacket** & *tx*, **BoapPacket** & *rx*) [protected]

6.46.2.9 **BError** **BoapClientObject::performRecv** (**BoapPacket** & *rx*) [protected]

Performs a receive.

6.46.2.10 **BError** **BoapClientObject::performRecv** (**BoapPacket** & *rx*) [protected]

6.46.2.11 **BError** **BoapClientObject::performSend** (**BoapPacket** & *tx*) [protected]

Performs a send to the named service.

6.46.2.12 **BError** **BoapClientObject::performSend** (**BoapPacket** & *tx*) [protected]

6.46.2.13 **BError** BoapClientObject::ping (**BUInt32** & *apiVersion*)

Pings the connection and finds the remotes version number.

6.46.2.14 **BError** BoapClientObject::pingLocked (**BUInt32** & *apiVersion*) [protected]

6.46.2.15 **BError** BoapClientObject::setConnectionPriority (**BoapPriority** *priority*)

Sets the connection priority.

6.46.2.16 **void** BoapClientObject::setMaxLength (**BUInt32** *maxLength*)

Sets the maximum packet length.

6.46.2.17 **void** BoapClientObject::setTimeout (**int** *timeout*)

Sets the timeout in micro seconds. -1 is wait indefinitely.

6.46.3 Member Data Documentation

6.46.3.1 **BUInt32** BoapClientObject::oapiVersion [protected]

6.46.3.2 **int** BoapClientObject::oconnected [protected]

6.46.3.3 **BMutex** BoapClientObject::olock [protected]

6.46.3.4 **BUInt32** BoapClientObject::omaxLength [protected]

6.46.3.5 **BString** BoapClientObject::oname [protected]

6.46.3.6 **BoapPriority** BoapClientObject::opriority [protected]

6.46.3.7 **int** BoapClientObject::oreconnect [protected]

Handle an automatic reconnect on timeout.

6.46.3.8 **BoapPacket** BoapClientObject::orx [protected]

6.46.3.9 **BoapService** BoapClientObject::oservice [protected]

6.46.3.10 **int** BoapClientObject::otimeout [protected]

6.46.3.11 **BoapPacket** BoapClientObject::otx [protected]

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

- [Boap.h](#)
- [BoapSimple.h](#)
- [Boap.cpp](#)
- [BoapSimple.cc](#)

6.47 Boapns::BoapEntry Class Reference

```
#include <BoapnsD.h>
```

Public Member Functions

- [BoapEntry](#) ([BString](#) pname=[BString](#)(), [BString](#) phostName=[BString](#)(), [BList](#)< [BString](#) > paddressList=[BList](#)< [BString](#) >(), [BUInt32](#) pport=[BUInt32](#)(), [BUInt32](#) pservice=[BUInt32](#)())

Public Attributes

- [BString](#) name
- [BString](#) hostName
- [BList](#)< [BString](#) > addressList
- [BUInt32](#) port
- [BUInt32](#) service

6.47.1 Constructor & Destructor Documentation

6.47.1.1 [Boapns::BoapEntry::BoapEntry](#) ([BString](#) pname = [BString](#) () , [BString](#) phostName = [BString](#) () , [BList](#)< [BString](#) > paddressList = [BList](#)<[BString](#) > () , [BUInt32](#) pport = [BUInt32](#) () , [BUInt32](#) pservice = [BUInt32](#) ())

6.47.2 Member Data Documentation

6.47.2.1 [BList](#)<[BString](#) > [Boapns::BoapEntry::addressList](#)

6.47.2.2 [BString](#) [Boapns::BoapEntry::hostName](#)

6.47.2.3 [BString](#) [Boapns::BoapEntry::name](#)

6.47.2.4 [BUInt32](#) [Boapns::BoapEntry::port](#)

6.47.2.5 [BUInt32](#) [Boapns::BoapEntry::service](#)

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

- [BoapnsD.h](#)
- [BoapnsD.cpp](#)

6.48 BoapFuncEntry Class Reference

```
#include <BoapSimple.h>
```

Public Member Functions

- [BoapFuncEntry](#) (int cmd, [BoapFunc](#) func)
- [BoapFuncEntry](#) (int cmd, [BoapFunc](#) func)

Public Attributes

- [BUInt32 ocmd](#)
- [BoapFunc ofunc](#)
- [UInt32 ocmd](#)

6.48.1 Constructor & Destructor Documentation

6.48.1.1 [BoapFuncEntry::BoapFuncEntry \(int cmd, BoapFunc func \)](#)

6.48.1.2 [BoapFuncEntry::BoapFuncEntry \(int cmd, BoapFunc func \)](#)

6.48.2 Member Data Documentation

6.48.2.1 [UInt32 BoapFuncEntry::ocmd](#)

6.48.2.2 [BUInt32 BoapFuncEntry::ocmd](#)

6.48.2.3 [BoapFunc BoapFuncEntry::ofunc](#)

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

- [Boap.h](#)
- [BoapSimple.h](#)
- [Boap.cpp](#)
- [BoapSimple.cc](#)

6.49 BoapMcClientObject Class Reference

```
#include <BoapMc.h>
```

Public Member Functions

- [BoapMcClientObject \(BComms &comms\)](#)
- virtual [~BoapMcClientObject \(\)](#)
- void [setAddress \(BUInt8 addressTo, BUInt8 addressFrom\)](#)
- [BUInt32 getApiVersion \(\)](#)

Returns the API version.

Protected Member Functions

- [BError performCall \(\)](#)
Performs a RPC call to the named service.
- [BError performSend \(\)](#)
Performs a send to the named service.
- [BError performRecv \(\)](#)
Performs a receive.

Protected Attributes

- [BUInt32 oapiVersion](#)
- [BComms & ocomms](#)
- [BUInt8 oaddressTo](#)
- [BUInt8 oaddressFrom](#)
- [BoapMcPacket opacket](#)

6.49.1 Constructor & Destructor Documentation

6.49.1.1 **BoapMcClientObject::BoapMcClientObject (BComms & comms)**

6.49.1.2 **BoapMcClientObject::~~BoapMcClientObject ()** [virtual]

6.49.2 Member Function Documentation

6.49.2.1 **BUInt32 BoapMcClientObject::getApiVersion ()**

Returns the API version.

6.49.2.2 **BError BoapMcClientObject::performCall ()** [protected]

Performs a RPC call to the named service.

6.49.2.3 **BError BoapMcClientObject::performRecv ()** [protected]

Performs a receive.

6.49.2.4 **BError BoapMcClientObject::performSend ()** [protected]

Performs a send to the named service.

6.49.2.5 **void BoapMcClientObject::setAddress (BUInt8 addressTo, BUInt8 addressFrom)**

6.49.3 Member Data Documentation

6.49.3.1 **BUInt8 BoapMcClientObject::oaddressFrom** [protected]

6.49.3.2 **BUInt8 BoapMcClientObject::oaddressTo** [protected]

6.49.3.3 **BUInt32 BoapMcClientObject::oapiVersion** [protected]

6.49.3.4 **BComms& BoapMcClientObject::ocomms** [protected]

6.49.3.5 **BoapMcPacket BoapMcClientObject::opacket** [protected]

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

- [BoapMc.h](#)
- [BoapMc.cpp](#)

6.50 BoapMcComms Class Reference

```
#include <BoapMc.h>
```

Public Member Functions

- [BoapMcComms](#) ([Bool](#) threaded=0, [BUInt](#) rxQueueSize=4)
- virtual [~BoapMcComms](#) ()
- void [setCommsMode](#) ([Bool](#) slave, [BUInt](#) txQueueSize)

Sets slave mode.
- void [setComms](#) ([BComms](#) &comms)

Sets the communications interface to use.
- void [setComms](#) ([BComms](#) *comms)

Sets the communications interface to use.
- void [setAddress](#) ([BUInt8](#) addressTo, [BUInt8](#) addressFrom)

Sets the to and from addresses.
- [BUInt32](#) [getApiVersion](#) ()

Returns the API version.
- [BUInt32](#) [setTimeout](#) ([BUInt32](#) timeoutUs)

Sets the call timeout returning the current value.
- virtual [BError](#) [processRx](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))

Process any RX packets queuing them as needed.
- virtual [BError](#) [processRequests](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))

Check and process all requests.
- virtual [BError](#) [processRequest](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))

Check and process any request.
- virtual [BError](#) [processPacket](#) ([BoapMcPacket](#) &rx, [BoapMcPacket](#) &tx)

Process a recieved packet.

Protected Member Functions

- [BError](#) [performCall](#) ()

Performs a RPC call to the remote side.
- [BError](#) [performSend](#) ()

Performs a RPC send to the remote side.
- [BError](#) [packetSend](#) ([BoapMcPacket](#) &packet)

Receives a packet.
- [BError](#) [packetRecv](#) ([BoapMcPacket](#) &packet)

Receives a packet.

Protected Attributes

- [Bool](#) othreaded
- [BMutex](#) olockCall

Lock for RPC calls. Only one at a time.
- [BMutex](#) olockTx

Lock for TX.
- [BComms](#) * ocomms
- [BUInt32](#) oapiVersion
- [Bool](#) oslave

- Set slave mode.*
- [BUInt32 otimeout](#)
 - The timeout in us for calls.*
- [BUInt8 oaddressTo](#)
- [BUInt8 oaddressFrom](#)
- [BoapMcPacket opacket](#)
 - Packet RX buffer.*
- [BoapMcPacket opacketTx](#)
 - Packet TX buffer for calls.*
- [BoapMcPacket opacketRx](#)
 - Packet RX buffer for calls.*
- [BSemaphore opacketRxSema](#)
 - Wait RX semaphore.*
- [BoapMcPacket opacketReqTx](#)
 - Packet TX buffer for requests.*
- [BoapMcPacket opacketReqRx](#)
 - Packet RX buffer for requests.*
- [BQueue< BoapMcPacket > opacketReqQueue](#)
 - Packet RX buffer queue for requests.*
- [BFifo< BoapMcPacket > opacketTxQueue](#)
 - Packet TX Queue.*
- [BSemaphoreCount opacketTxQueueWriteNum](#)
 - Packet TX Queue number.*
- [BSemaphore opacketTxSema](#)
 - Wait for TX semaphore.*

6.50.1 Constructor & Destructor Documentation

6.50.1.1 **BoapMcComms::BoapMcComms** (**Bool** *threaded* = 0, **BUInt** *rxQueueSize* = 4)

6.50.1.2 **BoapMcComms::~~BoapMcComms** () [virtual]

6.50.2 Member Function Documentation

6.50.2.1 **BUInt32** **BoapMcComms::getApiVersion** ()

Returns the API version.

6.50.2.2 **BError** **BoapMcComms::packetRecv** (**BoapMcPacket** & *packet*) [protected]

Receives a packet.

6.50.2.3 **BError** **BoapMcComms::packetSend** (**BoapMcPacket** & *packet*) [protected]

Receives a packet.

6.50.2.4 **BError** **BoapMcComms::performCall** () [protected]

Performs a RPC call to the remote side.

6.50.2.5 BError BoapMcComms::performSend () [protected]

Performs a RPC send to the remote side.

6.50.2.6 BError BoapMcComms::processPacket (BoapMcPacket & rx, BoapMcPacket & tx) [virtual]

Process a recieved packet.

6.50.2.7 BError BoapMcComms::processRequest (BTimeout timeoutUs = BTimeoutForever) [virtual]

Check and process any request.

6.50.2.8 BError BoapMcComms::processRequests (BTimeout timeoutUs = BTimeoutForever) [virtual]

Check and process all requests.

6.50.2.9 BError BoapMcComms::processRx (BTimeout timeoutUs = BTimeoutForever) [virtual]

Process any RX packets queuing them as needed.

!!! This should wait on comms for timeoutUs !!!

6.50.2.10 void BoapMcComms::setAddress (BUInt8 addressTo, BUInt8 addressFrom)

Sets the to and from addresses.

6.50.2.11 void BoapMcComms::setComms (BComms & comms)

Sets the communications interface to use.

6.50.2.12 void BoapMcComms::setComms (BComms * comms)

Sets the communications interface to use.

6.50.2.13 void BoapMcComms::setCommsMode (Bool slave, BUInt txQueueSize)

Sets slave mode.

6.50.2.14 BUInt32 BoapMcComms::setTimeout (BUInt32 timeoutUs)

Sets the call timeout returning the current value.

6.50.3 Member Data Documentation**6.50.3.1 BUInt8 BoapMcComms::oaddressFrom [protected]****6.50.3.2 BUInt8 BoapMcComms::oaddressTo [protected]****6.50.3.3 BUInt32 BoapMcComms::oapiVersion [protected]**

6.50.3.4 **BComms*** BoapMcComms::ocomms [protected]

6.50.3.5 **BMutex** BoapMcComms::olockCall [protected]

Lock for RPC calls. Only one at a time.

6.50.3.6 **BMutex** BoapMcComms::olockTx [protected]

Lock for TX.

6.50.3.7 **BoapMcPacket** BoapMcComms::opacket [protected]

Packet RX buffer.

6.50.3.8 **BQueue<BoapMcPacket>** BoapMcComms::opacketReqQueue [protected]

Packet RX buffer queue for requests.

6.50.3.9 **BoapMcPacket** BoapMcComms::opacketReqRx [protected]

Packet RX buffer for requests.

6.50.3.10 **BoapMcPacket** BoapMcComms::opacketReqTx [protected]

Packet TX buffer for requests.

6.50.3.11 **BoapMcPacket** BoapMcComms::opacketRx [protected]

Packet RX buffer for calls.

6.50.3.12 **BSemaphore** BoapMcComms::opacketRxSema [protected]

Wait RX semaphore.

6.50.3.13 **BoapMcPacket** BoapMcComms::opacketTx [protected]

Packet TX buffer for calls.

6.50.3.14 **BFifo<BoapMcPacket>** BoapMcComms::opacketTxQueue [protected]

Packet TX Queue.

6.50.3.15 **BSemaphoreCount** BoapMcComms::opacketTxQueueWriteNum [protected]

Packet TX Queue number.

6.50.3.16 **BSemaphore** BoapMcComms::opacketTxSema [protected]

Wait for TX semaphore.

6.50.3.17 **Bool** BoapMcComms::oslave [protected]

Set slave mode.

6.50.3.18 **Bool** BoapMcComms::othreaded [protected]

6.50.3.19 **BUInt32** BoapMcComms::otimeout [protected]

The timeout in us for calls.

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

- [BoapMc.h](#)
- [BoapMc.cpp](#)

6.51 BoapMcPacket Class Reference

```
#include <BoapMc.h>
```

Public Attributes

- [BoapMcPacketHead](#) head
- char [data](#) [256*sizeof([BoapMcPacketHead](#))]

6.51.1 Member Data Documentation

6.51.1.1 char BoapMcPacket::data[256*sizeof([BoapMcPacketHead](#))]

6.51.1.2 **BoapMcPacketHead** BoapMcPacket::head

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

- [BoapMc.h](#)

6.52 BoapMcPacketHead Struct Reference

```
#include <BoapMc.h>
```

Public Attributes

- [BUInt8](#) length
- [BUInt8](#) addressTo
- [BUInt8](#) addressFrom
- [BUInt8](#) cmd
- [BUInt16](#) error
- [BUInt16](#) checksum

6.52.1 Member Data Documentation

6.52.1.1 **BUInt8** BoapMcPacketHead::addressFrom

6.52.1.2 **BUInt8** BoapMcPacketHead::addressTo

6.52.1.3 **BUInt16** BoapMcPacketHead::checksum

6.52.1.4 **BUInt8** BoapMcPacketHead::cmd

6.52.1.5 **BUInt16** BoapMcPacketHead::error

6.52.1.6 **BUInt8** BoapMcPacketHead::length

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

- [BoapMc.h](#)

6.53 BoapMcServiceObject Class Reference

```
#include <BoapMc.h>
```

Public Member Functions

- [BoapMcServiceObject](#) ()
- virtual [~BoapMcServiceObject](#) ()
- virtual [BError process](#) ([BoapMcPacket](#) &rx, [BoapMcPacket](#) &tx)
- virtual [BError processEvent](#) ([BoapMcPacket](#) &rx)

Protected Member Functions

- [BError sendEvent](#) ([BoapMcPacket](#) &tx)

Protected Attributes

- [BUInt32 oapiVersion](#)

6.53.1 Constructor & Destructor Documentation

6.53.1.1 [BoapMcServiceObject::BoapMcServiceObject](#) ()

6.53.1.2 [BoapMcServiceObject::~~BoapMcServiceObject](#) () [virtual]

6.53.2 Member Function Documentation

6.53.2.1 [BError BoapMcServiceObject::process](#) ([BoapMcPacket](#) & rx, [BoapMcPacket](#) & tx) [virtual]

6.53.2.2 [BError BoapMcServiceObject::processEvent](#) ([BoapMcPacket](#) & rx) [virtual]

6.53.2.3 [BError BoapMcServiceObject::sendEvent](#) ([BoapMcPacket](#) & tx) [protected]

6.53.3 Member Data Documentation

6.53.3.1 BUInt32 BoapMcServiceObject::oapiVersion [protected]

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

- [BoapMc.h](#)
- [BoapMc.cpp](#)

6.54 BoapMcSignalObject Class Reference

```
#include <BoapMc.h>
```

Public Member Functions

- [BoapMcSignalObject](#) ([BComms](#) &comms)

Protected Member Functions

- [BError performSend](#) ([BoapMcPacket](#) &tx)

Protected Attributes

- [BComms](#) & [ocomms](#)

6.54.1 Constructor & Destructor Documentation

6.54.1.1 BoapMcSignalObject::BoapMcSignalObject ([BComms](#) & *comms*)

6.54.2 Member Function Documentation

6.54.2.1 BError BoapMcSignalObject::performSend ([BoapMcPacket](#) & *tx*) [protected]

6.54.3 Member Data Documentation

6.54.3.1 BComms& BoapMcSignalObject::ocomms [protected]

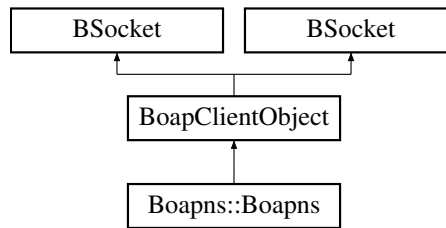
The documentation for this class was generated from the following files:

- [BoapMc.h](#)
- [BoapMc.cpp](#)

6.55 Boapns::Boapns Class Reference

```
#include <BoapnsC.h>
```

Inheritance diagram for Boapns::Boapns:



Public Member Functions

- [Boapns](#) ([BString](#) name="")
- [BError getVersion](#) ([BString](#) &version)
- [BError getEntryList](#) ([BList](#)< [BoapEntry](#) > &entryList)
- [BError getEntry](#) ([BString](#) name, [BoapEntry](#) &entry)
- [BError addEntry](#) ([BoapEntry](#) entry)
- [BError delEntry](#) ([BString](#) name)
- [BError getNewName](#) ([BString](#) &name)

Additional Inherited Members

6.55.1 Constructor & Destructor Documentation

6.55.1.1 [Boapns::Boapns::Boapns](#) ([BString](#) name = " ")

6.55.2 Member Function Documentation

6.55.2.1 [BError Boapns::Boapns::addEntry](#) ([BoapEntry](#) entry)

6.55.2.2 [BError Boapns::Boapns::delEntry](#) ([BString](#) name)

6.55.2.3 [BError Boapns::Boapns::getEntry](#) ([BString](#) name, [BoapEntry](#) & entry)

6.55.2.4 [BError Boapns::Boapns::getEntryList](#) ([BList](#)< [BoapEntry](#) > & entryList)

6.55.2.5 [BError Boapns::Boapns::getNewName](#) ([BString](#) & name)

6.55.2.6 [BError Boapns::Boapns::getVersion](#) ([BString](#) & version)

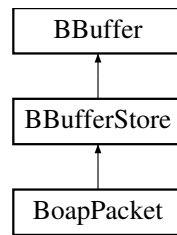
The documentation for this class was generated from the following files:

- [BoapnsC.h](#)
- [BoapnsC.cpp](#)

6.56 BoapPacket Class Reference

```
#include <BoapSimple.h>
```

Inheritance diagram for BoapPacket:



Public Member Functions

- [BoapPacket](#) ()
- [~BoapPacket](#) ()
- [BUInt32](#) [getCmd](#) ()
- [int](#) [peekHead](#) ([BoapPacketHead](#) &head)
- [int](#) [pushHead](#) ([BoapPacketHead](#) &head)
- [int](#) [popHead](#) ([BoapPacketHead](#) &head)
- [void](#) [updateHead](#) ()
- [BoapPacket](#) ()
- [~BoapPacket](#) ()
- [int](#) [resize](#) ([int](#) size)
- [BError](#) [setData](#) ([void](#) *data, [int](#) nbytes)
- [int](#) [nbytes](#) ()
- [char](#) * [data](#) ()
- [int](#) [pushHead](#) ([BoapPacketHead](#) &head)
- [int](#) [push](#) ([Int8](#) v)
- [int](#) [push](#) ([UInt8](#) v)
- [int](#) [push](#) ([Int16](#) v)
- [int](#) [push](#) ([UInt16](#) v)
- [int](#) [push](#) ([Int32](#) v)
- [int](#) [push](#) ([UInt32](#) v)
- [int](#) [push](#) ([BString](#) &v)
- [int](#) [push](#) ([Double](#) v)
- [int](#) [push](#) ([BError](#) &v)
- [int](#) [push](#) ([UInt32](#) nBytes, [const](#) [void](#) *data)
- [int](#) [popHead](#) ([BoapPacketHead](#) &head)
- [int](#) [pop](#) ([Int8](#) &v)
- [int](#) [pop](#) ([UInt8](#) &v)
- [int](#) [pop](#) ([Int16](#) &v)
- [int](#) [pop](#) ([UInt16](#) &v)
- [int](#) [pop](#) ([Int32](#) &v)
- [int](#) [pop](#) ([UInt32](#) &v)
- [int](#) [pop](#) ([BString](#) &v)
- [int](#) [pop](#) ([Double](#) &v)
- [int](#) [pop](#) ([BError](#) &v)
- [int](#) [pop](#) ([UInt32](#) nBytes, [void](#) *data)

Private Member Functions

- [void](#) [updateLen](#) ()

Private Attributes

- int [osize](#)
- int [onbytes](#)
- char * [odata](#)
- int [opos](#)

Additional Inherited Members

6.56.1 Constructor & Destructor Documentation

6.56.1.1 `BoapPacket::BoapPacket ()`

6.56.1.2 `BoapPacket::~~BoapPacket ()`

6.56.1.3 `BoapPacket::BoapPacket ()`

6.56.1.4 `BoapPacket::~~BoapPacket ()`

6.56.2 Member Function Documentation

6.56.2.1 `char * BoapPacket::data ()`

6.56.2.2 `BUInt32 BoapPacket::getCmd ()`

6.56.2.3 `int BoapPacket::nbytes ()`

6.56.2.4 `int BoapPacket::peekHead (BoapPacketHead & head)`

6.56.2.5 `int BoapPacket::pop (Int8 & v)`

6.56.2.6 `int BoapPacket::pop (UInt8 & v)`

6.56.2.7 `int BoapPacket::pop (Int16 & v)`

6.56.2.8 `int BoapPacket::pop (UInt16 & v)`

6.56.2.9 `int BoapPacket::pop (Int32 & v)`

6.56.2.10 `int BoapPacket::pop (UInt32 & v)`

6.56.2.11 `int BoapPacket::pop (BString & v)`

6.56.2.12 `int BoapPacket::pop (Double & v)`

6.56.2.13 `int BoapPacket::pop (BError & v)`

6.56.2.14 `int BoapPacket::pop (UInt32 nBytes, void * data)`

6.56.2.15 `int BoapPacket::popHead (BoapPacketHead & head)`

6.56.2.16 `int BoapPacket::popHead (BoapPacketHead & head)`

6.56.2.17 `int BoapPacket::push (Int8 v)`

- 6.56.2.18 `int BoapPacket::push (UInt8 v)`
- 6.56.2.19 `int BoapPacket::push (Int16 v)`
- 6.56.2.20 `int BoapPacket::push (UInt16 v)`
- 6.56.2.21 `int BoapPacket::push (Int32 v)`
- 6.56.2.22 `int BoapPacket::push (UInt32 v)`
- 6.56.2.23 `int BoapPacket::push (BString & v)`
- 6.56.2.24 `int BoapPacket::push (Double v)`
- 6.56.2.25 `int BoapPacket::push (BError & v)`
- 6.56.2.26 `int BoapPacket::push (UInt32 nBytes, const void * data)`
- 6.56.2.27 `int BoapPacket::pushHead (BoapPacketHead & head)`
- 6.56.2.28 `int BoapPacket::pushHead (BoapPacketHead & head)`
- 6.56.2.29 `int BoapPacket::resize (int size)`
- 6.56.2.30 `BError BoapPacket::setData (void * data, int nbytes)`
- 6.56.2.31 `void BoapPacket::updateHead ()`
- 6.56.2.32 `void BoapPacket::updateLen () [private]`

6.56.3 Member Data Documentation

- 6.56.3.1 `char* BoapPacket::odata [private]`
- 6.56.3.2 `int BoapPacket::onbytes [private]`
- 6.56.3.3 `int BoapPacket::opos [private]`
- 6.56.3.4 `int BoapPacket::osize [private]`

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

- [Boap.h](#)
- [BoapSimple.h](#)
- [Boap.cpp](#)
- [BoapSimple.cc](#)

6.57 BoapPacketHead Struct Reference

```
#include <BoapSimple.h>
```

Public Attributes

- [BUInt32 type](#)

- [BUInt32 length](#)
- [BUInt32 service](#)
- [BUInt32 cmd](#)
- [UInt32 length](#)
- [BoapType type](#)
- [BoapService service](#)
- [UInt32 cmd](#)
- [UInt32 reserved \[12\]](#)

6.57.1 Member Data Documentation

6.57.1.1 [UInt32 BoapPacketHead::cmd](#)

6.57.1.2 [BUInt32 BoapPacketHead::cmd](#)

6.57.1.3 [UInt32 BoapPacketHead::length](#)

6.57.1.4 [BUInt32 BoapPacketHead::length](#)

6.57.1.5 [UInt32 BoapPacketHead::reserved\[12\]](#)

6.57.1.6 [BoapService BoapPacketHead::service](#)

6.57.1.7 [BUInt32 BoapPacketHead::service](#)

6.57.1.8 [BUInt32 BoapPacketHead::type](#)

6.57.1.9 [BoapType BoapPacketHead::type](#)

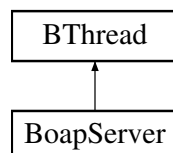
The documentation for this struct was generated from the following files:

- [Boap.h](#)
- [BoapSimple.h](#)

6.58 BoapServer Class Reference

```
#include <BoapSimple.h>
```

Inheritance diagram for BoapServer:



Public Types

- enum { [NOTHREADS](#) =0, [THREADED](#) =1 }

Public Member Functions

- [BoapServer](#) ()
- virtual [~BoapServer](#) ()
- virtual [BError](#) [init](#) ([BString](#) boapNsHost="", int port=0, int threaded=0, int isBoapns=0)
- virtual [BError](#) [run](#) (int inThread=0)
- virtual [BError](#) [process](#) ([BoapServerConnection](#) *conn, [BoapPacket](#) &rx, [BoapPacket](#) &tx)
- virtual [BError](#) [processEvent](#) ([BoapPacket](#) &rx)
- virtual [BError](#) [addObject](#) ([BoapServiceObject](#) *object)
- virtual [BError](#) [sendEvent](#) ([BoapPacket](#) &tx)
- virtual [BError](#) [processEvent](#) (int fd)
- virtual void [clientGone](#) ([BoapServerConnection](#) *client)
- [BSocket](#) & [getSocket](#) ()
- [BSocket](#) & [getEventSocket](#) ()
- [BString](#) [getHostName](#) ()
- int [getConnectionsNumber](#) ()
- virtual [BoapServerConnection](#) * [newConnection](#) (int fd, [BSocketAddressINET](#) address)
- [BoapServer](#) ()
- [BError](#) [init](#) (int boapNs=0)
- [BError](#) [run](#) ()
- [BError](#) [processEvent](#) ([BoapPacket](#) &rx)
- [BError](#) [addObject](#) ([BoapServiceObject](#) *object)
- [BError](#) [process](#) (int fd)
- [BError](#) [sendEvent](#) ([BoapPacket](#) &tx)
- [BSocket](#) & [getSocket](#) ()
- [BSocket](#) & [getEventSocket](#) ()
- [BError](#) [processEvent](#) (int fd)
- [BString](#) [getHostName](#) ()

Public Attributes

- [BUInt64](#) onumOperations

Private Member Functions

- void * [function](#) ()

Private Attributes

- int [othreaded](#)
- int [oisBoapns](#)
- [Boapns::Boapns](#) * [oboapns](#)
- [BList](#)< [BoapServerConnection](#) * > [oclients](#)
- [BEvent1Int](#) [oclientGoneEvent](#)
- [BList](#)< [BoapServiceEntry](#) > [oservices](#)
- [BPoll](#) [opoll](#)
- [BSocket](#) [onet](#)
- [BSocket](#) [onetEvent](#)
- [BSocketAddressINET](#) [onetEventAddress](#)
- [BString](#) [ohostName](#)
- int [oboapNs](#)
- [BoapPacket](#) [orx](#)
- [BoapPacket](#) [otx](#)

6.58.1 Member Enumeration Documentation

6.58.1.1 anonymous enum

Enumerator

NOTHEADS

THREADED

6.58.2 Constructor & Destructor Documentation

6.58.2.1 BoapServer::BoapServer ()

6.58.2.2 BoapServer::~~BoapServer () [virtual]

6.58.2.3 BoapServer::BoapServer ()

6.58.3 Member Function Documentation

6.58.3.1 BError BoapServer::addObject (BoapServiceObject * *object*)

6.58.3.2 BError BoapServer::addObject (BoapServiceObject * *object*) [virtual]

6.58.3.3 void BoapServer::clientGone (BoapServerConnection * *client*) [virtual]

6.58.3.4 void * BoapServer::function () [private],[virtual]

Reimplemented from [BThread](#).

6.58.3.5 int BoapServer::getConnectionsNumber ()

6.58.3.6 BSocket& BoapServer::getEventSocket ()

6.58.3.7 BSocket & BoapServer::getEventSocket ()

6.58.3.8 BString BoapServer::getHostName ()

6.58.3.9 BString BoapServer::getHostName ()

6.58.3.10 BSocket& BoapServer::getSocket ()

6.58.3.11 BSocket & BoapServer::getSocket ()

6.58.3.12 BError BoapServer::init (int *boapNs* = 0)

6.58.3.13 BError BoapServer::init (BString *boapNsHost* = "", int *port* = 0, int *threaded* = 0, int *isBoapns* = 0) [virtual]

6.58.3.14 BoapServerConnection * BoapServer::newConnection (int *fd*, BSocketAddressINET *address*) [virtual]

6.58.3.15 BError BoapServer::process (int *fd*)

6.58.3.16 BError BoapServer::process (BoapServerConnection * *conn*, BoapPacket & *rx*, BoapPacket & *tx*) [virtual]

- 6.58.3.17 **BError** BoapServer::processEvent (**BoapPacket** & rx)
- 6.58.3.18 **BError** BoapServer::processEvent (int fd)
- 6.58.3.19 **BError** BoapServer::processEvent (**BoapPacket** & rx) [virtual]
- 6.58.3.20 **BError** BoapServer::processEvent (int fd) [virtual]
- 6.58.3.21 **BError** BoapServer::run ()
- 6.58.3.22 **BError** BoapServer::run (int *inThread* = 0) [virtual]
- 6.58.3.23 **BError** BoapServer::sendEvent (**BoapPacket** & tx)
- 6.58.3.24 **BError** BoapServer::sendEvent (**BoapPacket** & tx) [virtual]

6.58.4 Member Data Documentation

- 6.58.4.1 int BoapServer::oobapNs [private]
- 6.58.4.2 **Boapns::Boapns*** BoapServer::oobapns [private]
- 6.58.4.3 **BEvent1Int** BoapServer::oclientGoneEvent [private]
- 6.58.4.4 **BList<BoapServerConnection*>** BoapServer::oclients [private]
- 6.58.4.5 **BString** BoapServer::ohostName [private]
- 6.58.4.6 int BoapServer::oisBoapns [private]
- 6.58.4.7 **BSocket** BoapServer::onet [private]
- 6.58.4.8 **BSocket** BoapServer::onetEvent [private]
- 6.58.4.9 **BSocketAddressINET** BoapServer::onetEventAddress [private]
- 6.58.4.10 **BUInt64** BoapServer::onumOperations
- 6.58.4.11 **BPoll** BoapServer::opoll [private]
- 6.58.4.12 **BoapPacket** BoapServer::orx [private]
- 6.58.4.13 **BList< BoapServiceEntry >** BoapServer::oservices [private]
- 6.58.4.14 int BoapServer::othreaded [private]
- 6.58.4.15 **BoapPacket** BoapServer::otx [private]

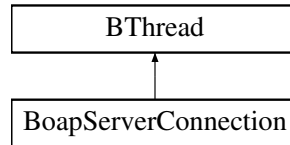
The documentation for this class was generated from the following files:

- [Boap.h](#)
- [BoapSimple.h](#)
- [Boap.cpp](#)
- [BoapSimple.cc](#)

6.59 BoapServerConnection Class Reference

```
#include <Boap.h>
```

Inheritance diagram for BoapServerConnection:



Public Member Functions

- [BoapServerConnection](#) ([BoapServer](#) &boapServer, int fd)
- virtual [~BoapServerConnection](#) ()
- virtual [BError](#) init ()
Initialise connection.
- virtual [BError](#) process ()
- virtual [BSocket](#) & [getSocket](#) ()
- virtual void [setMaxLength](#) ([BUInt32](#) maxLength)
- virtual [BError](#) [getHead](#) ([BoapPacketHead](#) &head)
- virtual [BError](#) [validate](#) ()
Validate the connection.

Private Member Functions

- void * [function](#) ()

Private Attributes

- [BoapServer](#) & oboapServer
- [BSocket](#) osocket
- [BoapPacket](#) orx
- [BoapPacket](#) otx
- [BUInt32](#) omaxLength

6.59.1 Constructor & Destructor Documentation

6.59.1.1 [BoapServerConnection::BoapServerConnection](#) ([BoapServer](#) & boapServer, int fd)

6.59.1.2 [BoapServerConnection::~BoapServerConnection](#) () [virtual]

6.59.2 Member Function Documentation

6.59.2.1 void * [BoapServerConnection::function](#) () [private],[virtual]

Reimplemented from [BThread](#).

6.59.2.2 **BError** BoapServerConnection::getHead (**BoapPacketHead** & *head*) [virtual]

6.59.2.3 **BSocket** & BoapServerConnection::getSocket () [virtual]

6.59.2.4 **BError** BoapServerConnection::init () [virtual]

Initialise connection.

6.59.2.5 **BError** BoapServerConnection::process () [virtual]

6.59.2.6 void BoapServerConnection::setMaxLength (**BUInt32** *maxLength*) [virtual]

6.59.2.7 **BError** BoapServerConnection::validate () [virtual]

Validate the connection.

6.59.3 Member Data Documentation

6.59.3.1 **BoapServer**& BoapServerConnection::oBoapServer [private]

6.59.3.2 **BUInt32** BoapServerConnection::oMaxLength [private]

6.59.3.3 **BoapPacket** BoapServerConnection::orx [private]

6.59.3.4 **BSocket** BoapServerConnection::oSocket [private]

6.59.3.5 **BoapPacket** BoapServerConnection::otx [private]

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

- [Boap.h](#)
- [Boap.cpp](#)

6.60 BoapServiceEntry Class Reference

```
#include <BoapSimple.h>
```

Public Member Functions

- [BoapServiceEntry](#) ([BoapService](#) service=0, [BoapServiceObject](#) *object=0)
- [BoapServiceEntry](#) ([BoapService](#) service=0, [BoapServiceObject](#) *object=0)

Public Attributes

- [BoapService](#) oservice
- [BoapServiceObject](#) * oobject

6.60.1 Constructor & Destructor Documentation

6.60.1.1 **BoapServiceEntry**::**BoapServiceEntry** (**BoapService** *service* = 0, **BoapServiceObject** * *object* = 0)
[inline]

6.60.1.2 `BoapServiceEntry::BoapServiceEntry (BoapService service = 0, BoapServiceObject * object = 0)`
`[inline]`

6.60.2 Member Data Documentation

6.60.2.1 `BoapServiceObject * BoapServiceEntry::oobject`

6.60.2.2 `BoapService BoapServiceEntry::oservice`

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

- [Boap.h](#)
- [BoapSimple.h](#)

6.61 BoapServiceObject Class Reference

```
#include <BoapSimple.h>
```

Public Member Functions

- [BoapServiceObject](#) ([BoapServer](#) &server, [BString](#) name="")
- virtual [~BoapServiceObject](#) ()
- [BError](#) setName ([BString](#) name)
- [BError](#) sendEvent ([BString](#) signalName, [BInt32](#) arg)
- virtual [BError](#) processEvent ([BString](#) objectName, [BString](#) name, [BInt32](#) arg)
- [BString](#) name ()
- [BError](#) doPing ([BoapServerConnection](#) *conn, [BoapPacket](#) &rx, [BoapPacket](#) &tx)
- [BError](#) doConnectionPriority ([BoapServerConnection](#) *conn, [BoapPacket](#) &rx, [BoapPacket](#) &tx)
- [BError](#) process ([BoapServerConnection](#) *conn, [BoapPacket](#) &rx, [BoapPacket](#) &tx)
- virtual [BError](#) processEvent ([BoapPacket](#) &rx)
- [BoapServiceObject](#) ([BoapServer](#) &server, [BString](#) name)
- virtual [~BoapServiceObject](#) ()
- [BError](#) sendEvent ([BString](#) signalName, [Int32](#) arg)
- virtual [BError](#) processEvent ([BString](#) objectName, [BString](#) name, [Int32](#) arg)
- [BString](#) name ()
- [BError](#) process ([BoapPacket](#) &rx, [BoapPacket](#) &tx)
- virtual [BError](#) processEvent ([BoapPacket](#) &rx)

Protected Member Functions

- [BError](#) sendEvent ([BoapPacket](#) &tx)
- [BError](#) sendEvent ([BoapPacket](#) &tx)

Protected Attributes

- [BoapServer](#) & oserver
- [BString](#) oname
- [BUInt32](#) oapiVersion
- [BList](#)< [BoapFuncEntry](#) > ofuncList

6.61.1 Constructor & Destructor Documentation

6.61.1.1 **BoapServiceObject::BoapServiceObject** (**BoapServer** & *server*, **BString** *name* = " ")

6.61.1.2 **BoapServiceObject::~~BoapServiceObject** () [virtual]

6.61.1.3 **BoapServiceObject::BoapServiceObject** (**BoapServer** & *server*, **BString** *name*)

6.61.1.4 **virtual BoapServiceObject::~~BoapServiceObject** () [virtual]

6.61.2 Member Function Documentation

6.61.2.1 **BError BoapServiceObject::doConnectionPriority** (**BoapServerConnection** * *conn*, **BoapPacket** & *rx*, **BoapPacket** & *tx*)

6.61.2.2 **BError BoapServiceObject::doPing** (**BoapServerConnection** * *conn*, **BoapPacket** & *rx*, **BoapPacket** & *tx*)

6.61.2.3 **BString BoapServiceObject::name** ()

6.61.2.4 **BString BoapServiceObject::name** ()

6.61.2.5 **BError BoapServiceObject::process** (**BoapPacket** & *rx*, **BoapPacket** & *tx*)

6.61.2.6 **BError BoapServiceObject::process** (**BoapServerConnection** * *conn*, **BoapPacket** & *rx*, **BoapPacket** & *tx*)

6.61.2.7 **virtual BError BoapServiceObject::processEvent** (**BString** *objectName*, **BString** *name*, **Int32** *arg*) [virtual]

6.61.2.8 **virtual BError BoapServiceObject::processEvent** (**BoapPacket** & *rx*) [virtual]

6.61.2.9 **BError BoapServiceObject::processEvent** (**BString** *objectName*, **BString** *name*, **BInt32** *arg*) [virtual]

6.61.2.10 **BError BoapServiceObject::processEvent** (**BoapPacket** & *rx*) [virtual]

6.61.2.11 **BError BoapServiceObject::sendEvent** (**BString** *signalName*, **Int32** *arg*)

6.61.2.12 **BError BoapServiceObject::sendEvent** (**BoapPacket** & *tx*) [protected]

6.61.2.13 **BError BoapServiceObject::sendEvent** (**BString** *signalName*, **BInt32** *arg*)

6.61.2.14 **BError BoapServiceObject::sendEvent** (**BoapPacket** & *tx*) [protected]

6.61.2.15 **BError BoapServiceObject::setName** (**BString** *name*)

6.61.3 Member Data Documentation

6.61.3.1 **BUInt32 BoapServiceObject::oapiVersion** [protected]

6.61.3.2 **BList< BoapFuncEntry > BoapServiceObject::ofuncList** [protected]

6.61.3.3 **BString BoapServiceObject::oname** [protected]

6.61.3.4 **BoapServer & BoapServiceObject::oserver** [protected]

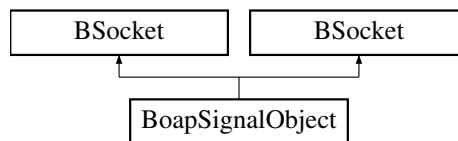
The documentation for this class was generated from the following files:

- [Boap.h](#)
- [BoapSimple.h](#)
- [Boap.cpp](#)
- [BoapSimple.cc](#)

6.62 BoapSignalObject Class Reference

```
#include <BoapSimple.h>
```

Inheritance diagram for BoapSignalObject:



Public Member Functions

- [BoapSignalObject \(\)](#)
- [BoapSignalObject \(\)](#)

Protected Member Functions

- [BError performSend \(BoapPacket &tx\)](#)
- [BError performSend \(BoapPacket &tx\)](#)

Protected Attributes

- [BoapPacket otx](#)
- [BoapPacket orx](#)

Additional Inherited Members

6.62.1 Constructor & Destructor Documentation

6.62.1.1 [BoapSignalObject::BoapSignalObject \(\)](#)

6.62.1.2 [BoapSignalObject::BoapSignalObject \(\)](#)

6.62.2 Member Function Documentation

6.62.2.1 [BError BoapSignalObject::performSend \(BoapPacket & tx \)](#) [protected]

6.62.2.2 [BError BoapSignalObject::performSend \(BoapPacket & tx \)](#) [protected]

6.62.3 Member Data Documentation

6.62.3.1 [BoapPacket BoapSignalObject::orx](#) [protected]

6.62.3.2 BoapPacket BoapSignalObject::otx [protected]

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

- [Boap.h](#)
- [BoapSimple.h](#)
- [Boap.cpp](#)
- [BoapSimple.cc](#)

6.63 BObj Class Reference

```
#include <BObj.h>
```

Public Member Functions

- [BObj](#) ()
- virtual [~BObj](#) ()
- virtual const char * [getType](#) () const
- virtual const [BObjMember](#) * [getMembers](#) () const
- virtual [BError](#) [getMembers](#) ([BDictString](#) &members)
- virtual [BError](#) [getMember](#) ([BString](#) name, [BString](#) &value)
- virtual [BError](#) [setMembers](#) ([BDictString](#) &members)
- virtual [BError](#) [setMember](#) ([BString](#) name, [BString](#) value)
- virtual void [membersPrint](#) () const
- *Prints out members.*
- virtual [BString](#) [getDebugString](#) ()
- *Returns contents as a debug string.*

6.63.1 Constructor & Destructor Documentation

6.63.1.1 [BObj::BObj](#) ()

6.63.1.2 [BObj::~~BObj](#) () [virtual]

6.63.2 Member Function Documentation

6.63.2.1 [BString](#) [BObj::getDebugString](#) () [virtual]

Returns contents as a debug string.

6.63.2.2 [BError](#) [BObj::getMember](#) ([BString](#) name, [BString](#) & value) [virtual]

6.63.2.3 const [BObjMember](#) * [BObj::getMembers](#) () const [virtual]

6.63.2.4 [BError](#) [BObj::getMembers](#) ([BDictString](#) & members) [virtual]

6.63.2.5 const char * [BObj::getType](#) () const [virtual]

6.63.2.6 void [BObj::membersPrint](#) () const [virtual]

Prints out members.

6.63.2.7 **BError** BObj::setMember (BString *name*, BString *value*) [virtual]

6.63.2.8 **BError** BObj::setMembers (BDictString & *members*) [virtual]

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

- [BObj.h](#)
- [BObj.cpp](#)

6.64 BObjMember Struct Reference

```
#include <BTypes.h>
```

Public Attributes

- [BType](#) type
- [BTypeComp](#) typeComp
- [BUInt16](#) dataOffset
- [BUInt16](#) size
- const char * [typeName](#)
- const char * [name](#)

6.64.1 Member Data Documentation

6.64.1.1 [BUInt16](#) BObjMember::dataOffset

6.64.1.2 const char* BObjMember::name

6.64.1.3 [BUInt16](#) BObjMember::size

6.64.1.4 [BType](#) BObjMember::type

6.64.1.5 [BTypeComp](#) BObjMember::typeComp

6.64.1.6 const char* BObjMember::typeName

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

- [BTypes.h](#)

6.65 BPoll Class Reference

This class provides an interface for polling a number of file descriptors. It uses round robin polling.

```
#include <BPoll.h>
```

Public Types

- typedef struct pollfd [PollFd](#)

Public Member Functions

- [BPoll](#) ()
- [~BPoll](#) ()
- void [append](#) (int fd, int events=POLLIN|POLLERR|POLLHUP|POLLNVAL)
Append a file descriptor to polling list.
- void [delFd](#) (int fd)
Remove a file descriptor from polling list.
- [BError doPoll](#) (int &fd, int timeoutUs=-1)
Perform polling operation.
- [BError doPollEvents](#) (int &fd, int &events, int timeoutUs=-1)
Perform polling operation and return events.
- int [getPollFdsNum](#) ()
- [PollFd](#) * [getPollFds](#) ()
- void [clear](#) ()

Private Member Functions

- int [nextFd](#) (int i)

Private Attributes

- int [ofdsNum](#)
The number of FD's in list.
- [PollFd](#) * [ofds](#)
The list of poll fd's.
- int [ofdsNext](#)
The next list entry for round robin polling.

6.65.1 Detailed Description

This class provides an interface for polling a number of file descriptors. It uses round robin polling.

6.65.2 Member Typedef Documentation

6.65.2.1 typedef struct pollfd [BPoll::PollFd](#)

6.65.3 Constructor & Destructor Documentation

6.65.3.1 [BPoll::BPoll](#) ()

6.65.3.2 [BPoll::~~BPoll](#) ()

6.65.4 Member Function Documentation

6.65.4.1 void [BPoll::append](#) (int fd, int events = POLLIN|POLLERR|POLLHUP|POLLNVAL)

Append a file descriptor to polling list.

6.65.4.2 `void BPoll::clear ()`

6.65.4.3 `void BPoll::delFd (int fd)`

Remove a file descriptor from polling list.

6.65.4.4 `BError BPoll::doPoll (int & fd, int timeoutUs = -1)`

Perform polling operation.

6.65.4.5 `BError BPoll::doPollEvents (int & fd, int & events, int timeoutUs = -1)`

Perform polling operation and return events.

6.65.4.6 `BPoll::PollFd * BPoll::getPollFds ()`

6.65.4.7 `int BPoll::getPollFdsNum ()`

6.65.4.8 `int BPoll::nextFd (int i)` [private]

6.65.5 Member Data Documentation

6.65.5.1 `PollFd* BPoll::ofds` [private]

The list of poll fd's.

6.65.5.2 `int BPoll::ofdsNext` [private]

The next list entry for round robin polling.

6.65.5.3 `int BPoll::ofdsNum` [private]

The number of FD's in list.

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

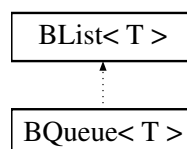
- [BPoll.h](#)
- [BPoll.cpp](#)

6.66 BQueue< T > Class Template Reference

Queue class.

```
#include <BQueue.h>
```

Inheritance diagram for BQueue< T >:



Public Member Functions

- [BQueue](#) ([BUInt](#) size)
- [~BQueue](#) ()
- void [clear](#) ()
Clear the queue.
- [BUInt](#) [writeAvailable](#) () const
- [BError](#) [write](#) (const T &v, [BTimeout](#) timeout=[BTimeoutForever](#))
Append an item onto the queue.
- [BUInt](#) [readAvailable](#) () const
- [BError](#) [read](#) (T &v, [BTimeout](#) timeout=[BTimeoutForever](#))
Get an item from the queue.

Private Attributes

- [BMutex](#) olock
- [BUInt](#) osize
- [BCondInt](#) onumber

Additional Inherited Members

6.66.1 Detailed Description

`template<class T>class BQueue< T >`

Queue class.

6.66.2 Constructor & Destructor Documentation

6.66.2.1 `template<class T > BQueue< T >::BQueue (BUInt size)`

6.66.2.2 `template<class T > BQueue< T >::~~BQueue ()`

6.66.3 Member Function Documentation

6.66.3.1 `template<class T > void BQueue< T >::clear () [virtual]`

Clear the queue.

Reimplemented from [BList< T >](#).

6.66.3.2 `template<class T> BError BQueue< T >::read (T & v, BTimeout timeout = BTimeoutForever)`

Get an item from the queue.

6.66.3.3 `template<class T > BUInt BQueue< T >::readAvailable () const`

6.66.3.4 `template<class T> BError BQueue< T >::write (const T & v, BTimeout timeout = BTimeoutForever)`

Append an item onto the queue.

6.66.3.5 `template<class T> BUInt BQueue<T>::writeAvailable () const`

6.66.4 Member Data Documentation

6.66.4.1 `template<class T> BMutex BQueue<T>::olock [private]`

6.66.4.2 `template<class T> BCondInt BQueue<T>::onumber [private]`

6.66.4.3 `template<class T> BUInt BQueue<T>::osize [private]`

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

- [BQueue.h](#)

6.67 BRefData Class Reference

```
#include <BRefData.h>
```

Public Member Functions

- [BRefData](#) ()
- [BRefData](#) (int [len](#))
- [BRefData](#) (const [BRefData](#) &refData)
- [~BRefData](#) ()
- [BRefData](#) * [copy](#) ()
Create a copy of this reference for writing, if necessary.
- [BRefData](#) * [addRef](#) ()
Increment the reference counter.
- int [deleteRef](#) ()
Decrement the reference counter.
- char * [data](#) ()
Return the raw data pointer.
- int [len](#) ()
Return the length in bytes.
- [BRefData](#) & [operator=](#) (const [BRefData](#) &refData)
- void [setLen](#) (int [len](#))
Set the length in bytes. Note should only be used if [orefCount](#) = 1.

Private Attributes

- [BAtomicCount](#) [orefCount](#)
The reference count, how many users.
- int [olen](#)
The actual length of data in [oData](#).
- void * [odata](#)
Pointer to the data.

6.67.1 Detailed Description

Referenced data storage. This is Thread safe to a degree. The reference counting is protected. However, [setLen\(\)](#) is not and should be protected at a higher level.

6.67.2 Constructor & Destructor Documentation

6.67.2.1 `BRefData::BRefData ()`

6.67.2.2 `BRefData::BRefData (int len)`

6.67.2.3 `BRefData::BRefData (const BRefData & refData)`

6.67.2.4 `BRefData::~~BRefData ()`

6.67.3 Member Function Documentation

6.67.3.1 `BRefData * BRefData::addRef ()`

Increment the reference counter.

6.67.3.2 `BRefData * BRefData::copy ()`

Create a copy of this reference for writing, if necessary.

6.67.3.3 `char* BRefData::data ()` `[inline]`

Return the raw data pointer.

6.67.3.4 `int BRefData::deleteRef ()`

Decrement the reference counter.

6.67.3.5 `int BRefData::len ()` `[inline]`

Return the length in bytes.

6.67.3.6 `BRefData & BRefData::operator= (const BRefData & refData)`

6.67.3.7 `void BRefData::setLen (int len)`

Set the length in bytes. Note should only be used if `orefCount = 1`.

6.67.4 Member Data Documentation

6.67.4.1 `void* BRefData::odata` `[private]`

Pointer to the data.

6.67.4.2 `int BRefData::olen` `[private]`

The actual length of data in `oData`.

6.67.4.3 BAtomicCount BRefData::orefCount [private]

The reference count, how many users.

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

- [BRefData.h](#)
- [BRefData.cpp](#)

6.68 BRtc Class Reference

Realtime clock.

```
#include <BRtc.h>
```

Public Member Functions

- [BRtc](#) ()
- [~BRtc](#) ()
- [BError init](#) (int rate)
Setup interrupt rate.
- void [wait](#) (int delayUs)
Wait specified uS.

Private Attributes

- int [ofd](#)
- int [orate](#)

6.68.1 Detailed Description

Realtime clock.

6.68.2 Constructor & Destructor Documentation

6.68.2.1 BRtc::BRtc ()

6.68.2.2 BRtc::~~BRtc ()

6.68.3 Member Function Documentation

6.68.3.1 BError BRtc::init (int rate)

Setup interrupt rate.

6.68.3.2 void BRtc::wait (int delayUs)

Wait specified uS.

6.68.4 Member Data Documentation

6.68.4.1 `int BRtc::ofd` `[private]`

6.68.4.2 `int BRtc::orate` `[private]`

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

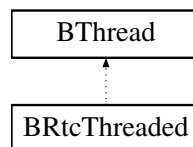
- [BRtc.h](#)
- [BRtc.cpp](#)

6.69 BRtcThreaded Class Reference

Threaded real time clock.

```
#include <BRtc.h>
```

Inheritance diagram for BRtcThreaded:



Public Member Functions

- [BRtcThreaded](#) ()
- [~BRtcThreaded](#) ()
- [BError init](#) (int rate)
Setup interrupt rate.
- void [wait](#) (int delayUs)
Wait specified uS.

Private Member Functions

- void * [function](#) ()

Private Attributes

- [BRtc orte](#)
- int [orate](#)
- [BCond ocond](#)

6.69.1 Detailed Description

Threaded real time clock.

6.69.2 Constructor & Destructor Documentation

6.69.2.1 `BRtcThreaded::BRtcThreaded ()`

6.69.2.2 `BRtcThreaded::~~BRtcThreaded ()`

6.69.3 Member Function Documentation

6.69.3.1 `void * BRtcThreaded::function ()` `[private]`, `[virtual]`

Reimplemented from [BThread](#).

6.69.3.2 `BError BRtcThreaded::init (int rate)`

Setup interrupt rate.

6.69.3.3 `void BRtcThreaded::wait (int delayUs)`

Wait specified uS.

6.69.4 Member Data Documentation

6.69.4.1 `BCond BRtcThreaded::ocond` `[private]`

6.69.4.2 `int BRtcThreaded::orate` `[private]`

6.69.4.3 `BRtc BRtcThreaded::ortc` `[private]`

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

- [BRtc.h](#)
- [BRtc.cpp](#)

6.70 BRWLock Class Reference

thread read-write locks

```
#include <BRWLock.h>
```

Public Member Functions

- [BRWLock](#) ()
- [BRWLock](#) (const [BRWLock](#) &rwlock)
- [~BRWLock](#) ()
- int [rdLock](#) ()
Set lock, wait if necessary.
- int [tryRdLock](#) ()
Test the lock.
- int [wrLock](#) ()
Set lock, wait if necessary.
- int [tryWrLock](#) ()
Test the lock.

- `int unlock ()`
Unlock the lock.
- `BRWLock & operator= (const BRWLock &rwlock)`

Private Attributes

- `pthread_rwlock_t olock`

6.70.1 Detailed Description

thread read-write locks

6.70.2 Constructor & Destructor Documentation

6.70.2.1 `BRWLock::BRWLock ()`

6.70.2.2 `BRWLock::BRWLock (const BRWLock & rwlock)`

6.70.2.3 `BRWLock::~BRWLock ()`

6.70.3 Member Function Documentation

6.70.3.1 `BRWLock & BRWLock::operator= (const BRWLock & rwlock)`

6.70.3.2 `int BRWLock::rdLock ()`

Set lock, wait if necessary.

6.70.3.3 `int BRWLock::tryRdLock ()`

Test the lock.

6.70.3.4 `int BRWLock::tryWrLock ()`

Test the lock.

6.70.3.5 `int BRWLock::unlock ()`

Unlock the lock.

6.70.3.6 `int BRWLock::wrLock ()`

Set lock, wait if necessary.

6.70.4 Member Data Documentation

6.70.4.1 `pthread_rwlock_t BRWLock::olock` `[private]`

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

- [BRWLock.h](#)

- [BRWLock.cpp](#)

6.71 BSema Class Reference

Sempahore class.

```
#include <BSema.h>
```

Public Member Functions

- [BSema](#) (int value=0)
- [BSema](#) (const [BSema](#) &sema)
- [~BSema](#) ()
- int [post](#) ()
Post condition.
- int [wait](#) ()
Wait for contition.
- int [timedWait](#) (int timeUs)
Wait for condition with timeout.
- int [tryWait](#) ()
Test for the condition.
- int [getValue](#) () const
- [BSema](#) & [operator=](#) (const [BSema](#) &sema)

Private Attributes

- sem_t [osema](#)

6.71.1 Detailed Description

Sempahore class.

6.71.2 Constructor & Destructor Documentation

6.71.2.1 [BSema::BSema](#) (int *value* = 0)

6.71.2.2 [BSema::BSema](#) (const [BSema](#) & *sema*)

6.71.2.3 [BSema::~~BSema](#) ()

6.71.3 Member Function Documentation

6.71.3.1 int [BSema::getValue](#) () const

6.71.3.2 [BSema](#) & [BSema::operator=](#) (const [BSema](#) & *sema*)

6.71.3.3 int [BSema::post](#) ()

Post condition.

6.71.3.4 `int BSema::timedWait (int timeUs)`

Wait for condition with timeout.

6.71.3.5 `int BSema::tryWait ()`

Test for the condition.

6.71.3.6 `int BSema::wait ()`

Wait for contition.

6.71.4 Member Data Documentation

6.71.4.1 `sem_t BSema::osema` [private]

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

- [BSema.h](#)
- [BSema.cpp](#)

6.72 BSemaphore Class Reference

Semaphore class.

```
#include <BSemaphore.h>
```

Public Member Functions

- [BSemaphore](#) ()
- [BSemaphore](#) (const [BSemaphore](#) &semaphore)
- [~BSemaphore](#) ()
- [Bool](#) wait ([BTimeout](#) timeoutUs=[BTimeoutForever](#))
Wait for the semaphore.
- void [set](#) ()
Set the semaphore.
- int [getValue](#) () const
- [BSemaphore](#) & [operator=](#) (const [BSemaphore](#) &semaphore)

Private Attributes

- `sem_t` [osema](#)

6.72.1 Detailed Description

Semaphore class.

6.72.2 Constructor & Destructor Documentation

6.72.2.1 BSemaphore::BSemaphore ()

6.72.2.2 BSemaphore::BSemaphore (const BSemaphore & semaphore)

6.72.2.3 BSemaphore::~~BSemaphore ()

6.72.3 Member Function Documentation

6.72.3.1 int BSemaphore::getValue () const

6.72.3.2 BSemaphore & BSemaphore::operator= (const BSemaphore & semaphore)

6.72.3.3 void BSemaphore::set ()

Set the semaphore.

6.72.3.4 Bool BSemaphore::wait (BTimeout timeoutUs = BTimeoutForever)

Wait for the semaphore.

6.72.4 Member Data Documentation

6.72.4.1 sem_t BSemaphore::osema [private]

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

- [BSemaphore.h](#)
- [BSemaphore.cpp](#)

6.73 BSemaphoreCount Class Reference

```
#include <BSemaphore.h>
```

Public Member Functions

- [BSemaphoreCount](#) ()
- [BSemaphoreCount](#) (const [BSemaphoreCount](#) &semaphore)
- [~BSemaphoreCount](#) ()
- void [setValue](#) (BUInt v)
- void [add](#) (int v=1)
Set the semaphore.
- Bool [wait](#) (BUInt v=1, BTimeout timeoutUs=BTimeoutForever)
Wait for the semaphore.
- Bool [take](#) (BUInt v=1, BTimeout timeoutUs=BTimeoutForever)
Take for the semaphore.
- BUInt [value](#) ()
- [BSemaphoreCount](#) & [operator=](#) (const [BSemaphoreCount](#) &semaphore)

Private Attributes

- [BMutex olock](#)
- [BSemaphore osema](#)
- volatile [BUInt ovalue](#)

6.73.1 Constructor & Destructor Documentation

6.73.1.1 `BSemaphoreCount::BSemaphoreCount ()`

6.73.1.2 `BSemaphoreCount::BSemaphoreCount (const BSemaphoreCount & semaphore)`

6.73.1.3 `BSemaphoreCount::~~BSemaphoreCount ()`

6.73.2 Member Function Documentation

6.73.2.1 `void BSemaphoreCount::add (int v = 1)`

Set the semaphore.

6.73.2.2 `BSemaphoreCount & BSemaphoreCount::operator= (const BSemaphoreCount & semaphore)`

6.73.2.3 `void BSemaphoreCount::setValue (BUInt v)`

6.73.2.4 `Bool BSemaphoreCount::take (BUInt v = 1, BTimeout timeoutUs = BTimeoutForever)`

Take for the semaphore.

6.73.2.5 `BUInt BSemaphoreCount::value ()`

6.73.2.6 `Bool BSemaphoreCount::wait (BUInt v = 1, BTimeout timeoutUs = BTimeoutForever)`

Wait for the semaphore.

6.73.3 Member Data Documentation

6.73.3.1 `BMutex BSemaphoreCount::olock` [private]

6.73.3.2 `BSemaphore BSemaphoreCount::osema` [private]

6.73.3.3 `volatile BUInt BSemaphoreCount::ovalue` [private]

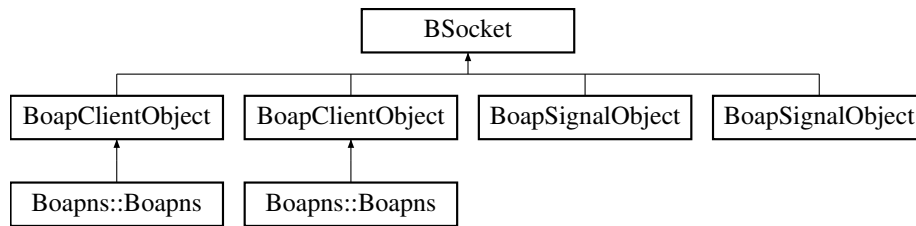
The documentation for this class was generated from the following files:

- [BSemaphore.h](#)
- [BSemaphore.cpp](#)

6.74 BSocket Class Reference

```
#include <BSocket.h>
```

Inheritance diagram for BSocket:



Public Types

- enum [NType](#) { [STREAM](#), [DGRAM](#) }
- enum [Priority](#) { [PriorityLow](#), [PriorityNormal](#), [PriorityHigh](#) }

Public Member Functions

- [BSocket](#) ()
- [BSocket](#) (int fd)
- [BSocket](#) (NType type)
- [BSocket](#) (int domain, int type, int protocol)
- [~BSocket](#) ()
- [BError init](#) (int domain, int type, int protocol)
- [BError init](#) (NType type)
- void [setFd](#) (int fd)
- int [getFd](#) ()
- [BError bind](#) (const [BSocketAddress](#) &add)
- [BError connect](#) (const [BSocketAddress](#) &add)
- [BError shutdown](#) (int how)
- [BError close](#) ()
- [BError listen](#) (int backlog=5)
- [BError accept](#) (int &fd)
- [BError accept](#) (int &fd, [BSocketAddress](#) &address)
- [BError send](#) (const void *buf, [BSize](#) nbytes, [BSize](#) &nbytesSent, int flags=0)
- [BError sendTo](#) (const [BSocketAddress](#) &address, const void *buf, [BSize](#) nbytes, [BSize](#) &nbytesSent, int flags=0)
- [BError recv](#) (void *buf, [BSize](#) maxbytes, [BSize](#) &nbytesRecv, int flags=0)
- [BError recvFrom](#) ([BSocketAddress](#) &address, void *buf, [BSize](#) maxbytes, [BSize](#) &nbytesRecv, int flags=0)
- [BError recvWithTimeout](#) (void *buf, [BSize](#) maxbytes, [BSize](#) &nbytesRecv, int timeout, int flags=0)
- [BError recvFromWithTimeout](#) ([BSocketAddress](#) &address, void *buf, [BSize](#) maxbytes, [BSize](#) &nbytesRecv, int timeout, int flags=0)
- [BError setSockOpt](#) (int level, int optname, void *optval, unsigned int optlen)
- [BError getSockOpt](#) (int level, int optname, void *optval, unsigned int *optlen)
- [BError setReuseAddress](#) (int on)
- [BError setBroadCast](#) (int on)
- [BError setPriority](#) ([Priority](#) priority)
- [BError getMTU](#) (uint32_t &mtu)
- [BError getAddress](#) ([BSocketAddress](#) &address)

Private Attributes

- int [osocket](#)

6.74.1 Member Enumeration Documentation

6.74.1.1 enum BSocket::NType

Enumerator

STREAM

DGRAM

6.74.1.2 enum BSocket::Priority

Enumerator

PriorityLow

PriorityNormal

PriorityHigh

6.74.2 Constructor & Destructor Documentation

6.74.2.1 BSocket::BSocket ()

6.74.2.2 BSocket::BSocket (int *fd*)

6.74.2.3 BSocket::BSocket (NType *type*)

6.74.2.4 BSocket::BSocket (int *domain*, int *type*, int *protocol*)

6.74.2.5 BSocket::~~BSocket ()

6.74.3 Member Function Documentation

6.74.3.1 BError BSocket::accept (int & *fd*)

6.74.3.2 BError BSocket::accept (int & *fd*, BSocketAddress & *address*)

6.74.3.3 BError BSocket::bind (const BSocketAddress & *add*)

6.74.3.4 BError BSocket::close ()

6.74.3.5 BError BSocket::connect (const BSocketAddress & *add*)

6.74.3.6 BError BSocket::getAddress (BSocketAddress & *address*)

6.74.3.7 int BSocket::getFd ()

6.74.3.8 BError BSocket::getMTU (uint32_t & *mtu*)

6.74.3.9 BError BSocket::getSockOpt (int *level*, int *optname*, void * *optval*, unsigned int * *optlen*)

6.74.3.10 BError BSocket::init (int *domain*, int *type*, int *protocol*)

6.74.3.11 BError BSocket::init (NType *type*)

6.74.3.12 BError BSocket::listen (int *backlog* = 5)

- 6.74.3.13 **BError** BSocket::recv (void * *buf*, BSize *maxbytes*, BSize & *nbytesRecv*, int *flags* = 0)
- 6.74.3.14 **BError** BSocket::recvFrom (BSocketAddress & *address*, void * *buf*, BSize *maxbytes*, BSize & *nbytesRecv*, int *flags* = 0)
- 6.74.3.15 **BError** BSocket::recvFromWithTimeout (BSocketAddress & *address*, void * *buf*, BSize *maxbytes*, BSize & *nbytesRecv*, int *timeout*, int *flags* = 0)
- 6.74.3.16 **BError** BSocket::recvWithTimeout (void * *buf*, BSize *maxbytes*, BSize & *nbytesRecv*, int *timeout*, int *flags* = 0)
- 6.74.3.17 **BError** BSocket::send (const void * *buf*, BSize *nbytes*, BSize & *nbytesSent*, int *flags* = 0)
- 6.74.3.18 **BError** BSocket::sendTo (const BSocketAddress & *address*, const void * *buf*, BSize *nbytes*, BSize & *nbytesSent*, int *flags* = 0)
- 6.74.3.19 **BError** BSocket::setBroadCast (int *on*)
- 6.74.3.20 void BSocket::setFd (int *fd*)
- 6.74.3.21 **BError** BSocket::setPriority (Priority *priority*)
- 6.74.3.22 **BError** BSocket::setReuseAddress (int *on*)
- 6.74.3.23 **BError** BSocket::setSockOpt (int *level*, int *optname*, void * *optval*, unsigned int *optlen*)
- 6.74.3.24 **BError** BSocket::shutdown (int *how*)

6.74.4 Member Data Documentation

- 6.74.4.1 int BSocket::osocket [private]

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

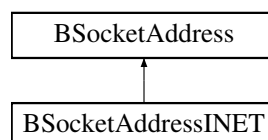
- [BSocket.h](#)
- [BSocket.cpp](#)

6.75 BSocketAddress Class Reference

Socket Address.

```
#include <BSocket.h>
```

Inheritance diagram for BSocketAddress:



Public Types

- typedef struct sockaddr [SockAddr](#)

Public Member Functions

- [BSocketAddress](#) ()
- [BSocketAddress](#) (const [BSocketAddress](#) &add)
- [BSocketAddress](#) ([SockAddr](#) *address, int [len](#))
- [~BSocketAddress](#) ()
- [BError](#) set ([SockAddr](#) *address, int [len](#))
- const [SockAddr](#) * raw () const
- int [len](#) () const
- [BSocketAddress](#) & operator= (const [BSocketAddress](#) &add)
- operator const [SockAddr](#) * () const
- int operator== (const [BSocketAddress](#) &add) const
- int operator!= (const [BSocketAddress](#) &add) const

Private Attributes

- int [olen](#)
- [SockAddr](#) * [oaddress](#)

6.75.1 Detailed Description

Socket Address.

6.75.2 Member Typedef Documentation

6.75.2.1 typedef struct sockaddr [BSocketAddress::SockAddr](#)

6.75.3 Constructor & Destructor Documentation

6.75.3.1 [BSocketAddress::BSocketAddress](#) ()

6.75.3.2 [BSocketAddress::BSocketAddress](#) (const [BSocketAddress](#) & add)

6.75.3.3 [BSocketAddress::BSocketAddress](#) ([SockAddr](#) * address, int [len](#))

6.75.3.4 [BSocketAddress::~BSocketAddress](#) ()

6.75.4 Member Function Documentation

6.75.4.1 int [BSocketAddress::len](#) () const

6.75.4.2 [BSocketAddress::operator const SockAddr *](#) () const `[inline]`

6.75.4.3 int [BSocketAddress::operator!=](#) (const [BSocketAddress](#) & add) const

6.75.4.4 [BSocketAddress](#) & [BSocketAddress::operator=](#) (const [BSocketAddress](#) & add)

6.75.4.5 int [BSocketAddress::operator==](#) (const [BSocketAddress](#) & add) const

6.75.4.6 const [BSocketAddress::SockAddr](#) * [BSocketAddress::raw](#) () const

6.75.4.7 [BError](#) [BSocketAddress::set](#) ([SockAddr](#) * address, int [len](#))

6.75.5 Member Data Documentation

6.75.5.1 `SockAddr* BSocketAddress::oaddress` [private]

6.75.5.2 `int BSocketAddress::olen` [private]

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

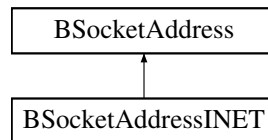
- [BSocket.h](#)
- [BSocket.cpp](#)

6.76 BSocketAddressINET Class Reference

IP aware socket address.

```
#include <BSocket.h>
```

Inheritance diagram for BSocketAddressINET:



Public Types

- typedef struct sockaddr_in [SockAddrIP](#)

Public Member Functions

- [BError set](#) ([BString](#) hostName, uint32_t port)
- [BError set](#) (uint32_t address, uint32_t port)
- [BError set](#) ([BString](#) hostName, [BString](#) service, [BString](#) type)
- void [setPort](#) (uint32_t port)
- uint32_t [address](#) ()
Returns socket ip address.
- uint32_t [port](#) ()
Returns socket port.
- [BString getString](#) ()
Return string version of address <ip>:<port>

Static Public Member Functions

- static [BString getHostName](#) ()
Get this hosts network name.
- static [BList](#)< uint32_t > [getIpAddresses](#) ()
Get a list of all the IP addresses of this host.
- static [BList](#)< [BString](#) > [getIpAddressList](#) ()
Get a list of all the IP addresses of this host under hostname.
- static [BList](#)< [BString](#) > [getIpAddressListAll](#) ()
Get a list of all the IP addresses of this host looking at physical interfaces.

6.76.1 Detailed Description

IP aware socket address.

6.76.2 Member Typedef Documentation

6.76.2.1 `typedef struct sockaddr_in BSocketAddressINET::SockAddrIP`

6.76.3 Member Function Documentation

6.76.3.1 `uint32_t BSocketAddressINET::address ()`

Returns socket ip address.

6.76.3.2 `BString BSocketAddressINET::getHostName () [static]`

Get this hosts network name.

6.76.3.3 `BList< uint32_t > BSocketAddressINET::getIpAddresses () [static]`

Get a list of all the IP addresses of this host.

6.76.3.4 `BList< BString > BSocketAddressINET::getIpAddressList () [static]`

Get a list of all the IP addresses of this host under hostname.

6.76.3.5 `BList< BString > BSocketAddressINET::getIpAddressListAll () [static]`

Get a list of all the IP addresses of this host looking at physical interfaces.

6.76.3.6 `BString BSocketAddressINET::getString ()`

Return string version of address <ip>:<port>

6.76.3.7 `uint32_t BSocketAddressINET::port ()`

Returns socket port.

6.76.3.8 `BError BSocketAddressINET::set (BString hostName, uint32_t port)`

6.76.3.9 `BError BSocketAddressINET::set (uint32_t address, uint32_t port)`

6.76.3.10 `BError BSocketAddressINET::set (BString hostName, BString service, BString type)`

6.76.3.11 `void BSocketAddressINET::setPort (uint32_t port)`

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

- [BSocket.h](#)
- [BSocket.cpp](#)

6.77 BSpI Class Reference

BSpI class.

```
#include <BSpI.h>
```

Public Types

- enum [Mode](#) { [Mode0](#) = 0, [Mode1](#) = 1, [Mode2](#) = 2, [Mode3](#) = 3 }

Public Member Functions

- [BSpI](#) ()
- [BError](#) [init](#) ([BString](#) devName, [BUInt](#) speed=1000000, [Mode](#) mode=[Mode1](#), [Bool](#) csActive=0)
- [BError](#) [transact](#) ([BUInt8](#) dev, void *txBuf, int txLen, int pad, void *rxBuf, int rxLen)

Private Attributes

- [BString](#) odevName
- int odev

6.77.1 Detailed Description

[BSpI](#) class.

6.77.2 Member Enumeration Documentation

6.77.2.1 enum BSpI::Mode

Enumerator

Mode0

Mode1

Mode2

Mode3

6.77.3 Constructor & Destructor Documentation

6.77.3.1 BSpI::BSpI ()

6.77.4 Member Function Documentation

6.77.4.1 BError BSpI::init (BString devName, BUInt speed = 1000000, Mode mode = Mode1, Bool csActive = 0)

6.77.4.2 BError BSpI::transact (BUInt8 dev, void * txBuf, int txLen, int pad, void * rxBuf, int rxLen)

6.77.5 Member Data Documentation

6.77.5.1 int BSpI::odev [private]

6.77.5.2 BString BSpI::odevName [private]

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

- [BSpi.h](#)
- [BSpi.cpp](#)

6.78 BString Class Reference

```
#include <BString.h>
```

Public Member Functions

- [BString](#) ()
- [BString](#) (const [BString](#) &string)
- [BString](#) (const char *str)
- [BString](#) (const char *str, unsigned int len)
- [BString](#) (char ch)
- [BString](#) (BInt v)
- [BString](#) (BUInt v)
- [BString](#) (BUInt64 v)
- [BString](#) (double v)
- [~BString](#) ()
- [BString copy](#) () const
Return an independant copy.
- int [len](#) () const
Length of string.
- const char * [retStr](#) () const
Ptr to char representation.*
- char * [retStrDup](#) () const
Ptr to newly malloc'd char.*
- int [retInt](#) () const
Return string as a int.
- unsigned int [retUInt](#) () const
Return string as a int.
- double [retDouble](#) () const
Return string as a double.
- int [compare](#) (const [BString](#) &string) const
Compare strings.
- int [compareWild](#) (const [BString](#) &string) const
Compare string to string with wildcards.
- int [compareWildExpression](#) (const [BString](#) &string) const
Compare string to space delimited patterns.
- int [compareRegex](#) (const [BString](#) &pattern, int ignoreCase=0) const
Compare strings.
- [BString](#) & [truncate](#) (int len)
Truncate to length len.
- [BString](#) & [pad](#) (int len)
Pad to length len.
- void [clear](#) ()
Clear the string.
- [BString](#) & [toUpper](#) ()
Convert to uppercase.
- [BString](#) & [toLower](#) ()

- Convert to lowercase.*

 - [BString lowerFirst](#) ()

Return string with lowercase first character.
- void [removeNL](#) ()

Remove if present NL from last char.
- [BString justify](#) (int leftMargin, int width)

Justify the string to the given width.
- [BString fixedLen](#) (int length, int rightJustify=0)

return string formatted to fixed length
- [BString firstLine](#) ()

Return first line.
- [BString translateChar](#) (char ch, [BString](#) replace=" ")

Translate character converting them to the given string.
- [BString reverse](#) () const

Reverse character order.
- [BString subString](#) (int start, int len) const

Returns substring.
- int [del](#) (int start, int len)

Delete substring.
- int [insert](#) (int start, [BString](#) str)

Insert substring.
- int [append](#) (const [BString](#) &str)

Append a string.
- [BString add](#) (const [BString](#) &str) const

Add strings returning result.
- [BString](#) & [printf](#) (const char *fmt,...)

Formatted print into the string.
- int [find](#) (char ch) const

Find ch in string searching forwards.
- int [find](#) ([BString](#) str) const

Find string in string searching forwards.
- int [findReverse](#) (char ch) const

Find ch in string searching backwards.
- [BString csvEncode](#) () const

Encode a string for CSV.
- [BString](#) & [csvDecode](#) (const [BString](#) str)

Decode a string from CSV.
- [BString base64Encode](#) () const

Encode a string to base64.
- [BError](#) [base64Decode](#) ([BString](#) &str) const

Decode a string from base64.
- [BList](#)< [BString](#) > [getTokenList](#) ([BString](#) separators)

Break string into tokens.
- [BList](#)< [BString](#) > [getTokenList](#) (char separator)

Break string into tokens.
- [BString removeSeparators](#) ([BString](#) separators)

Remove any char from sepatators from string.
- [BString pullToken](#) ([BString](#) terminators)

Pull token from start of string.
- [BString pullSeparators](#) ([BString](#) separators)

Pull separators from start of string.

- [BString pullWord \(\)](#)
Pull a word out of the head of the string.
- [BString pullLine \(\)](#)
Pull a line out of the head of the string.
- [BList< BString > split \(char splitChar\)](#)
Split string into an array based on the character separator.
- [BString dirname \(\)](#)
- [BString basename \(\)](#)
- [BString extension \(\)](#)
- [BUInt32 hash \(\)](#) const
- [char & get \(int pos\)](#)
- [const char & get \(int pos\)](#) const
- [BString & operator= \(const BString &string\)](#)
- [char & operator\[\] \(int pos\)](#)
- [int operator== \(const BString &s\)](#) const
- [int operator== \(const char *s\)](#) const
- [int operator> \(const BString &s\)](#) const
- [int operator> \(const char *s\)](#) const
- [int operator< \(const BString &s\)](#) const
- [int operator< \(const char *s\)](#) const
- [int operator>= \(const BString &s\)](#) const
- [int operator<= \(const BString &s\)](#) const
- [int operator!= \(const BString &s\)](#) const
- [int operator!= \(const char *s\)](#) const
- [BString operator+ \(const BString &s\)](#) const
- [BString operator+ \(const char *s\)](#) const
- [BString operator+= \(const BString &s\)](#)
- [BString operator+= \(const char *s\)](#)
- [BString operator+ \(char ch\)](#) const
- [BString operator+ \(BInt i\)](#) const
- [BString operator+ \(BUInt i\)](#) const
- [BString operator+ \(BUInt64 i\)](#) const
- [operator const char * \(\)](#) const
- [BString field \(int field\)](#) const
- [char ** fields \(\)](#)

Static Public Member Functions

- static [BString convert \(char ch\)](#)
Converts char to string.
- static [BString convert \(BInt value\)](#)
Converts int to string.
- static [BString convert \(BUInt value\)](#)
Converts uint to string.
- static [BString convert \(double value, int eFormat=0\)](#)
Converts double to string.
- static [BString convert \(BUInt64 value\)](#)
Converts long long to string.
- static [BString convertHex \(BInt value\)](#)
Converts int to string as hex value.
- static [BString convertHex \(BUInt value\)](#)
Converts uint to string as hex value.

Protected Attributes

- [BRefData](#) * *ostr*

Private Member Functions

- void [init](#) (const char **str*)
- int [inString](#) (int *pos*) const
- int [isSpace](#) (char *ch*) const

6.78.1 Constructor & Destructor Documentation

6.78.1.1 [BString::BString](#) ()

6.78.1.2 [BString::BString](#) (const [BString](#) & *string*)

6.78.1.3 [BString::BString](#) (const char * *str*)

6.78.1.4 [BString::BString](#) (const char * *str*, unsigned int *len*)

6.78.1.5 [BString::BString](#) (char *ch*)

6.78.1.6 [BString::BString](#) ([BInt](#) *v*)

6.78.1.7 [BString::BString](#) ([BUInt](#) *v*)

6.78.1.8 [BString::BString](#) ([BUInt64](#) *v*)

6.78.1.9 [BString::BString](#) (double *v*)

6.78.1.10 [BString::~~BString](#) ()

6.78.2 Member Function Documentation

6.78.2.1 [BString](#) [BString::add](#) (const [BString](#) & *str*) const

Add strings returning result.

6.78.2.2 [int](#) [BString::append](#) (const [BString](#) & *str*)

Append a string.

6.78.2.3 [BError](#) [BString::base64Decode](#) ([BString](#) & *str*) const

Decode a string from base64.

6.78.2.4 [BString](#) [BString::base64Encode](#) () const

Encode a string to base64.

6.78.2.5 `BString BString::basename ()`

6.78.2.6 `void BString::clear ()`

Clear the string.

6.78.2.7 `int BString::compare (const BString & string) const`

Compare strings.

6.78.2.8 `int BString::compareRegex (const BString & pattern, int ignoreCase = 0) const`

Compare strings.

6.78.2.9 `int BString::compareWild (const BString & string) const`

Compare string to string with wildcards.

6.78.2.10 `int BString::compareWildExpression (const BString & string) const`

Compare string to space delimited patterns.

6.78.2.11 `BString BString::convert (char ch) [static]`

Converts char to string.

6.78.2.12 `BString BString::convert (BInt value) [static]`

Converts int to string.

6.78.2.13 `BString BString::convert (BUInt value) [static]`

Converts uint to string.

6.78.2.14 `BString BString::convert (double value, int eFormat = 0) [static]`

Converts double to string.

6.78.2.15 `BString BString::convert (BUInt64 value) [static]`

Converts long long to string.

6.78.2.16 `BString BString::convertHex (BInt value) [static]`

Converts int to string as hex value.

6.78.2.17 `BString BString::convertHex (BUInt value) [static]`

Converts uint to string as hex value.

6.78.2.18 `BString BString::copy () const`

Return an independant copy.

6.78.2.19 `BString & BString::csvDecode (const BString str)`

Decode a string from CSV.

6.78.2.20 `BString BString::csvEncode () const`

Encode a string for CSV.

6.78.2.21 `int BString::del (int start, int len)`

Delete substring.

6.78.2.22 `BString BString::dirname ()`

6.78.2.23 `BString BString::extension ()`

6.78.2.24 `BString BString::field (int field) const`

6.78.2.25 `char ** BString::fields ()`

6.78.2.26 `int BString::find (char ch) const`

Find *ch* in string searching forwards.

6.78.2.27 `int BString::find (BString str) const`

Find string in string searching forwards.

6.78.2.28 `int BString::findReverse (char ch) const`

Find *ch* in string searching backwards.

6.78.2.29 `BString BString::firstLine ()`

Return first line.

6.78.2.30 `BString BString::fixedLen (int length, int rightJustify = 0)`

return string formatted to fixed length

6.78.2.31 `char & BString::get (int pos)`

6.78.2.32 `const char & BString::get (int pos) const`

6.78.2.33 `BList< BString > BString::getTokenList (BString separators)`

Break string into tokens.

6.78.2.34 **BList< BString > BString::getTokenList (char *separator*)**

Break string into tokens.

6.78.2.35 **BUInt32 BString::hash () const**

6.78.2.36 **void BString::init (const char * *str*)** [private]

6.78.2.37 **int BString::insert (int *start*, BString *str*)**

Insert substring.

6.78.2.38 **int BString::inString (int *pos*) const** [private]

6.78.2.39 **int BString::isSpace (char *ch*) const** [private]

6.78.2.40 **BString BString::justify (int *leftMargin*, int *width*)**

Justify the string to the given width.

6.78.2.41 **int BString::len () const**

Length of string.

6.78.2.42 **BString BString::lowerFirst ()**

Return string with lowercase first character.

6.78.2.43 **BString::operator const char * () const** [inline]

6.78.2.44 **int BString::operator!= (const BString & *s*) const** [inline]

6.78.2.45 **int BString::operator!= (const char * *s*) const** [inline]

6.78.2.46 **BString BString::operator+ (const BString & *s*) const** [inline]

6.78.2.47 **BString BString::operator+ (const char * *s*) const** [inline]

6.78.2.48 **BString BString::operator+ (char *ch*) const** [inline]

6.78.2.49 **BString BString::operator+ (BInt *i*) const** [inline]

6.78.2.50 **BString BString::operator+ (BUInt *i*) const** [inline]

6.78.2.51 **BString BString::operator+ (BUInt64 *i*) const** [inline]

6.78.2.52 **BString BString::operator+= (const BString & *s*)** [inline]

6.78.2.53 **BString BString::operator+= (const char * *s*)** [inline]

6.78.2.54 **int BString::operator< (const BString & *s*) const** [inline]

6.78.2.55 **int BString::operator< (const char * *s*) const** [inline]

6.78.2.56 `int BString::operator<= (const BString & s) const` `[inline]`

6.78.2.57 `BString & BString::operator= (const BString & string)`

6.78.2.58 `int BString::operator== (const BString & s) const` `[inline]`

6.78.2.59 `int BString::operator== (const char * s) const` `[inline]`

6.78.2.60 `int BString::operator> (const BString & s) const` `[inline]`

6.78.2.61 `int BString::operator> (const char * s) const` `[inline]`

6.78.2.62 `int BString::operator>= (const BString & s) const` `[inline]`

6.78.2.63 `char & BString::operator[] (int pos)`

6.78.2.64 `BString & BString::pad (int len)`

Pad to length len.

6.78.2.65 `BString & BString::printf (const char * fmt, ...)`

Formatted print into the string.

6.78.2.66 `BString BString::pullLine ()`

Pull a line out of the head of the string.

6.78.2.67 `BString BString::pullSeparators (BString separators)`

Pull separators from start of string.

6.78.2.68 `BString BString::pullToken (BString terminators)`

Pull token from start of string.

6.78.2.69 `BString BString::pullWord ()`

Pull a word out of the head of the string.

6.78.2.70 `void BString::removeNL ()`

Remove if present NL from last char.

6.78.2.71 `BString BString::removeSeparators (BString separators)`

Remove any char from separators from string.

6.78.2.72 `double BString::retDouble () const`

Return string as a double.

6.78.2.73 `int BString::retInt () const`

Return string as a int.

6.78.2.74 `const char * BString::retStr () const`

Ptr to char* representation.

6.78.2.75 `char * BString::retStrDup () const`

Ptr to newly malloc'd char*.

6.78.2.76 `unsigned int BString::retUInt () const`

Return string as a int.

6.78.2.77 `BString BString::reverse () const`

Reverse character order.

6.78.2.78 `BList< BString > BString::split (char splitChar)`

Split string into an array based on the character separator.

6.78.2.79 `BString BString::subString (int start, int len) const`

Returns substring.

6.78.2.80 `BString & BString::toLower ()`

Convert to lowercase.

6.78.2.81 `BString & BString::toUpper ()`

Convert to uppercase.

6.78.2.82 `BString BString::translateChar (char ch, BString replace = " ")`

Translate character converting them to the given string.

6.78.2.83 `BString & BString::truncate (int len)`

Truncate to length len.

6.78.3 **Member Data Documentation****6.78.3.1** `BRefData* BString::ostr` [protected]

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

- [BString.h](#)
- [BString.cpp](#)

6.79 BStringLocked Class Reference

```
#include <BStringLocked.h>
```

Public Member Functions

- [BStringLocked](#) ()
- [BStringLocked](#) (const [BStringLocked](#) &s)
- [BStringLocked](#) (const [BString](#) &s)
- int [len](#) () const
Length of string.
- [operator BString](#) () const
- [BStringLocked operator+](#) (const [BStringLocked](#) &s) const
- [BStringLocked](#) & [operator=](#) (const [BStringLocked](#) &s)

Private Attributes

- [BStringMutex olock](#)
- [BString ostr](#)

6.79.1 Constructor & Destructor Documentation

6.79.1.1 [BStringLocked::BStringLocked](#) () [\[inline\]](#)

6.79.1.2 [BStringLocked::BStringLocked](#) (const [BStringLocked](#) & s) [\[inline\]](#)

6.79.1.3 [BStringLocked::BStringLocked](#) (const [BString](#) & s) [\[inline\]](#)

6.79.2 Member Function Documentation

6.79.2.1 int [BStringLocked::len](#) () const [\[inline\]](#)

Length of string.

6.79.2.2 [BStringLocked::operator BString](#) () const [\[inline\]](#)

6.79.2.3 [BStringLocked BStringLocked::operator+](#) (const [BStringLocked](#) & s) const [\[inline\]](#)

6.79.2.4 [BStringLocked& BStringLocked::operator=](#) (const [BStringLocked](#) & s) [\[inline\]](#)

6.79.3 Member Data Documentation

6.79.3.1 [BStringMutex BStringLocked::olock](#) [\[mutable\]](#), [\[private\]](#)

6.79.3.2 [BString BStringLocked::ostr](#) [\[private\]](#)

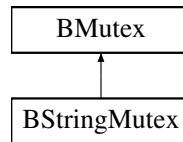
The documentation for this class was generated from the following file:

- [BStringLocked.h](#)

6.80 BStringMutex Class Reference

```
#include <BStringLocked.h>
```

Inheritance diagram for BStringMutex:



Public Member Functions

- [BStringMutex \(\)](#)

Additional Inherited Members

6.80.1 Constructor & Destructor Documentation

6.80.1.1 BStringMutex::BStringMutex () [inline]

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

- [BStringLocked.h](#)

6.81 BTable Class Reference

```
#include <BTable.h>
```

Public Member Functions

- [BTable \(\)](#)
- [~BTable \(\)](#)
- void [clear \(\)](#)
- void [setTitle](#) ([BArray](#)< [BString](#) > title)
- void [addRow](#) ([BArray](#)< [BString](#) > data)
- void [print \(\)](#)

Private Member Functions

- void [calculateWidths \(\)](#)
- void [printLine](#) ([BArray](#)< [BString](#) > line, int comment=0)

Private Attributes

- [BArray](#)< [BString](#) > otitle
- [BList](#)< [BArray](#)< [BString](#) > > odata
- [BArray](#)< int > ocolumnWidths

6.81.1 Constructor & Destructor Documentation

6.81.1.1 `BTable::BTable ()`

6.81.1.2 `BTable::~~BTable ()`

6.81.2 Member Function Documentation

6.81.2.1 `void BTable::addRow (BArray< BString > data)`

6.81.2.2 `void BTable::calculateWidths ()` [private]

6.81.2.3 `void BTable::clear ()`

6.81.2.4 `void BTable::print ()`

6.81.2.5 `void BTable::printLine (BArray< BString > line, int comment = 0)` [private]

6.81.2.6 `void BTable::setTitle (BArray< BString > title)`

6.81.3 Member Data Documentation

6.81.3.1 `BArray<int> BTable::ocolumnWidths` [private]

6.81.3.2 `BList<BArray<BString>> BTable::odata` [private]

6.81.3.3 `BArray<BString> BTable::otitle` [private]

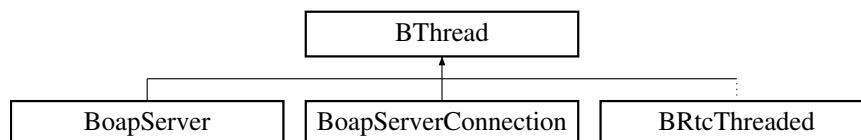
The documentation for this class was generated from the following files:

- [BTable.h](#)
- [BTable.cpp](#)

6.82 BThread Class Reference

```
#include <BThread.h>
```

Inheritance diagram for BThread:



Public Member Functions

- [BThread \(\)](#)
- virtual [~BThread \(\)](#)
- int [setInitPriority](#) (int policy, int priority)
- int [setInitStackSize](#) (size_t stackSize)
- int [start \(\)](#)
- void * [result \(\)](#)
- int [running \(\)](#)

- int [setPriority](#) (int policy, int priority)
- int [cancel](#) ()
- void * [waitForCompletion](#) ()
- pthread_t [getThread](#) ()
- virtual void * [function](#) ()

Static Private Member Functions

- static void * [startFunc](#) (void *)

Private Attributes

- pthread_t [othread](#)
- size_t [ostackSize](#)
- int [opolicy](#)
- int [opriority](#)
- int [orunning](#)
- void * [oreult](#)

6.82.1 Constructor & Destructor Documentation

6.82.1.1 [BThread::BThread \(\)](#)

6.82.1.2 [BThread::~~BThread \(\)](#) [virtual]

6.82.2 Member Function Documentation

6.82.2.1 [int BThread::cancel \(\)](#)

6.82.2.2 [void * BThread::function \(\)](#) [virtual]

Reimplemented in [BoapServer](#), [BoapServerConnection](#), and [BRtcThreaded](#).

6.82.2.3 [pthread_t BThread::getThread \(\)](#)

6.82.2.4 [void * BThread::result \(\)](#)

6.82.2.5 [int BThread::running \(\)](#)

6.82.2.6 [int BThread::setInitPriority \(int policy, int priority \)](#)

6.82.2.7 [int BThread::setInitStackSize \(size_t stackSize \)](#)

6.82.2.8 [int BThread::setPriority \(int policy, int priority \)](#)

6.82.2.9 [int BThread::start \(\)](#)

6.82.2.10 [void * BThread::startFunc \(void * arg \)](#) [static], [private]

6.82.2.11 [void * BThread::waitForCompletion \(\)](#)

6.82.3 Member Data Documentation

- 6.82.3.1 `int BThread::opolicy` `[private]`
- 6.82.3.2 `int BThread::opriority` `[private]`
- 6.82.3.3 `void* BThread::oresult` `[private]`
- 6.82.3.4 `int BThread::orunning` `[private]`
- 6.82.3.5 `size_t BThread::ostackSize` `[private]`
- 6.82.3.6 `pthread_t BThread::othread` `[private]`

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

- [BThread.h](#)
- [BThread.cpp](#)

6.83 BTime Class Reference

```
#include <BTime.h>
```

Public Member Functions

- [BTime](#) ([BUInt32](#) t=0)
- void [set](#) ([BUInt32](#) seconds)
Set the date and time.
- void [set](#) ([BUInt](#) year, [BUInt](#) month, [BUInt](#) day, [BUInt](#) hour=0, [BUInt](#) minute=0, [BUInt](#) second=0)
Set the date and time.
- void [setYearDay](#) ([BUInt](#) year, [BUInt](#) yearDay, [BUInt](#) hour=0, [BUInt](#) minute=0, [BUInt](#) second=0)
Set the date and time.
- void [getDate](#) ([BUInt](#) &year, [BUInt](#) &month, [BUInt](#) &day) const
Return the date information.
- void [getTime](#) ([BUInt](#) &hour, [BUInt](#) &minute, [BUInt](#) &second) const
Return the time information.
- [BUInt32](#) [getSeconds](#) () const
Return the number of seconds.
- int [isSet](#) () const
Check if set.
- int [isLeapYear](#) ()
Returns if a leap year.
- void [addSeconds](#) (int seconds)
Add the given number of seconds.
- [BString](#) [getString](#) ([BString](#) format="isoT") const
Gets the date/time in string format.
- [BError](#) [setString](#) (const [BString](#) dateTime)
Sets the date/time from string format.
- int [operator==](#) (const [BTime](#) &time) const
- int [operator!=](#) (const [BTime](#) &time) const
- int [operator>](#) (const [BTime](#) &time) const
- int [operator>=](#) (const [BTime](#) &time) const
- int [operator<](#) (const [BTime](#) &time) const
- int [operator<=](#) (const [BTime](#) &time) const
- [BTime](#) [operator+](#) (int seconds) const
- [BTime](#) & [operator+=](#) (int seconds)

Private Attributes

- [BUInt32 otime](#)

Time in seconds since 1970. range 1970-01-02 to 2106-02-07.

6.83.1 Constructor & Destructor Documentation

6.83.1.1 `BTime::BTime (BUInt32 t = 0)`

6.83.2 Member Function Documentation

6.83.2.1 `void BTime::addSeconds (int seconds)`

Add the given number of seconds.

6.83.2.2 `void BTime::getDate (BUInt & year, BUInt & month, BUInt & day) const`

Return the date information.

6.83.2.3 `BUInt32 BTime::getSeconds () const`

Return the number of seconds.

6.83.2.4 `BString BTime::getString (BString format = "isoT") const`

Gets the date/time in string format.

6.83.2.5 `void BTime::getTime (BUInt & hour, BUInt & minute, BUInt & second) const`

Return the time information.

6.83.2.6 `int BTime::isLeapYear ()`

Returns if a leap year.

6.83.2.7 `int BTime::isSet () const` `[inline]`

Check if set.

6.83.2.8 `int BTime::operator!= (const BTime & time) const` `[inline]`

6.83.2.9 `BTime BTime::operator+ (int seconds) const` `[inline]`

6.83.2.10 `BTime& BTime::operator+= (int seconds)` `[inline]`

6.83.2.11 `int BTime::operator< (const BTime & time) const` `[inline]`

6.83.2.12 `int BTime::operator<= (const BTime & time) const` `[inline]`

6.83.2.13 `int BTime::operator== (const BTime & time) const` `[inline]`

6.83.2.14 `int BTime::operator> (const BTime & time) const` `[inline]`

6.83.2.15 `int BTime::operator>= (const BTime & time) const` `[inline]`

6.83.2.16 `void BTime::set (BUInt32 seconds)`

Set the date and time.

6.83.2.17 `void BTime::set (BUInt year, BUInt month, BUInt day, BUInt hour = 0, BUInt minute = 0, BUInt second = 0)`

Set the date and time.

6.83.2.18 `BError BTime::setString (const BString dateTime)`

Sets the date/time from string format.

6.83.2.19 `void BTime::setYearDay (BUInt year, BUInt yearDay, BUInt hour = 0, BUInt minute = 0, BUInt second = 0)`

Set the date and time.

6.83.3 Member Data Documentation

6.83.3.1 `BUInt32 BTime::otime` `[private]`

Time in seconds since 1970. range 1970-01-02 to 2106-02-07.

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

- [BTime.h](#)
- [BTime.cpp](#)

6.84 BTimer Class Reference

Stopwatch style timer.

```
#include <BTimer.h>
```

Public Member Functions

- [BTimer](#) ()
- [~BTimer](#) ()
- void [start](#) ()
Start timer.
- void [stop](#) ()
Stop timer.
- void [clear](#) ()
Clear timer.
- double [getElapsedTime](#) ()
Returns the elapsed time from the last start.
- void [add](#) ([BTimer](#) &timer)
Add two timers.

- double `average` ()
Average time is duration between `start()` and `stop()` / number of stops.
- double `peak` ()
Peak time.

Static Private Member Functions

- static double `getTime` ()

Private Attributes

- `BMutex` `oLock`
- unsigned int `onum`
- double `ostartTime`
- double `oendTime`
- double `oaverage`
- double `opeak`

6.84.1 Detailed Description

Stopwatch style timer.

6.84.2 Constructor & Destructor Documentation

6.84.2.1 `BTimer::BTimer ()`

6.84.2.2 `BTimer::~~BTimer ()`

6.84.3 Member Function Documentation

6.84.3.1 `void BTimer::add (BTimer & timer)`

Add two timers.

6.84.3.2 `double BTimer::average ()`

Average time is duration between `start()` and `stop()` / number of stops.

6.84.3.3 `void BTimer::clear ()`

Clear timer.

6.84.3.4 `double BTimer::getElapsedTime ()`

Returns the elapsed time from the last start.

6.84.3.5 `double BTimer::getTime ()` `[static], [private]`

6.84.3.6 `double BTimer::peak ()`

Peak time.

6.84.3.7 void BTimer::start ()

Start timer.

6.84.3.8 void BTimer::stop ()

Stop timer.

6.84.4 Member Data Documentation

6.84.4.1 double BTimer::oaverage [private]

6.84.4.2 double BTimer::oendTime [private]

6.84.4.3 BMutex BTimer::olock [private]

6.84.4.4 unsigned int BTimer::onum [private]

6.84.4.5 double BTimer::opeak [private]

6.84.4.6 double BTimer::ostartTime [private]

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

- [BTimer.h](#)
- [BTimer.cpp](#)

6.85 BTimeStamp Class Reference

```
#include <BTimeStamp.h>
```

Public Member Functions

- [BTimeStamp](#) ()
- [BTimeStamp](#) (int year, int month=1, int day=1, int hour=0, int minute=0, int second=0, int microsecond=0)
- [BTimeStamp](#) (const BString str)
- [~BTimeStamp](#) ()
- void [clear](#) ()
Clear the date/time.
- void [setFirst](#) ()
Set the first date available.
- void [setLast](#) ()
Set the last date available.
- void [set](#) (time_t time, int microSeconds)
Set time using Unix time (seconds from 1970-01-01)
- void [set](#) (int year=0, int month=1, int day=1, int hour=0, int minute=0, int second=0, int microsecond=0)
- void [set](#) (const BTimeStampMs &timeStamp)
Set the timeStamp to given MS time stamp.
- void [setYDay](#) (int year=0, int yday=0, int hour=0, int minute=0, int second=0, int microsecond=0)
- void [setTime](#) (int hour=0, int minute=0, int second=0, int microsecond=0)
- void [setNow](#) ()

Set the timeStamp to now.

- int [year](#) () const
- int [yday](#) () const
- int [month](#) () const
- int [day](#) () const
- int [hour](#) () const
- int [minute](#) () const
- int [second](#) () const
- int [microSecond](#) () const
- void [getDate](#) (int &[year](#), int &mon, int &[day](#)) const
- [BString](#) [getString](#) ([BString](#) separator="T") const

Get the time as an ISO date/time string.

- [BError](#) [setString](#) (const [BString](#) dateTime)

Set the time from an ISO date/time.

- [BString](#) [getStringNoMs](#) ([BString](#) separator="T") const

Get the time as an ISO date/time string without microseconds.

- [BString](#) [getStringFormatted](#) ([BString](#) format) const

Gets the time in a string form as per the format. Format syntax as per strftime()

- void [addMilliSeconds](#) (int milliSeconds)

Add the given number of milli seconds. This should be less than a year.

- void [addMicroSeconds](#) (int64_t microSeconds)

Add the given number of micro seconds. This should be less than a year.

- void [addSeconds](#) (int seconds)

Add the given number of seconds. This should be less than a year.

- uint32_t [getYearSeconds](#) () const

Get number of seconds within the year.

- uint64_t [getYearMicroSeconds](#) () const

Get number of micro seconds within the year.

- int [isSet](#) () const
- int [compare](#) (const [BTimeStamp](#) &timeStamp) const

Compare two dates.

- [operator](#) [BString](#) () const
- [BTimeStamp](#) & [operator=](#) (const [BTimeStampMs](#) &timeStamp)
- int [operator==](#) (const [BTimeStamp](#) &timeStamp) const
- int [operator!=](#) (const [BTimeStamp](#) &timeStamp) const
- int [operator>](#) (const [BTimeStamp](#) &timeStamp) const
- int [operator>=](#) (const [BTimeStamp](#) &timeStamp) const
- int [operator<](#) (const [BTimeStamp](#) &timeStamp) const
- int [operator<=](#) (const [BTimeStamp](#) &timeStamp) const

Static Public Member Functions

- static int [isLeap](#) (int [year](#))
- static [BInt64](#) [difference](#) ([BTimeStamp](#) t2, [BTimeStamp](#) t1)

Public Attributes

- uint16_t [oyear](#)
Year (0 .. 65535)
- uint16_t [oyday](#)
Day in year (0 .. 365)
- uint8_t [ohour](#)
Hour (0 .. 23)
- uint8_t [omminute](#)
Minute (0 .. 59)
- uint8_t [osecond](#)
Second (0 .. 59)
- uint8_t [ospare](#)
Padding.
- uint32_t [omicroSecond](#)
MicroSecond (0 .. 999999)

6.85.1 Constructor & Destructor Documentation

6.85.1.1 `BTimeStamp::BTimeStamp ()`

6.85.1.2 `BTimeStamp::BTimeStamp (int year, int month = 1, int day = 1, int hour = 0, int minute = 0, int second = 0, int microsecond = 0)`

6.85.1.3 `BTimeStamp::BTimeStamp (const BString str)`

6.85.1.4 `BTimeStamp::~~BTimeStamp ()`

6.85.2 Member Function Documentation

6.85.2.1 `void BTimeStamp::addMicroSeconds (int64_t microSeconds)`

Add the given number of micro seconds. This should be less that a year.

6.85.2.2 `void BTimeStamp::addMilliSeconds (int milliSeconds)`

Add the given number of milli seconds. This should be less that a year.

6.85.2.3 `void BTimeStamp::addSeconds (int seconds)`

Add the given number of seconds. This should be less that a year.

6.85.2.4 `void BTimeStamp::clear ()`

Clear the date/time.

6.85.2.5 `int BTimeStamp::compare (const BTimeStamp & timeStamp) const`

Compare two dates.

6.85.2.6 `int BTimeStamp::day () const`

6.85.2.7 `BInt64 BTimeStamp::difference (BTimeStamp t2, BTimeStamp t1) [static]`

6.85.2.8 `void BTimeStamp::getDate (int & year, int & mon, int & day) const`

6.85.2.9 `BString BTimeStamp::getString (BString separator = "T") const`

Get the time as an ISO date/time string.

6.85.2.10 `BString BTimeStamp::getStringFormatted (BString format) const`

Gets the time in a string form as per the format. Format syntax as per strftime()

6.85.2.11 `BString BTimeStamp::getStringNoMs (BString separator = "T") const`

Get the time as an ISO date/time string without microseconds.

6.85.2.12 `uint64_t BTimeStamp::getYearMicroSeconds () const`

Get number of micro seconds within the year.

6.85.2.13 `uint32_t BTimeStamp::getYearSeconds () const`

Get number of seconds within the year.

6.85.2.14 `int BTimeStamp::hour () const`

6.85.2.15 `int BTimeStamp::isLeap (int year) [static]`

6.85.2.16 `int BTimeStamp::isSet () const [inline]`

6.85.2.17 `int BTimeStamp::microSecond () const`

6.85.2.18 `int BTimeStamp::minute () const`

6.85.2.19 `int BTimeStamp::month () const`

6.85.2.20 `BTimeStamp::operator BString () const [inline]`

6.85.2.21 `int BTimeStamp::operator!= (const BTimeStamp & timeStamp) const [inline]`

6.85.2.22 `int BTimeStamp::operator< (const BTimeStamp & timeStamp) const [inline]`

6.85.2.23 `int BTimeStamp::operator<= (const BTimeStamp & timeStamp) const [inline]`

6.85.2.24 `BTimeStamp& BTimeStamp::operator= (const BTimeStampMs & timeStamp) [inline]`

6.85.2.25 `int BTimeStamp::operator== (const BTimeStamp & timeStamp) const [inline]`

6.85.2.26 `int BTimeStamp::operator> (const BTimeStamp & timeStamp) const [inline]`

6.85.2.27 `int BTimeStamp::operator>= (const BTimeStamp & timeStamp) const [inline]`

6.85.2.28 `int BTimeStamp::second () const`

6.85.2.29 `void BTimeStamp::set (time_t time, int microSeconds)`

Set time using Unix time (seconds from 1970-01-01)

6.85.2.30 `void BTimeStamp::set (int year = 0, int month = 1, int day = 1, int hour = 0, int minute = 0, int second = 0, int microsecond = 0)`

6.85.2.31 `void BTimeStamp::set (const BTimeStampMs & timeStamp)`

Set the timeStamp to given MS time stamp.

6.85.2.32 `void BTimeStamp::setFirst ()`

Set the first date available.

6.85.2.33 `void BTimeStamp::setLast ()`

Set the last date available.

6.85.2.34 `void BTimeStamp::setNow ()`

Set the timeStamp to now.

6.85.2.35 `BError BTimeStamp::setString (const BString dateTime)`

Set the time from an ISO date/time.

6.85.2.36 `void BTimeStamp::setTime (int hour = 0, int minute = 0, int second = 0, int microsecond = 0)`

6.85.2.37 `void BTimeStamp::setYDay (int year = 0, int yday = 0, int hour = 0, int minute = 0, int second = 0, int microsecond = 0)`

6.85.2.38 `int BTimeStamp::yday () const`

6.85.2.39 `int BTimeStamp::year () const`

6.85.3 Member Data Documentation

6.85.3.1 `uint8_t BTimeStamp::ohour`

Hour (0 .. 23)

6.85.3.2 `uint32_t BTimeStamp::omicroSecond`

MicroSecond (0 .. 999999)

6.85.3.3 `uint8_t BTimeStamp::ominate`

Minute (0 .. 59)

6.85.3.4 `uint8_t BTimeStamp::osecond`

Second (0 .. 59)

6.85.3.5 `uint8_t BTimeStamp::ospare`

Padding.

6.85.3.6 `uint16_t BTimeStamp::oyday`

Day in year (0 .. 365)

6.85.3.7 `uint16_t BTimeStamp::oyear`

Year (0 .. 65535)

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

- [BTimeStamp.h](#)
- [BTimeStamp.cpp](#)

6.86 BTimeStampMs Class Reference

```
#include <BTimeStampMs.h>
```

Public Member Functions

- [BTimeStampMs](#) ([BString](#) str="")
- [~BTimeStampMs](#) ()
- void [clear](#) ()
Clear the date/time.
- void [setNow](#) ()
Set the timeStamp to now.
- [BTimeStampMs](#) & [addMilliseconds](#) (int milliseconds)
Add the given number of milli seconds. This should be less than a year.
- [BTimeStampMs](#) & [subMilliseconds](#) (int milliseconds)
Add the given number of milli seconds. This should be less than a year.
- [BTimeStampMs](#) & [addSeconds](#) (int seconds)
Add the given number of seconds. This should be less than a year.
- [BTimeStampMs](#) & [subSeconds](#) (int seconds)
Subtract the given number of seconds. This should be less than a year.
- `uint32_t` [getYearSeconds](#) ()
Get number of seconds within the year.
- `uint64_t` [getYearMilliseconds](#) ()
Get number of seconds within the year.
- [BString](#) [getString](#) ([BString](#) separator="T")
Get the time as an ISO date/time string.
- [BString](#) [getStringNoMs](#) ([BString](#) separator="T")
Get the time as an ISO date/time string with no ms.
- [BError](#) [setString](#) ([BString](#) dateTime)

- Set the time from an ISO date/time.*

 - [BString getDurationString](#) ([BString](#) separator="T")

Get the time as an ISO date/time string but with month's and days starting from 0.

 - [BString getDurationStringNoMs](#) ([BString](#) separator="T")

Get the time as an ISO date/time string but with month's and days starting from 0 with no ms.

 - [BError setDurationString](#) ([BString](#) dateTime)

Set the time from an ISO date/time string but with month's and days starting from 0.

 - [BString getStringRaw](#) ()
 - void [getDate](#) (int &year, int &mon, int &day)

Get the year, month and day.

 - int [compare](#) (const [BTimeStampMs](#) &timeStamp)

Compare two dates.

 - int [operator>](#) (const [BTimeStampMs](#) &timeStamp)
 - int [operator>=](#) (const [BTimeStampMs](#) &timeStamp)
 - int [operator<](#) (const [BTimeStampMs](#) &timeStamp)
 - int [operator<=](#) (const [BTimeStampMs](#) &timeStamp)

Static Public Member Functions

- static int [isLeap](#) (int year)
- static [BUInt64 difference](#) ([BTimeStampMs](#) t2, [BTimeStampMs](#) t1)

Public Attributes

- uint16_t [year](#)
Year (2000 .. 3000)
- uint16_t [yday](#)
Day in year (0 .. 365)
- uint16_t [hour](#)
Hour (0 .. 23)
- uint16_t [minute](#)
Minute (0 .. 59)
- uint16_t [second](#)
Second (0 .. 59)
- uint16_t [milliSecond](#)
MilliSecond (0 .. 999)
- int32_t [sampleNumber](#)
The sample number this time refers to.

6.86.1 Constructor & Destructor Documentation

6.86.1.1 [BTimeStampMs::BTimeStampMs](#) ([BString](#) str = " ")

6.86.1.2 [BTimeStampMs::~~BTimeStampMs](#) ()

6.86.2 Member Function Documentation

6.86.2.1 [BTimeStampMs & BTimeStampMs::addMilliseconds](#) (int *milliseconds*)

Add the given number of milli seconds. This should be less that a year.

6.86.2.2 BTimeStampMs & BTimeStampMs::addSeconds (int seconds)

Add the given number of seconds. This should be less than a year.

6.86.2.3 void BTimeStampMs::clear ()

Clear the date/time.

6.86.2.4 int BTimeStampMs::compare (const BTimeStampMs & timeStamp)

Compare two dates.

6.86.2.5 BUInt64 BTimeStampMs::difference (BTimeStampMs t2, BTimeStampMs t1) [static]**6.86.2.6 void BTimeStampMs::getDate (int & year, int & mon, int & day)**

Get the year, month and day.

6.86.2.7 BString BTimeStampMs::getDurationString (BString separator = "T")

Get the time as an ISO date/time string but with month's and days starting from 0.

6.86.2.8 BString BTimeStampMs::getDurationStringNoMs (BString separator = "T")

Get the time as an ISO date/time string but with month's and days starting from 0 with no ms.

6.86.2.9 BString BTimeStampMs::getString (BString separator = "T")

Get the time as an ISO date/time string.

6.86.2.10 BString BTimeStampMs::getStringNoMs (BString separator = "T")

Get the time as an ISO date/time string with no ms.

6.86.2.11 BString BTimeStampMs::getStringRaw ()**6.86.2.12 uint64_t BTimeStampMs::getYearMilliseconds ()**

Get number of seconds within the year.

6.86.2.13 uint32_t BTimeStampMs::getYearSeconds ()

Get number of seconds within the year.

6.86.2.14 int BTimeStampMs::isLeap (int year) [static]**6.86.2.15 int BTimeStampMs::operator< (const BTimeStampMs & timeStamp) [inline]****6.86.2.16 int BTimeStampMs::operator<= (const BTimeStampMs & timeStamp) [inline]**

6.86.2.17 `int BTimeStampMs::operator> (const BTimeStampMs & timeStamp)` `[inline]`

6.86.2.18 `int BTimeStampMs::operator>= (const BTimeStampMs & timeStamp)` `[inline]`

6.86.2.19 `BError BTimeStampMs::setDurationString (BString dateTime)`

Set the time from an ISO date/time string but with month's and days starting from 0.

6.86.2.20 `void BTimeStampMs::setNow ()`

Set the timeStamp to now.

6.86.2.21 `BError BTimeStampMs::setString (BString dateTime)`

Set the time from an ISO date/time.

6.86.2.22 `BTimeStampMs & BTimeStampMs::subMilliseconds (int milliseconds)`

Add the given number of milli seconds. This should be less that a year.

6.86.2.23 `BTimeStampMs & BTimeStampMs::subSeconds (int seconds)`

Subtract the given number of seconds. This should be less that a year.

6.86.3 Member Data Documentation

6.86.3.1 `uint16_t BTimeStampMs::hour`

Hour (0 .. 23)

6.86.3.2 `uint16_t BTimeStampMs::milliSecond`

MilliSecond (0 .. 999)

6.86.3.3 `uint16_t BTimeStampMs::minute`

Minute (0 .. 59)

6.86.3.4 `int32_t BTimeStampMs::sampleNumber`

The sample number this time refers to.

6.86.3.5 `uint16_t BTimeStampMs::second`

Second (0 .. 59)

6.86.3.6 `uint16_t BTimeStampMs::yday`

Day in year (0 .. 365)

6.86.3.7 uint16_t BTimeStampMs::year

Year (2000 .. 3000)

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

- [BTimeStampMs.h](#)
- [BTimeStampMs.cpp](#)

6.87 BUrl Class Reference

Basic access to a Url.

```
#include <BUrl.h>
```

Public Member Functions

- [BUrl](#) ()
- [~BUrl](#) ()
- [BError readString](#) (BString url, BString &str)
Reads URL.

Static Private Member Functions

- static size_t [writeData](#) (void *data, size_t size, size_t elSize, void *stream)

Private Attributes

- [BString](#) ores

Static Private Attributes

- static int [oinit](#)

6.87.1 Detailed Description

Basic access to a Url.

6.87.2 Constructor & Destructor Documentation

6.87.2.1 [BUrl::BUrl](#) ()

6.87.2.2 [BUrl::~~BUrl](#) ()

6.87.3 Member Function Documentation

6.87.3.1 [BError BUrl::readString](#) (BString url, BString & str)

Reads URL.

6.87.3.2 `size_t BUrl::writeData (void * data, size_t size, size_t elSize, void * stream)` `[static]`, `[private]`

6.87.4 Member Data Documentation

6.87.4.1 `int BUrl::oinit` `[static]`, `[private]`

6.87.4.2 `BString BUrl::ores` `[private]`

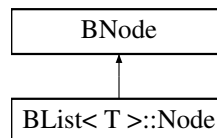
The documentation for this class was generated from the following files:

- [BUrl.h](#)
- [BUrl.cpp](#)

6.88 BList< T >::Node Class Reference

```
#include <BList.h>
```

Inheritance diagram for BList< T >::Node:



Public Member Functions

- [Node](#) (const T &i)

Public Attributes

- T [item](#)

6.88.1 Constructor & Destructor Documentation

6.88.1.1 `template<class T> BList< T >::Node::Node (const T & i)` `[inline]`

6.88.2 Member Data Documentation

6.88.2.1 `template<class T> T BList< T >::Node::item`

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

- [BList.h](#)

Chapter 7

File Documentation

7.1 BArray.h File Reference

```
#include <BTypes.h>
#include <vector>
#include <algorithm>
```

Classes

- class [BArray< T >](#)

Macros

- #define [BArrayLoop](#)(list, i) for([BUInt](#) i = 0; i < list.number(); i++)

7.1.1 Macro Definition Documentation

7.1.1.1 #define [BArrayLoop](#)(*list*, *i*) for([BUInt](#) i = 0; i < list.number(); i++)

7.2 BAtomic.h File Reference

```
#include <BTypes.h>
```

Classes

- class [BAtomic< Type >](#)
[BAtomic](#) class.

Typedefs

- typedef [BAtomic< BInt32 >](#) [BAtomicInt32](#)
- typedef [BAtomic< BInt64 >](#) [BAtomicInt64](#)
- typedef [BAtomic< BUInt32 >](#) [BAtomicUInt32](#)
- typedef [BAtomic< BUInt64 >](#) [BAtomicUInt64](#)

7.2.1 Typedef Documentation

7.2.1.1 typedef `BAtomic<BInt32> BAtomicInt32`

7.2.1.2 typedef `BAtomic<BInt64> BAtomicInt64`

7.2.1.3 typedef `BAtomic<BUInt32> BAtomicUInt32`

7.2.1.4 typedef `BAtomic<BUInt64> BAtomicUInt64`

7.3 BAtomicCount.h File Reference

```
#include <bits/atomicity.h>
```

Classes

- class [BAtomicCount](#)
BAtomicCount class.

7.4 BBuffer.cpp File Reference

```
#include <stdlib.h>
#include <memory.h>
#include <BBuffer.h>
#include <BEndian.h>
#include <BTimeStamp.h>
#include <BComplex.h>
```

Variables

- const int [roundSize](#) = 256

7.4.1 Variable Documentation

7.4.1.1 const int [roundSize](#) = 256

7.5 BBuffer.h File Reference

```
#include <BTypes.h>
#include <BString.h>
#include <BError.h>
#include <BComplex.h>
#include <BEndian.h>
```

Classes

- class [BBuffer](#)
- class [BBufferStore](#)

Macros

- `#define BBigEndian 0`

7.5.1 Macro Definition Documentation

7.5.1.1 `#define BBigEndian 0`

7.6 BComms.cpp File Reference

```
#include <BComms.h>
```

7.7 BComms.h File Reference

```
#include <BTypes.h>
#include <BEvent.h>
#include <BError.h>
```

Classes

- class [BComms](#)

7.8 BComplex.h File Reference

```
#include <BTypes.h>
#include <complex>
#include <algorithm>
```

Typedefs

- `typedef std::complex< double > BComplex`
- `typedef std::complex< float > BComplex32`
- `typedef std::complex< double > BComplex64`

7.8.1 Typedef Documentation

7.8.1.1 `typedef std::complex<double> BComplex`

This is based on the Standard C++ library complex class and has all of the functionality of that class.

7.8.1.2 `typedef std::complex<float> BComplex32`

7.8.1.3 `typedef std::complex<double> BComplex64`

7.9 BCond.cpp File Reference

```
#include <BCond.h>
#include <sys/time.h>
#include <stdio.h>
```

7.10 BCond.h File Reference

```
#include <pthread.h>
```

Classes

- class [BCond](#)

7.11 BCondInt.cpp File Reference

```
#include <BCondInt.h>
#include <sys/time.h>
#include <stdio.h>
#include <errno.h>
```

Functions

- static struct timespec [getTimeout](#) (uint32_t timeOutUs)

7.11.1 Function Documentation

7.11.1.1 static struct timespec [getTimeout](#) (uint32_t *timeOutUs*) [static]

7.12 BCondInt.h File Reference

```
#include <BTypes.h>
#include <pthread.h>
```

Classes

- class [BCondInt](#)
Thread conditional value.
- class [BCondValue](#)
Thread conditional value.
- class [BCondBool](#)
Thread conditional boolean.
- class [BCondWrap](#)
- class [BCondResource](#)
Resource lock.

7.13 BConfig.cpp File Reference

```
#include <BConfig.h>
#include <string.h>
```

7.14 BConfig.h File Reference

```
#include <BDict.h>
#include <BFile.h>
#include <BMutex.h>
```

Classes

- class [BConfig](#)
This class implements the configuration file access.

7.15 BCrc16.cpp File Reference

```
#include <BCrc16.h>
```

Functions

- [BUInt16 bcrc16](#) (void *buf, [BUInt16](#) len)

Variables

- static const [BUInt8 table_crc_hi](#) []
- static const [BUInt8 table_crc_lo](#) []

7.15.1 Function Documentation

7.15.1.1 [BUInt16 bcrc16](#) (void * buf, [BUInt16](#) len)

7.15.2 Variable Documentation

7.15.2.1 [const BUInt8 table_crc_hi\[\]](#) [static]

Initial value:

```
= {
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
```

```

    0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
}

```

7.15.2.2 const BUInt8 table_crc_lo[] [static]

Initial value:

```

= {
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
    0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
    0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
    0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
    0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
    0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
    0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
    0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
    0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
    0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
    0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
    0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
    0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
    0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
    0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
    0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
    0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
    0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
    0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
    0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
    0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
    0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
    0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
    0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
    0x43, 0x83, 0x41, 0x81, 0x80, 0x40
}

```

7.16 BCrc16.h File Reference

```
#include <BTypes.h>
```

Functions

- [BUInt16 bcrc16](#) (void *buf, BUInt16 len)

7.16.1 Function Documentation

7.16.1.1 BUInt16 bcrc16 (void * buf, BUInt16 len)

7.17 BDate.cpp File Reference

```
#include <BDate.h>
#include <sys/time.h>
```


Functions

- void [toBString](#) ([BDate](#) &*v*, [BString](#) &*s*)
- void [fromBString](#) ([BString](#) &*s*, [BDate](#) &*v*)

Variables

- static int [mon_yday](#) [2][13]

7.17.1 Function Documentation

7.17.1.1 void [fromBString](#) ([BString](#) & *s*, [BDate](#) & *v*)

7.17.1.2 void [toBString](#) ([BDate](#) & *v*, [BString](#) & *s*)

7.17.2 Variable Documentation

7.17.2.1 int [mon_yday](#)[2][13] [static]

Initial value:

```
= {
    { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 },
    { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366 }
}
```

7.18 BDate.h File Reference

```
#include <stdint.h>
#include <BError.h>
```

Classes

- class [BDate](#)

Functions

- void [toBString](#) ([BDate](#) &*v*, [BString](#) &*s*)
- void [fromBString](#) ([BString](#) &*s*, [BDate](#) &*v*)

7.18.1 Function Documentation

7.18.1.1 void [fromBString](#) ([BString](#) & *s*, [BDate](#) & *v*)

7.18.1.2 void [toBString](#) ([BDate](#) & *v*, [BString](#) & *s*)

7.19 BDebug.cpp File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <syslog.h>
#include <sys/time.h>
#include <stdarg.h>
#include <fcntl.h>
#include <execinfo.h>
#include <ctype.h>
#include <BDebug.h>
#include <errno.h>
#include <linux/unistd.h>
```

Macros

- `#define BTRACE_SIZE 100`

Functions

- void `hd8` (void *data, unsigned int n)
- void `hd8a` (void *data, unsigned int n)
- void `hda8` (void *data, unsigned int n)
- void `hd32` (void *data, unsigned int n)
- void `hda32` (void *data, unsigned int n)
- double `getTime` ()
- void `setDebug` (int d)
- void `fprintf` (int log, const char *fmt,...)
- pid_t `gettid` ()

Variables

- int `bdebug`
- const unsigned int `STRBUF_SIZE` = (64 * 1024)

7.19.1 Macro Definition Documentation

7.19.1.1 `#define BTRACE_SIZE 100`

7.19.2 Function Documentation

7.19.2.1 pid_t `gettid` ()

7.19.2.2 double `getTime` ()

7.19.2.3 void `hd32` (void * data, unsigned int n)

7.19.2.4 void `hd8` (void * data, unsigned int n)

7.19.2.5 void `hd8a` (void * data, unsigned int n)

7.19.2.6 void [hda32](#) (void * *data*, unsigned int *n*)

7.19.2.7 void [hda8](#) (void * *data*, unsigned int *n*)

7.19.2.8 void [setDebug](#) (int *d*)

7.19.2.9 void [tprintf](#) (int *log*, const char * *fmt*, ...)

7.19.3 Variable Documentation

7.19.3.1 int [bdebug](#)

7.19.3.2 const unsigned int STRBUF_SIZE = (64 * 1024)

7.20 BDebug.h File Reference

```
#include <stdio.h>
#include <syslog.h>
#include <time.h>
```

Classes

- class [BDebugBacktrace](#)

Macros

- #define [BDebug_STD](#) 0x000001
- #define [dprintf](#)(level, fmt, a...)

General debug functions.
- #define [nprintf](#)(fmt, a...) syslog(LOG_NOTICE, fmt, ##a)

Warnings and errors logging.
- #define [wprintf](#)(fmt, a...) syslog(LOG_WARNING, fmt, ##a)
- #define [eprintf](#)(fmt, a...) syslog(LOG_ERR, fmt, ##a)

Functions

- void [hd8](#) (void *data, unsigned int n)
- void [hd8a](#) (void *data, unsigned int n)
- void [hda8](#) (void *data, unsigned int n)
- void [hd32](#) (void *data, unsigned int n)
- void [hds32](#) (void *data, unsigned int n)
- double [getTime](#) ()
- void [setDebug](#) (int debug)
- void [tprintf](#) (int log, const char *fmt,...)
- pid_t [gettid](#) ()

Variables

- int [bdebug](#)

7.20.1 Macro Definition Documentation

7.20.1.1 `#define BDebug_STD 0x000001`

7.20.1.2 `#define dprintf(level, fmt, a...)`

General debug functions.

7.20.1.3 `#define eprintf(fmt, a...) syslog(LOG_ERR, fmt, ##a)`

7.20.1.4 `#define nprintf(fmt, a...) syslog(LOG_NOTICE, fmt, ##a)`

Warnings and errors logging.

7.20.1.5 `#define wprintf(fmt, a...) syslog(LOG_WARNING, fmt, ##a)`

7.20.2 Function Documentation

7.20.2.1 `pid_t gettid ()`

7.20.2.2 `double getTime ()`

7.20.2.3 `void hd32 (void * data, unsigned int n)`

7.20.2.4 `void hd8 (void * data, unsigned int n)`

7.20.2.5 `void hd8a (void * data, unsigned int n)`

7.20.2.6 `void hda8 (void * data, unsigned int n)`

7.20.2.7 `void hds32 (void * data, unsigned int n)`

7.20.2.8 `void setDebug (int debug)`

7.20.2.9 `void tprintf (int log, const char * fmt, ...)`

7.20.3 Variable Documentation

7.20.3.1 `int bdebug`

7.21 BDict.cpp File Reference

```
#include <BDict.h>
```

Functions

- void [toBString](#) (const [BDictString](#) &v, [BString](#) &s)
- void [fromBString](#) (const [BString](#) &str, [BDictString](#) &v)
- [BString](#) [bdictStringToString](#) (const [BDictString](#) &dict)

7.21.1 Function Documentation

7.21.1.1 **BString** bdictStringToString (const **BDictString** & *dict*)

7.21.1.2 void fromBString (const **BString** & *str*, **BDictString** & *v*)

7.21.1.3 void toBString (const **BDictString** & *v*, **BString** & *s*)

7.22 BDict.h File Reference

```
#include <BNameValue.h>
```

Classes

- class [BDictItem< Type >](#)
Template based Dictionary class.
- class [BDict< Type >](#)

Typedefs

- typedef [BDict< BString >](#) [BDictString](#)

Functions

- void [toBString](#) (const [BDictString](#) &*v*, [BString](#) &*s*)
- void [fromBString](#) (const [BString](#) &*s*, [BDictString](#) &*v*)
- [BString](#) [bdictStringToString](#) (const [BDictString](#) &*dict*)

7.22.1 Typedef Documentation

7.22.1.1 typedef [BDict<BString>](#) [BDictString](#)

7.22.2 Function Documentation

7.22.2.1 **BString** bdictStringToString (const **BDictString** & *dict*)

7.22.2.2 void fromBString (const **BString** & *s*, **BDictString** & *v*)

7.22.2.3 void toBString (const **BDictString** & *v*, **BString** & *s*)

7.23 BDictMap.h File Reference

```
#include <BString.h>
#include <map>
```

Classes

- class [BDictMap< Value >](#)

Typedefs

- typedef [BDictMap](#)< [BString](#) > [BDictMapString](#)

7.23.1 Typedef Documentation

7.23.1.1 typedef [BDictMap](#)<[BString](#)> [BDictMapString](#)

7.24 BDir.cpp File Reference

```
#include <BDir.h>
#include <dirent.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
```

Functions

- static int [wild](#) (const dirent *e)

Variables

- static [BString](#) [wildString](#)

7.24.1 Function Documentation

7.24.1.1 static int wild (const dirent * e) [static]

7.24.2 Variable Documentation

7.24.2.1 [BString](#) [wildString](#) [static]

7.25 BDir.h File Reference

```
#include <BList.h>
#include <BString.h>
#include <BError.h>
#include <sys/stat.h>
```

Classes

- class [BDir](#)
File system directory class.

7.26 BDuration.cpp File Reference

```
#include <BDuration.h>
#include <sys/time.h>
```

7.27 BDuration.h File Reference

```
#include <stdint.h>
#include <BError.h>
```

Classes

- class [BDuration](#)

7.28 BEndian.cpp File Reference

```
#include <BEndian.h>
#include <memory.h>
```

Functions

- void [bswap_copy](#) (int swap, const void *src, void *dst, [BUInt32](#) nBytes, const char *swapType)

7.28.1 Function Documentation

7.28.1.1 void [bswap_copy](#) (int *swap*, const void * *src*, void * *dst*, [BUInt32](#) *nBytes*, const char * *swapType*)

7.29 BEndian.h File Reference

```
#include <BTypes.h>
#include <byteswap.h>
```

Macros

- #define [htobe16](#)(x) __bswap_16 (x)
- #define [htole16](#)(x) (x)
- #define [be16toh](#)(x) __bswap_16 (x)
- #define [le16toh](#)(x) (x)
- #define [htobe32](#)(x) __bswap_32 (x)
- #define [htole32](#)(x) (x)
- #define [be32toh](#)(x) __bswap_32 (x)
- #define [le32toh](#)(x) (x)
- #define [htobe64](#)(x) __bswap_64 (x)
- #define [htole64](#)(x) (x)
- #define [be64toh](#)(x) __bswap_64 (x)
- #define [le64toh](#)(x) (x)

Functions

- void [bswap_p8](#) (const void *s, void *d)
- void [bswap_p16](#) (const void *s, void *d)
- void [bswap_p32](#) (const void *s, void *d)

- void [bswap_p64](#) (const void *s, void *d)
- void [bswap_copy](#) (int swap, const void *src, void *dst, [BUInt32](#) nBytes, const char *swapType)
- uint16_t [htole](#) (uint16_t v)
- int16_t [htole](#) (int16_t v)
- uint32_t [htole](#) (uint32_t v)
- int32_t [htole](#) (int32_t v)
- uint64_t [htole](#) (uint64_t v)
- int64_t [htole](#) (int64_t v)
- double [htole](#) (double v)
- float [htole](#) (float v)
- uint16_t [htobe](#) (uint16_t v)
- int16_t [htobe](#) (int16_t v)
- uint32_t [htobe](#) (uint32_t v)
- int32_t [htobe](#) (int32_t v)
- uint64_t [htobe](#) (uint64_t v)
- int64_t [htobe](#) (int64_t v)
- double [htobe](#) (double v)
- float [htobe](#) (float v)
- uint16_t [letoh](#) (uint16_t v)
- int16_t [letoh](#) (int16_t v)
- uint32_t [letoh](#) (uint32_t v)
- int32_t [letoh](#) (int32_t v)
- uint64_t [letoh](#) (uint64_t v)
- int64_t [letoh](#) (int64_t v)
- double [letoh](#) (double v)
- float [letoh](#) (float v)
- uint16_t [betoh](#) (uint16_t v)
- int16_t [betoh](#) (int16_t v)
- uint32_t [betoh](#) (uint32_t v)
- int32_t [betoh](#) (int32_t v)
- uint64_t [betoh](#) (uint64_t v)
- int64_t [betoh](#) (int64_t v)
- double [betoh](#) (double v)
- float [betoh](#) (float v)

7.29.1 Macro Definition Documentation

7.29.1.1 `#define be16toh(x) __bswap_16(x)`

7.29.1.2 `#define be32toh(x) __bswap_32(x)`

7.29.1.3 `#define be64toh(x) __bswap_64(x)`

7.29.1.4 `#define htobe16(x) __bswap_16(x)`

7.29.1.5 `#define htobe32(x) __bswap_32(x)`

7.29.1.6 `#define htobe64(x) __bswap_64(x)`

7.29.1.7 `#define htole16(x)(x)`

7.29.1.8 `#define htole32(x)(x)`

7.29.1.9 `#define htole64(x)(x)`

7.29.1.10 `#define le16toh(x)(x)`

7.29.1.11 `#define le32toh(x)(x)`

7.29.1.12 `#define le64toh(x)(x)`

7.29.2 Function Documentation

7.29.2.1 `uint16_t betoh (uint16_t v) [inline]`

7.29.2.2 `int16_t betoh (int16_t v) [inline]`

7.29.2.3 `uint32_t betoh (uint32_t v) [inline]`

7.29.2.4 `int32_t betoh (int32_t v) [inline]`

7.29.2.5 `uint64_t betoh (uint64_t v) [inline]`

7.29.2.6 `int64_t betoh (int64_t v) [inline]`

7.29.2.7 `double betoh (double v) [inline]`

7.29.2.8 `float betoh (float v) [inline]`

7.29.2.9 `void bswap_copy (int swap, const void * src, void * dst, BUInt32 nBytes, const char * swapType)`

7.29.2.10 `void bswap_p16 (const void * s, void * d) [inline]`

7.29.2.11 `void bswap_p32 (const void * s, void * d) [inline]`

7.29.2.12 `void bswap_p64 (const void * s, void * d) [inline]`

7.29.2.13 `void bswap_p8 (const void * s, void * d) [inline]`

7.29.2.14 `uint16_t htobe (uint16_t v) [inline]`

7.29.2.15 `int16_t htobe (int16_t v) [inline]`

7.29.2.16 `uint32_t htobe (uint32_t v) [inline]`

7.29.2.17 `int32_t htobe (int32_t v) [inline]`

7.29.2.18 `uint64_t htobe (uint64_t v) [inline]`

7.29.2.19 `int64_t htobe (int64_t v) [inline]`

7.29.2.20 `double htobe (double v) [inline]`

7.29.2.21 `float htobe (float v) [inline]`

7.29.2.22 `uint16_t htobe (uint16_t v) [inline]`

7.29.2.23 `int16_t htobe (int16_t v) [inline]`

7.29.2.24 `uint32_t htobe (uint32_t v) [inline]`

```

7.29.2.25  int32_t htogle ( int32_t v )  [inline]

7.29.2.26  uint64_t htogle ( uint64_t v )  [inline]

7.29.2.27  int64_t htogle ( int64_t v )  [inline]

7.29.2.28  double htogle ( double v )  [inline]

7.29.2.29  float htogle ( float v )  [inline]

7.29.2.30  uint16_t letoh ( uint16_t v )  [inline]

7.29.2.31  int16_t letoh ( int16_t v )  [inline]

7.29.2.32  uint32_t letoh ( uint32_t v )  [inline]

7.29.2.33  int32_t letoh ( int32_t v )  [inline]

7.29.2.34  uint64_t letoh ( uint64_t v )  [inline]

7.29.2.35  int64_t letoh ( int64_t v )  [inline]

7.29.2.36  double letoh ( double v )  [inline]

7.29.2.37  float letoh ( float v )  [inline]

```

7.30 BEntry.cpp File Reference

```

#include <ctype.h>
#include <stdio.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <BEntry.h>

```

7.31 BEntry.h File Reference

```

#include <BList.h>
#include <BString.h>

```

Classes

- class [BEntry](#)
Manipulate a name value pair.
- class [BEntryList](#)
List of Entries. Where an entry is a name value pair.
- class [BEntryFile](#)
File of Entries.

7.32 BError.cpp File Reference

```
#include <BError.h>
```

7.33 BError.h File Reference

```
#include <BString.h>
```

Classes

- class [BError](#)
Error return class.

Enumerations

- enum [BErrorNum](#) {
 [ErrorOk](#) = 0, [ErrorMisc](#) = 1, [ErrorWarning](#) = 2, [ErrorParam](#) = 3,
 [ErrorTimeout](#) = 4, [ErrorNotAvailable](#) = 5, [ErrorData](#) = 6, [ErrorChecksum](#) = 7,
 [ErrorOverrun](#) = 8, [ErrorUnderrun](#) = 9, [ErrorInit](#) = 10, [ErrorConfig](#) = 11,
 [ErrorNotImplemented](#) = 12, [ErrorResourceLimit](#) = 13, [ErrorEndOfFile](#) = 14, [ErrorFile](#) = 15,
 [ErrorFormat](#) = 16, [ErrorComms](#) = 17, [ErrorAccessDenied](#) = 18, [ErrorNoData](#) = 19,
 [ErrorEndOfData](#) = 20, [ErrorDataPresent](#) = 21, [ErrorAppBase](#) = 64 }

7.33.1 Enumeration Type Documentation

7.33.1.1 enum BErrorNum

Enumerator

ErrorOk

ErrorMisc

ErrorWarning

ErrorParam

ErrorTimeout

ErrorNotAvailable

ErrorData

ErrorChecksum

ErrorOverrun

ErrorUnderrun

ErrorInit

ErrorConfig

ErrorNotImplemented

ErrorResourceLimit

ErrorEndOfFile

ErrorFile

ErrorFormat

ErrorComms

ErrorAccessDenied
ErrorNoData
ErrorEndOfData
ErrorDataPresent
ErrorAppBase

7.34 BErrorTime.cpp File Reference

```
#include <BErrorTime.h>
```

7.35 BErrorTime.h File Reference

```
#include <BString.h>  
#include <BTimeStamp.h>
```

Classes

- class [BErrorTime](#)
Error return class.

7.36 BEvent.cpp File Reference

```
#include <BEvent.h>  
#include <BPoll.h>  
#include <stdlib.h>  
#include <unistd.h>  
#include <sys/ioctl.h>
```

7.37 BEvent.h File Reference

```
#include <BTypes.h>  
#include <BQueue.h>
```

Classes

- class [BEvent](#)
- class [BEventPipe](#)
This class provides an interface for sending simple integer events via a pipe file descriptor.

Typedefs

- typedef [BQueue](#)< [BEvent](#) > [BEventQueue](#)
This class provides an interface for sending simple integer events via a [BQueue](#).

Enumerations

- enum [BEventType](#) { [BEventTypeNone](#) = 0 }

7.37.1 Typedef Documentation

7.37.1.1 typedef BQueue<BEvent> BEventQueue

This class provides an interface for sending simple integer events via a [BQueue](#).

7.37.2 Enumeration Type Documentation

7.37.2.1 enum BEventType

Enumerator

BEventTypeNone

7.38 BEvent1.cpp File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <BEvent1.h>
#include <BPoll.h>
```

7.39 BEvent1.h File Reference

```
#include <stdint.h>
#include <BError.h>
```

Classes

- class [BEvent1](#)

This class provides a base class for all event objects that can be sent over the events interface.

- class [BEvent1Error](#)
- class [BEvent1Pipe](#)

This class provides a base interface for sending events via a pipe. This allows threads to send events that can be picked up by the poll system call.

- class [BEvent1Int](#)

This class provides an interface for sending simple integer events via a file descriptor. This allows threads to send events that can be picked up by the poll system call.

Enumerations

- enum [BEvent1Type](#) { [BEvent1TypeNone](#), [BEvent1TypeInt](#), [BEvent1TypeError](#) }

7.39.1 Enumeration Type Documentation

7.39.1.1 enum BEvent1Type

Enumerator

BEvent1TypeNone

BEvent1TypeInt

BEvent1TypeError

7.40 BFifo.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BMutex.h>
#include <BFifo.inc>
```

Classes

- class [BFifo< Type >](#)

7.41 BFifo.inc File Reference

7.42 BFifoCirc.cpp File Reference

```
#include <BFifoCirc.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/mman.h>
```

Macros

- #define [dprintf](#)(fmt, a...)

7.42.1 Macro Definition Documentation

7.42.1.1 #define dprintf(fmt, a...)

7.43 BFifoCirc.h File Reference

```
#include <stdint.h>
#include <BError.h>
#include <BCondInt.h>
#include <BMutex.h>
#include <BFifoCirc.inc>
```

Classes

- class [BFifoCircPos](#)
This class implements a pointer into the Fifo's circular buffer.
- class [BFifoCirc< Type >](#)
This class implements a thread safe FIFO buffer.

7.44 BFifoCirc.inc File Reference

7.45 BFile.cpp File Reference

```
#include <stdarg.h>
#include <BFile.h>
#include <sys/stat.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>
```

Macros

- #define [STRBUF](#) 10240

7.45.1 Macro Definition Documentation

7.45.1.1 #define STRBUF 10240

7.46 BFile.h File Reference

```
#include <stdio.h>
#include <BTypes.h>
#include <BString.h>
#include <BError.h>
```

Classes

- class [BFile](#)
File operations class.

7.47 BFileCsv.cpp File Reference

```
#include <BFileCsv.h>
#include <errno.h>
```

7.48 BFileCsv.h File Reference

```
#include <BFile.h>
```

Classes

- class [BFileCsv](#)

7.49 BFileData.cpp File Reference

```
#include <BFileCsv.h>
#include <BFileData.h>
#include <errno.h>
```

7.50 BFileData.h File Reference

```
#include <BError.h>
```

Classes

- class [BFileData](#)

7.51 BList.h File Reference

```
#include <BList_func.h>
```

Classes

- class [BNode](#)
- class [BIter](#)
Iterator for [BList](#).
- class [BList< T >](#)
Template based list class.
- class [BList< T >::Node](#)

Macros

- `#define BListLoop(list, i) for(BIter i = list.begin(); !list.isEnd(i); list.next(i))`

7.51.1 Macro Definition Documentation

7.51.1.1 `#define BListLoop(list, i) for(BIter i = list.begin(); !list.isEnd(i); list.next(i))`

7.52 BList_func.h File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <memory.h>
```

7.53 BMutex.cpp File Reference

```
#include <BMutex.h>
```

Macros

- `#define MDEBUG 0`

7.53.1 Macro Definition Documentation

7.53.1.1 `#define MDEBUG 0`

7.54 BMutex.h File Reference

```
#include <pthread.h>
```

Classes

- class [BMutex](#)
Mutex class.
- class [BMutexLock](#)

7.55 BMySQL.cpp File Reference

```
#include <stdlib.h>
#include <string.h>
#include <BMySQL.h>
```

7.56 BMySQL.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BDict.h>
#include <BMutex.h>
#include <mysql/mysql.h>
```

Classes

- class [BMySQL](#)

7.57 BNameValue.h File Reference

```
#include <BList.h>
#include <BString.h>
```

Classes

- class [BNameValue< T >](#)
- class [BNameValueList< T >](#)

7.58 Boap.cpp File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
#include <Boap.h>
#include <byteswap.h>
#include <BoapnsD.h>
#include <BoapnsC.h>
```

Macros

- `#define` [DEBUG](#) 0
- `#define` [APIVERSION_TEST](#) 1
- `#define` [dprintf](#)(fmt, a...)
- `#define` [IS_BIG_ENDIAN](#) 1

Variables

- `const int` [boapPort](#) = 12000
The default BOAP connection port.

7.58.1 Macro Definition Documentation

7.58.1.1 `#define` [APIVERSION_TEST](#) 1

7.58.1.2 `#define` [DEBUG](#) 0

7.58.1.3 `#define dprintf(fmt, a...)`

7.58.1.4 `#define IS_BIG_ENDIAN 1`

7.58.2 Variable Documentation

7.58.2.1 `const int boapPort = 12000`

The default BOAP connection port.

7.59 Boap.h File Reference

```
#include <stdint.h>
#include <BTypes.h>
#include <BPoll.h>
#include <BSocket.h>
#include <BThread.h>
#include <BError.h>
#include <BEvent1.h>
#include <BMutex.h>
#include <BTimeStamp.h>
#include <BBuffer.h>
```

Classes

- struct [BoapPacketHead](#)
- class [BoapPacket](#)
- class [BoapClientObject](#)
- class [BoapSignalObject](#)
- class [BoapServiceEntry](#)
- class [BoapServerConnection](#)
- class [BoapServer](#)
- class [BoapFuncEntry](#)
- class [BoapServiceObject](#)

Namespaces

- [Boapns](#)

Typedefs

- typedef [BUInt32](#) [BoapService](#)
- typedef [BError](#)([BoapServiceObject::*](#) [BoapFunc](#))([BoapServerConnection](#) *conn, [BoapPacket](#) &rx, [BoapPacket](#) &tx)

Enumerations

- enum [BoapType](#) {
 [BoapTypeRpc](#), [BoapTypeRpcReply](#), [BoapTypeSignal](#), [BoapTypeRpcError](#),
 [BoapTypeRpc](#), [BoapTypeSignal](#) }
- enum [BoapPriority](#) { [BoapPriorityLow](#), [BoapPriorityNormal](#), [BoapPriorityHigh](#) }

Variables

- const [BUInt32 BoapMagic](#) = 0x424F4100

7.59.1 Typedef Documentation

7.59.1.1 typedef BError(BoapServiceObject::* BoapFunc)(BoapServerConnection *conn, BoapPacket &rx, BoapPacket &tx)

7.59.1.2 typedef BUInt32 BoapService

7.59.2 Enumeration Type Documentation

7.59.2.1 enum BoapPriority

Enumerator

BoapPriorityLow
BoapPriorityNormal
BoapPriorityHigh

7.59.2.2 enum BoapType

Enumerator

BoapTypeRpc
BoapTypeRpcReply
BoapTypeSignal
BoapTypeRpcError
BoapTypeRpc
BoapTypeSignal

7.59.3 Variable Documentation

7.59.3.1 const [BUInt32 BoapMagic](#) = 0x424F4100

7.60 BoapMc.cpp File Reference

```
#include <BoapMc.h>
#include <BCrc16.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
```

Macros

- #define [DEBUG_LOCAL](#) 0
- #define [DEBUG_LOCAL1](#) 0
- #define [dlprintf](#)(fmt, a...)
- #define [dl1printf](#)(fmt, a...)

7.60.1 Macro Definition Documentation

7.60.1.1 `#define DEBUG_LOCAL 0`

7.60.1.2 `#define DEBUG_LOCAL1 0`

7.60.1.3 `#define dl1printf(fmt, a...)`

7.60.1.4 `#define dlprintf(fmt, a...)`

7.61 BoapMc.h File Reference

```
#include <BTypes.h>
#include <BMutex.h>
#include <BSemaphore.h>
#include <BQueue.h>
#include <BFifo.h>
#include <BComms.h>
```

Classes

- struct [BoapMcPacketHead](#)
- class [BoapMcPacket](#)
- class [BoapMcClientObject](#)
- class [BoapMcSignalObject](#)
- class [BoapMcServiceObject](#)
- class [BoapMcComms](#)

Enumerations

- enum [BoapMcType](#) { [BoapMcTypeRequest](#) = 0x00, [BoapMcTypeReply](#) = 0x80 }

Functions

- struct [BoapMcPacketHead](#) `__attribute__((aligned(8), packed))`

Variables

- [BUInt8](#) length
- [BUInt8](#) addressTo
- [BUInt8](#) addressFrom
- [BUInt8](#) cmd
- [BUInt16](#) error
- [BUInt16](#) checksum
- class [BoapMcPacket](#) `__attribute__((aligned(8), packed))`

7.61.1 Enumeration Type Documentation

7.61.1.1 enum [BoapMcType](#)

Enumerator

[BoapMcTypeRequest](#)

BoapMcTypeReply

7.61.2 Function Documentation

7.61.2.1 struct BoapMcPacketHead __attribute__ ((aligned(8), packed))

7.61.3 Variable Documentation

7.61.3.1 class BoapMcPacket __attribute__

7.61.3.2 BUInt8 addressFrom

7.61.3.3 BUInt8 addressTo

7.61.3.4 BUInt16 checksum

7.61.3.5 BUInt8 cmd

7.61.3.6 BUInt16 error

7.61.3.7 BUInt8 length

7.62 BoapnsC.cpp File Reference

```
#include <BoapnsC.h>
```

Namespaces

- [Boapns](#)

7.63 BoapnsC.h File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <Boap.h>
#include <BString.h>
#include <BList.h>
#include <BArray.h>
#include <BoapnsD.h>
```

Classes

- class [Boapns::Boapns](#)

Namespaces

- [Boapns](#)

Variables

- const [BUInt32 Boapns::apiVersion](#) = 0

7.64 BoapnsD.cpp File Reference

```
#include <BoapnsD.h>
```

Namespaces

- [Boapns](#)

7.65 BoapnsD.h File Reference

```
#include <Boap.h>
#include <BObj.h>
#include <BDate.h>
#include <BTimeStamp.h>
#include <BComplex.h>
#include <BList.h>
#include <BArray.h>
```

Classes

- class [Boapns::BoapEntry](#)

Namespaces

- [Boapns](#)

7.66 BoapSimple.cc File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <Boap.h>
#include <BoapnsD.h>
#include <BoapnsC.h>
```

Macros

- #define [DEBUG](#) 0
- #define [dprintf](#)(fmt, a...)

Variables

- const int [roundSize](#) = 256

7.66.1 Macro Definition Documentation

7.66.1.1 `#define DEBUG 0`

7.66.1.2 `#define dprintf(fmt, a...)`

7.66.2 Variable Documentation

7.66.2.1 `const int roundSize = 256`

7.67 BoapSimple.h File Reference

```
#include <stdint.h>
#include <BPoll.h>
#include <BSocket.h>
#include <BError.h>
```

Classes

- struct [BoapPacketHead](#)
- class [BoapPacket](#)
- class [BoapClientObject](#)
- class [BoapSignalObject](#)
- class [BoapServiceEntry](#)
- class [BoapServer](#)
- class [BoapFuncEntry](#)
- class [BoapServiceObject](#)

Typedefs

- typedef int8_t [Int8](#)
- typedef uint8_t [UInt8](#)
- typedef int16_t [Int16](#)
- typedef uint16_t [UInt16](#)
- typedef int32_t [Int32](#)
- typedef uint32_t [UInt32](#)
- typedef double [Double](#)
- typedef uint32_t [BoapService](#)
- typedef [BError](#)(BoapServiceObject::* [BoapFunc](#))(BoapPacket &rx, BoapPacket &tx)

Enumerations

- enum [BoapType](#) {
[BoapTypeRpc](#), [BoapTypeRpcReply](#), [BoapTypeSignal](#), [BoapTypeRpcError](#),
[BoapTypeRpc](#), [BoapTypeSignal](#) }

7.67.1 Typedef Documentation

7.67.1.1 `typedef BError(BoapServiceObject::* BoapFunc)(BoapPacket &rx, BoapPacket &tx)`

7.67.1.2 `typedef uint32_t BoapService`

7.67.1.3 `typedef double Double`

7.67.1.4 `typedef int16_t Int16`

7.67.1.5 `typedef int32_t Int32`

7.67.1.6 `typedef int8_t Int8`

7.67.1.7 `typedef uint16_t UInt16`

7.67.1.8 `typedef uint32_t UInt32`

7.67.1.9 `typedef uint8_t UInt8`

7.67.2 Enumeration Type Documentation

7.67.2.1 `enum BoapType`

Enumerator

BoapTypeRpc

BoapTypeRpcReply

BoapTypeSignal

BoapTypeRpcError

BoapTypeRpc

BoapTypeSignal

7.68 BObj.cpp File Reference

```
#include <BObj.h>
```

7.69 BObj.h File Reference

```
#include <BTypes.h>
#include <BDict.h>
#include <BString.h>
#include <BError.h>
```

Classes

- class [BObj](#)

7.70 BObjStringFormat.cpp File Reference

```
#include <BObjStringFormat.h>
#include <BTime.h>
#include <math.h>
```

Functions

- [BString toBString \(BString n, Bool v\)](#)
- [BString toBString \(BString n, BInt8 v\)](#)
- [BString toBString \(BString n, BUInt8 v\)](#)
- [BString toBString \(BString n, BInt16 v\)](#)
- [BString toBString \(BString n, BUInt16 v\)](#)
- [BString toBString \(BString n, BInt32 v\)](#)
- [BString toBString \(BString n, BUInt32 v\)](#)
- [BString toBString \(BString n, BInt64 v\)](#)
- [BString toBString \(BString n, BUInt64 v\)](#)
- [BString toBString \(BString n, BFloat32 v\)](#)
- [BString toBString \(BString n, BFloat64 v\)](#)
- [BString toBString \(BString n, BChar v\)](#)
- [BString toBString \(BString n, const BChar *v\)](#)
- [BString toBString \(BString n, BString v\)](#)
- [BString toBString \(BString n, BError v\)](#)
- [BString toBString \(BString n, BTime v\)](#)
- [BString toBString \(BString name, const BObjMember *m, const void *obj, BStringList ignoreFields\)](#)
- [BString toBString \(BString n, BObj &obj\)](#)
- [BString toBStringJson \(BString n, Bool v\)](#)
- [BString toBStringJson \(BString n, BInt8 v\)](#)
- [BString toBStringJson \(BString n, BUInt8 v\)](#)
- [BString toBStringJson \(BString n, BInt16 v\)](#)
- [BString toBStringJson \(BString n, BUInt16 v\)](#)
- [BString toBStringJson \(BString n, BInt32 v\)](#)
- [BString toBStringJson \(BString n, BUInt32 v\)](#)
- [BString toBStringJson \(BString n, BInt64 v\)](#)
- [BString toBStringJson \(BString n, BUInt64 v\)](#)
- [BString toBStringJson \(BString n, BFloat32 v\)](#)
- [BString toBStringJson \(BString n, BFloat64 v\)](#)
- [BString toBStringJson \(BString n, BChar v\)](#)
- [BString toBStringJson \(BString n, const BChar *v\)](#)
- [BString toBStringJson \(BString n, BString v\)](#)
- [BString toBStringJson \(BString n, BError v\)](#)
- [BString toBStringJson \(BString n, BTime v\)](#)
- [BString toBStringJson \(BString n, const BObjMember *m, const void *obj, BStringList ignoreFields\)](#)
- [BString toBStringJson \(BString n, BObj &obj\)](#)
- [BError toBDictStringFromJson \(BString json, BDictString &ds\)](#)

7.70.1 Function Documentation

- 7.70.1.1 **BError toBDictStringFromJson** (**BString** *json*, **BDictString** & *ds*)
- 7.70.1.2 **BString toBString** (**BString** *n*, **Bool** *v*)
- 7.70.1.3 **BString toBString** (**BString** *n*, **BInt8** *v*)
- 7.70.1.4 **BString toBString** (**BString** *n*, **BUInt8** *v*)
- 7.70.1.5 **BString toBString** (**BString** *n*, **BInt16** *v*)
- 7.70.1.6 **BString toBString** (**BString** *n*, **BUInt16** *v*)
- 7.70.1.7 **BString toBString** (**BString** *n*, **BInt32** *v*)
- 7.70.1.8 **BString toBString** (**BString** *n*, **BUInt32** *v*)
- 7.70.1.9 **BString toBString** (**BString** *n*, **BInt64** *v*)
- 7.70.1.10 **BString toBString** (**BString** *n*, **BUInt64** *v*)
- 7.70.1.11 **BString toBString** (**BString** *n*, **BFloat32** *v*)
- 7.70.1.12 **BString toBString** (**BString** *n*, **BFloat64** *v*)
- 7.70.1.13 **BString toBString** (**BString** *n*, **BChar** *v*)
- 7.70.1.14 **BString toBString** (**BString** *n*, **const BChar** * *v*)
- 7.70.1.15 **BString toBString** (**BString** *n*, **BString** *v*)
- 7.70.1.16 **BString toBString** (**BString** *n*, **BError** *v*)
- 7.70.1.17 **BString toBString** (**BString** *n*, **BTime** *v*)
- 7.70.1.18 **BString toBString** (**BString** *name*, **const BObjMember** * *m*, **const void** * *obj*, **BStringList** *ignoreFields*)
- 7.70.1.19 **BString toBString** (**BString** *n*, **BObj** & *obj*)
- 7.70.1.20 **BString toBStringJson** (**BString** *n*, **Bool** *v*)
- 7.70.1.21 **BString toBStringJson** (**BString** *n*, **BInt8** *v*)
- 7.70.1.22 **BString toBStringJson** (**BString** *n*, **BUInt8** *v*)
- 7.70.1.23 **BString toBStringJson** (**BString** *n*, **BInt16** *v*)
- 7.70.1.24 **BString toBStringJson** (**BString** *n*, **BUInt16** *v*)
- 7.70.1.25 **BString toBStringJson** (**BString** *n*, **BInt32** *v*)
- 7.70.1.26 **BString toBStringJson** (**BString** *n*, **BUInt32** *v*)
- 7.70.1.27 **BString toBStringJson** (**BString** *n*, **BInt64** *v*)

- 7.70.1.28 **BString toBStringJson** (**BString** *n*, **BUInt64** *v*)
- 7.70.1.29 **BString toBStringJson** (**BString** *n*, **BFloat32** *v*)
- 7.70.1.30 **BString toBStringJson** (**BString** *n*, **BFloat64** *v*)
- 7.70.1.31 **BString toBStringJson** (**BString** *n*, **BChar** *v*)
- 7.70.1.32 **BString toBStringJson** (**BString** *n*, const **BChar** * *v*)
- 7.70.1.33 **BString toBStringJson** (**BString** *n*, **BString** *v*)
- 7.70.1.34 **BString toBStringJson** (**BString** *n*, **BError** *v*)
- 7.70.1.35 **BString toBStringJson** (**BString** *n*, **BTime** *v*)
- 7.70.1.36 **BString toBStringJson** (**BString** *n*, const **BObjMember** * *m*, const void * *obj*, **BStringList** *ignoreFields*)
- 7.70.1.37 **BString toBStringJson** (**BString** *n*, **BObj** & *obj*)

7.71 BObjStringFormat.h File Reference

```
#include <BObj.h>
#include <BString.h>
#include <BTime.h>
```

Functions

- **BString toBString** (**BString** name, **Bool** value)
- **BString toBString** (**BString** name, **BInt8** value)
- **BString toBString** (**BString** name, **BUInt8** value)
- **BString toBString** (**BString** name, **BInt16** value)
- **BString toBString** (**BString** name, **BUInt16** value)
- **BString toBString** (**BString** name, **BInt32** value)
- **BString toBString** (**BString** name, **BUInt32** value)
- **BString toBString** (**BString** name, **BFloat32** value)
- **BString toBString** (**BString** name, **BFloat64** value)
- **BString toBString** (**BString** name, **BChar** value)
- **BString toBString** (**BString** name, const **BChar** *value)
- **BString toBString** (**BString** name, **BString** value)
- **BString toBString** (**BString** name, **BError** value)
- **BString toBString** (**BString** name, **BTime** time)
- **BString toBString** (**BString** name, const **BObjMember** *members, const void *obj, **BStringList** ignoreFields=**BStringList**())
- **BString toBString** (**BString** name, **BObj** &obj)
- **BString toBStringJson** (**BString** name, **Bool** value)
- **BString toBStringJson** (**BString** name, **BInt8** value)
- **BString toBStringJson** (**BString** name, **BUInt8** value)
- **BString toBStringJson** (**BString** name, **BInt16** value)
- **BString toBStringJson** (**BString** name, **BUInt16** value)
- **BString toBStringJson** (**BString** name, **BInt32** value)
- **BString toBStringJson** (**BString** name, **BUInt32** value)
- **BString toBStringJson** (**BString** name, **BFloat32** value)

- [BString toBStringJson](#) ([BString](#) name, [BFloat64](#) value)
- [BString toBStringJson](#) ([BString](#) name, [BChar](#) value)
- [BString toBStringJson](#) ([BString](#) name, const [BChar](#) *value)
- [BString toBStringJson](#) ([BString](#) name, [BString](#) value)
- [BString toBStringJson](#) ([BString](#) name, [BError](#) value)
- [BString toBStringJson](#) ([BString](#) name, [BTime](#) time)
- [BString toBStringJson](#) ([BString](#) name, const [BObjMember](#) *members, const void *obj, [BStringList](#) ignoreFields=[BStringList](#)())
- [BString toBStringJson](#) ([BString](#) name, [BObj](#) &obj)
- [BError toBDictStringFromJson](#) ([BString](#) json, [BDictString](#) &ds)
- [BString base64_encode](#) (void *data, [BUInt](#) len)
- [BError base64_decode](#) ([BString](#) strIn, [BString](#) &strOut)

7.71.1 Function Documentation

7.71.1.1 [BError base64_decode](#) ([BString](#) *strIn*, [BString](#) & *strOut*)

7.71.1.2 [BString base64_encode](#) (void * *data*, [BUInt](#) *len*)

7.71.1.3 [BError toBDictStringFromJson](#) ([BString](#) *json*, [BDictString](#) & *ds*)

7.71.1.4 [BString toBString](#) ([BString](#) *name*, [Bool](#) *value*)

7.71.1.5 [BString toBString](#) ([BString](#) *name*, [BInt8](#) *value*)

7.71.1.6 [BString toBString](#) ([BString](#) *name*, [BUInt8](#) *value*)

7.71.1.7 [BString toBString](#) ([BString](#) *name*, [BInt16](#) *value*)

7.71.1.8 [BString toBString](#) ([BString](#) *name*, [BUInt16](#) *value*)

7.71.1.9 [BString toBString](#) ([BString](#) *name*, [BInt32](#) *value*)

7.71.1.10 [BString toBString](#) ([BString](#) *name*, [BUInt32](#) *value*)

7.71.1.11 [BString toBString](#) ([BString](#) *name*, [BFloat32](#) *value*)

7.71.1.12 [BString toBString](#) ([BString](#) *name*, [BFloat64](#) *value*)

7.71.1.13 [BString toBString](#) ([BString](#) *name*, [BChar](#) *value*)

7.71.1.14 [BString toBString](#) ([BString](#) *name*, const [BChar](#) * *value*)

7.71.1.15 [BString toBString](#) ([BString](#) *name*, [BString](#) *value*)

7.71.1.16 [BString toBString](#) ([BString](#) *name*, [BError](#) *value*)

7.71.1.17 [BString toBString](#) ([BString](#) *name*, [BTime](#) *time*)

7.71.1.18 [BString toBString](#) ([BString](#) *name*, const [BObjMember](#) * *members*, const void * *obj*, [BStringList](#) *ignoreFields* =[BStringList](#)())

7.71.1.19 [BString toBString](#) ([BString](#) *name*, [BObj](#) & *obj*)

7.71.1.20 [BString toBStringJson](#) ([BString](#) *name*, [Bool](#) *value*)

- 7.71.1.21 **BString toBStringJson** (**BString** *name*, **BInt8** *value*)
- 7.71.1.22 **BString toBStringJson** (**BString** *name*, **BUInt8** *value*)
- 7.71.1.23 **BString toBStringJson** (**BString** *name*, **BInt16** *value*)
- 7.71.1.24 **BString toBStringJson** (**BString** *name*, **BUInt16** *value*)
- 7.71.1.25 **BString toBStringJson** (**BString** *name*, **BInt32** *value*)
- 7.71.1.26 **BString toBStringJson** (**BString** *name*, **BUInt32** *value*)
- 7.71.1.27 **BString toBStringJson** (**BString** *name*, **BFloat32** *value*)
- 7.71.1.28 **BString toBStringJson** (**BString** *name*, **BFloat64** *value*)
- 7.71.1.29 **BString toBStringJson** (**BString** *name*, **BChar** *value*)
- 7.71.1.30 **BString toBStringJson** (**BString** *name*, **const BChar *** *value*)
- 7.71.1.31 **BString toBStringJson** (**BString** *name*, **BString** *value*)
- 7.71.1.32 **BString toBStringJson** (**BString** *name*, **BError** *value*)
- 7.71.1.33 **BString toBStringJson** (**BString** *name*, **BTime** *time*)
- 7.71.1.34 **BString toBStringJson** (**BString** *name*, **const BObjMember *** *members*, **const void *** *obj*, **BStringList** *ignoreFields* = **BStringList()**)
- 7.71.1.35 **BString toBStringJson** (**BString** *name*, **BObj &** *obj*)

7.72 BPoll.cpp File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <BPoll.h>
```

7.73 BPoll.h File Reference

```
#include <BList.h>
#include <BError.h>
#include <sys/poll.h>
```

Classes

- class [BPoll](#)

This class provides an interface for polling a number of file descriptors. It uses round robin polling.

7.74 BQueue.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BList.h>
#include <BMutex.h>
#include <BCondInt.h>
```

Classes

- class [BQueue< T >](#)

Queue class.

Typedefs

- typedef [BQueue< BInt32 >](#) [BQueueInt](#)

7.74.1 Typedef Documentation

7.74.1.1 typedef [BQueue<BInt32>](#) [BQueueInt](#)

7.75 BRefData.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <BRefData.h>
```

Macros

- #define [CHUNK](#) 16

7.75.1 Macro Definition Documentation

7.75.1.1 #define [CHUNK](#) 16

7.76 BRefData.h File Reference

```
#include <BAAtomicCount.h>
```

Classes

- class [BRefData](#)

7.77 BRtc.cpp File Reference

```
#include <BRtc.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <linux/rtc.h>
```

7.78 BRtc.h File Reference

```
#include <BError.h>
#include <BThread.h>
#include <BCond.h>
```

Classes

- class [BRtc](#)
Realtime clock.
- class [BRtcThreaded](#)
Threaded real time clock.

7.79 BRWLock.cpp File Reference

```
#include <BRWLock.h>
```

7.80 BRWLock.h File Reference

```
#include <pthread.h>
```

Classes

- class [BRWLock](#)
thread read-write locks

7.81 BSema.cpp File Reference

```
#include <BSema.h>
#include <errno.h>
#include <sys/time.h>
```


7.82 BSema.h File Reference

```
#include <sys/types.h>
#include <semaphore.h>
```

Classes

- class [BSema](#)
Sempahore class.

7.83 BSemaphore.cpp File Reference

```
#include <BSemaphore.h>
#include <sys/time.h>
```

7.84 BSemaphore.h File Reference

```
#include <BTypes.h>
#include <BMutex.h>
#include <semaphore.h>
```

Classes

- class [BSemaphore](#)
Semaphore class.
- class [BSemaphoreCount](#)

7.85 BSocket.cpp File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <net/if.h>
#include "BSocket.h"
```

Macros

- #define [IP_MTU](#) 14

7.85.1 Macro Definition Documentation

7.85.1.1 #define IP_MTU 14

7.86 BSocket.h File Reference

```
#include <BString.h>
#include <BError.h>
#include <BTypes.h>
#include <stdint.h>
#include <sys/types.h>
#include <sys/prctl.h>
```

Classes

- class [BSocketAddress](#)
Socket Address.
- class [BSocketAddressINET](#)
IP aware socket address.
- class [BSocket](#)

7.87 BSpi.cpp File Reference

```
#include <BSpi.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/ioctl.h>
#include <linux/spi/spidev.h>
```

7.88 BSpi.h File Reference

```
#include <BTypes.h>
#include <BError.h>
```

Classes

- class [BSpi](#)
BSpi class.

7.89 BString.cpp File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdarg.h>
#include <ctype.h>
#include <BString.h>
#include <BError.h>
#include <regex.h>
```

Macros

- `#define STRIP 0x7f`
- `#define MINUS '-'`

Functions

- static int `gmatch` (const char *s, const char *p)
- std::ostream & `operator<<` (std::ostream &o, BString &s)
- std::istream & `operator>>` (std::istream &i, BString &s)
- int `bstringListinList` (BStringList &list, BString s)
- BString `blistToString` (const BStringList &list)
- BStringList `bstringToList` (BString str, int stripSpaces)
- BStringList `charToList` (const char **str)
- BString `barrayToString` (const BStringArray &list)
- BStringArray `bstringToArray` (BString str, int stripSpaces)
- BStringArray `charToArray` (const char **str)
- void `toBString` (BString &v, BString &s)
- void `toBString` (BStringList &v, BString &s)
- void `toBString` (BInt32 &v, BString &s)
- void `toBString` (BUInt32 &v, BString &s)
- void `toBString` (BUInt64 &v, BString &s)
- void `toBString` (BFloat64 &v, BString &s)
- void `fromBString` (BString &s, BString &v)
- void `fromBString` (BString &s, BStringList &v)
- void `fromBString` (BString &s, BInt32 &v)
- void `fromBString` (BString &s, BUInt32 &v)
- void `fromBString` (BString &s, BUInt64 &v)
- void `fromBString` (BString &s, BFloat64 &v)

Variables

- static const BUInt8 `base64_decode_table` []

7.89.1 Macro Definition Documentation

7.89.1.1 `#define MINUS '-'`

7.89.1.2 `#define STRIP 0x7f`

7.90 BString.h File Reference

Classes

- ## Functions

- ### 7.90.1 Function Documentation

7.90.1.8 `std::istream& operator>> (std::istream & i, BString & s)`

7.90.1.9 void toBString (BString & v, BString & s)

7.90.1.10 void toBString (BStringList & v, BString & s)

7.90.1.11 void toBString (BInt32 & v, BString & s)

7.90.1.12 void toBString (BUInt32 & v, BString & s)

7.90.1.13 void toBString (BUInt64 & v, BString & s)

7.90.1.14 void toBString (BFloat64 & v, BString & s)

7.91 BStringLocked.h File Reference

```
#include <BString.h>
#include <BMutex.h>
```

Classes

- class [BStringMutex](#)
- class [BStringLocked](#)

7.92 BTable.cpp File Reference

```
#include <BTable.h>
```

7.93 BTable.h File Reference

```
#include <BArray.h>
#include <BString.h>
```

Classes

- class [BTable](#)

7.94 BThread.cpp File Reference

```
#include <BThread.h>
#include <unistd.h>
#include <errno.h>
#include <sys/types.h>
```

7.95 BThread.h File Reference

```
#include <pthread.h>
```

Classes

- class [BThread](#)

7.96 BTime.cpp File Reference

```
#include <BTime.h>
```

Functions

- static bool [yearIsLeap](#) ([BUInt16](#) year)
- static [BUInt16](#) [yearDays](#) ([BUInt16](#) year)

Variables

- static [BUInt16](#) [monDays](#) [2][13]

7.96.1 Function Documentation

7.96.1.1 static [BUInt16](#) [yearDays](#) ([BUInt16](#) year) [inline],[static]

7.96.1.2 static bool [yearIsLeap](#) ([BUInt16](#) year) [inline],[static]

7.96.2 Variable Documentation

7.96.2.1 [BUInt16](#) [monDays](#)[2][13] [static]

Initial value:

```
= {  
    { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 },  
    { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366 }  
}
```

7.97 BTime.h File Reference

```
#include <BTypes.h>  
#include <BError.h>  
#include <BString.h>
```

Classes

- class [BTime](#)

7.98 BTimer.cpp File Reference

```
#include <BTimer.h>
#include <sys/time.h>
```

7.99 BTimer.h File Reference

```
#include <BMutex.h>
```

Classes

- class [BTimer](#)
Stopwatch style timer.

7.100 BTimeStamp.cpp File Reference

```
#include <BTimeStamp.h>
#include <BTimeStampMs.h>
#include <sys/time.h>
```

Functions

- void [toBString](#) ([BTimeStamp](#) &*v*, [BString](#) &*s*)
- void [fromBString](#) ([BString](#) &*s*, [BTimeStamp](#) &*v*)

Variables

- static int [mon_yday](#) [2][13]

7.100.1 Function Documentation

7.100.1.1 void [fromBString](#) ([BString](#) & *s*, [BTimeStamp](#) & *v*)

7.100.1.2 void [toBString](#) ([BTimeStamp](#) & *v*, [BString](#) & *s*)

7.100.2 Variable Documentation

7.100.2.1 int [mon_yday](#)[2][13] [static]

Initial value:

```
= {
    { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 },
    { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366 }
}
```


7.101 BTimeStamp.h File Reference

```
#include <stdint.h>
#include <BError.h>
```

Classes

- class [BTimeStamp](#)

Functions

- void [toBString](#) ([BTimeStamp](#) &v, [BString](#) &s)
- void [fromBString](#) ([BString](#) &s, [BTimeStamp](#) &v)

7.101.1 Function Documentation

7.101.1.1 void [fromBString](#) ([BString](#) & s, [BTimeStamp](#) & v)

7.101.1.2 void [toBString](#) ([BTimeStamp](#) & v, [BString](#) & s)

7.102 BTimeStampMs.cpp File Reference

```
#include <BTimeStampMs.h>
#include <sys/time.h>
```

Variables

- static int [mon_yday](#) [2][13]

7.102.1 Variable Documentation

7.102.1.1 int [mon_yday](#)[2][13] `[static]`

Initial value:

```
= {
    { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 },
    { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366 }
}
```

7.103 BTimeStampMs.h File Reference

```
#include <stdint.h>
#include <BError.h>
```

Classes

- class [BTimeStampMs](#)

7.104 BTypes.h File Reference

```
#include <stdint.h>
#include <sys/types.h>
#include <vector>
```

Classes

- struct [BObjMember](#)

Typedefs

- typedef bool [Bool](#)
- typedef int8_t [BInt8](#)
- typedef uint8_t [BUInt8](#)
- typedef int16_t [BInt16](#)
- typedef uint16_t [BUInt16](#)
- typedef int32_t [BInt32](#)
- typedef uint32_t [BUInt32](#)
- typedef int64_t [BInt64](#)
- typedef uint64_t [BUInt64](#)
- typedef float [BFloat32](#)
- typedef double [BFloat64](#)
- typedef char [BChar](#)
- typedef [BInt32](#) [BInt](#)
- typedef [BUInt32](#) [BUInt](#)
- typedef [BFloat32](#) [BFloat](#)
- typedef [BFloat64](#) [BDouble](#)
- typedef size_t [BSize](#)
- typedef std::vector< [BFloat32](#) > [BArrayFloat](#)
- typedef std::vector< [BFloat64](#) > [BArrayDouble](#)
- typedef [BUInt32](#) [BTimeout](#)

Enumerations

- enum [BType](#) {
[BTypeNone](#), [BTypeBool](#), [BTypeInt8](#), [BTypeUInt8](#),
[BTypeInt16](#), [BTypeUInt16](#), [BTypeInt32](#), [BTypeUInt32](#),
[BTypeInt64](#), [BTypeUInt64](#), [BTypeFloat32](#), [BTypeFloat64](#),
[BTypeChar](#), [BTypeString](#), [BTypeError](#), [BTypeTime](#),
[BTypeObj](#) = 100 }
- enum [BTypeComp](#) {
[BTypeCompSingle](#), [BTypeCompArray](#), [BTypeCompArrayFixed](#), [BTypeCompList](#),
[BTypeCompDict](#) }

Functions

- [BTimeout timeoutTicks](#) ([BTimeout timeoutUs](#))
- void [byteSwap8](#) (void *d, void *s)
- void [byteSwap16](#) (void *d, void *s)
- void [byteSwap32](#) (void *d, void *s)
- void [byteSwap64](#) (void *d, void *s)

Variables

- const [BTimeout BTimeoutForever](#) = 0xFFFFFFFF

7.104.1 Typedef Documentation

7.104.1.1 typedef std::vector<BFloat64> **BArrayDouble**

7.104.1.2 typedef std::vector<BFloat32> **BArrayFloat**

7.104.1.3 typedef char **BChar**

7.104.1.4 typedef BFloat64 **BDouble**

7.104.1.5 typedef BFloat32 **BFloat**

7.104.1.6 typedef float **BFloat32**

7.104.1.7 typedef double **BFloat64**

7.104.1.8 typedef BInt32 **BInt**

7.104.1.9 typedef int16_t **BInt16**

7.104.1.10 typedef int32_t **BInt32**

7.104.1.11 typedef int64_t **BInt64**

7.104.1.12 typedef int8_t **BInt8**

7.104.1.13 typedef bool **Bool**

7.104.1.14 typedef size_t **BSize**

7.104.1.15 typedef BUInt32 **BTimeout**

7.104.1.16 typedef BUInt32 **BUInt**

7.104.1.17 typedef uint16_t **BUInt16**

7.104.1.18 typedef uint32_t **BUInt32**

7.104.1.19 typedef uint64_t **BUInt64**

7.104.1.20 typedef uint8_t **BUInt8**

7.104.2 Enumeration Type Documentation

7.104.2.1 enum BType

Enumerator

BTypeNone
BTypeBool
BTypeInt8
BTypeUInt8
BTypeInt16
BTypeUInt16
BTypeInt32
BTypeUInt32
BTypeInt64
BTypeUInt64
BTypeFloat32
BTypeFloat64
BTypeChar
BTypeString
BTypeError
BTypeTime
BTypeObj

7.104.2.2 enum BTypeComp

Enumerator

BTypeCompSingle
BTypeCompArray
BTypeCompArrayFixed
BTypeCompList
BTypeCompDict

7.104.3 Function Documentation

- 7.104.3.1 void byteSwap16 (void * *d*, void * *s*) [inline]
- 7.104.3.2 void byteSwap32 (void * *d*, void * *s*) [inline]
- 7.104.3.3 void byteSwap64 (void * *d*, void * *s*) [inline]
- 7.104.3.4 void byteSwap8 (void * *d*, void * *s*) [inline]
- 7.104.3.5 BTimeout timeoutTicks (BTimeout *timeoutUs*) [inline]

7.104.4 Variable Documentation

- 7.104.4.1 const BTimeout BTimeoutForever = 0xFFFFFFFF

7.105 BUrl.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
#include <BUrl.h>
#include <curl/curl.h>
```

7.106 BUrl.h File Reference

```
#include <stdio.h>
#include <BString.h>
#include <BError.h>
```

Classes

- class [BUrl](#)

Basic access to a Url.

Index

- ~BBuffer
 - BBuffer, [21](#)
- ~BBufferStore
 - BBufferStore, [23](#)
- ~BComms
 - BComms, [26](#)
- ~BCond
 - BCond, [27](#)
- ~BCondBool
 - BCondBool, [28](#)
- ~BCondInt
 - BCondInt, [30](#)
- ~BCondResource
 - BCondResource, [32](#)
- ~BCondValue
 - BCondValue, [33](#)
- ~BCondWrap
 - BCondWrap, [35](#)
- ~BDate
 - BDate, [39](#)
- ~BDebugBacktrace
 - BDebugBacktrace, [41](#)
- ~BDir
 - BDir, [47](#)
- ~BDuration
 - BDuration, [49](#)
- ~BEntryFile
 - BEntryFile, [53](#)
- ~BEvent1
 - BEvent1, [62](#)
- ~BEvent1Int
 - BEvent1Int, [64](#)
- ~BEvent1Pipe
 - BEvent1Pipe, [65](#)
- ~BEventPipe
 - BEventPipe, [67](#)
- ~BFifo
 - BFifo, [69](#)
- ~BFifoCirc
 - BFifoCirc, [72](#)
- ~BFile
 - BFile, [77](#)
- ~BList
 - BList, [84](#)
- ~BMutex
 - BMutex, [88](#)
- ~BMutexLock
 - BMutexLock, [89](#)
- ~BMySQL
 - BMySQL, [90](#)
- ~BObj
 - BObj, [120](#)
- ~BPoll
 - BPoll, [122](#)
- ~BQueue
 - BQueue, [124](#)
- ~BRWLock
 - BRWLock, [130](#)
- ~BRefData
 - BRefData, [126](#)
- ~BRtc
 - BRtc, [127](#)
- ~BRtcThreaded
 - BRtcThreaded, [129](#)
- ~BSema
 - BSema, [131](#)
- ~BSemaphore
 - BSemaphore, [133](#)
- ~BSemaphoreCount
 - BSemaphoreCount, [134](#)
- ~BSocket
 - BSocket, [136](#)
- ~BSocketAddress
 - BSocketAddress, [138](#)
- ~BString
 - BString, [145](#)
- ~BTable
 - BTable, [153](#)
- ~BThread
 - BThread, [154](#)
- ~BTimeStamp
 - BTimeStamp, [161](#)
- ~BTimeStampMs
 - BTimeStampMs, [165](#)
- ~BTimer
 - BTimer, [158](#)
- ~BUrl
 - BUrl, [168](#)
- ~BoapClientObject
 - BoapClientObject, [95](#)
- ~BoapMcClientObject
 - BoapMcClientObject, [99](#)
- ~BoapMcComms
 - BoapMcComms, [101](#)
- ~BoapMcServiceObject
 - BoapMcServiceObject, [105](#)
- ~BoapPacket
 - BoapPacket, [109](#)

- ~BoapServer
 - BoapServer, [113](#)
- ~BoapServerConnection
 - BoapServerConnection, [115](#)
- ~BoapServiceObject
 - BoapServiceObject, [118](#)
- __attribute__
 - BoapMc.h, [198](#)
- APIVERSION_TEST
 - Boap.cpp, [194](#)
- accept
 - BSocket, [136](#)
- add
 - BAtomic, [19](#)
 - BAtomicCount, [20](#)
 - BSemaphoreCount, [134](#)
 - BString, [145](#)
 - BTimer, [158](#)
- addEntry
 - Boapns::Boapns, [107](#)
- addMicroSeconds
 - BDuration, [49](#)
 - BTimeStamp, [161](#)
- addMilliSeconds
 - BDuration, [49](#)
 - BTimeStamp, [161](#)
 - BTimeStampMs, [165](#)
- addObject
 - BoapServer, [113](#)
- addRef
 - BRefData, [126](#)
- addRow
 - BTable, [153](#)
- addSeconds
 - BDuration, [49](#)
 - BTime, [156](#)
 - BTimeStamp, [161](#)
 - BTimeStampMs, [165](#)
- address
 - BSocketAddressINET, [140](#)
- addressFrom
 - BoapMc.h, [198](#)
 - BoapMcPacketHead, [105](#)
- addressList
 - Boapns::BoapEntry, [97](#)
- addressTo
 - BoapMc.h, [198](#)
 - BoapMcPacketHead, [105](#)
- apiVersion
 - Boapns, [15](#)
- append
 - BArray, [18](#)
 - BDict, [43](#)
 - BList, [84](#)
 - BPoll, [122](#)
 - BString, [145](#)
- arg
 - BEvent, [61](#)
- average
 - BTimer, [158](#)
- BComms
 - WaitError, [25](#)
 - WaitNone, [25](#)
 - WaitRead, [25](#)
 - WaitWrite, [25](#)
- BError.h
 - ErrorAccessDenied, [187](#)
 - ErrorAppBase, [188](#)
 - ErrorChecksum, [187](#)
 - ErrorComms, [187](#)
 - ErrorConfig, [187](#)
 - ErrorData, [187](#)
 - ErrorDataPresent, [188](#)
 - ErrorEndOfData, [188](#)
 - ErrorEndOfFile, [187](#)
 - ErrorFile, [187](#)
 - ErrorFormat, [187](#)
 - ErrorInit, [187](#)
 - ErrorMisc, [187](#)
 - ErrorNoData, [188](#)
 - ErrorNotAvailable, [187](#)
 - ErrorNotImplemented, [187](#)
 - ErrorOk, [187](#)
 - ErrorOverrun, [187](#)
 - ErrorParam, [187](#)
 - ErrorResourceLimit, [187](#)
 - ErrorTimeout, [187](#)
 - ErrorUnderrun, [187](#)
 - ErrorWarning, [187](#)
- BErrorTime
 - Error, [60](#)
 - None, [60](#)
- BEvent.h
 - BEventTypeNone, [189](#)
- BEvent1.h
 - BEvent1 TypeError, [190](#)
 - BEvent1 TypeInt, [190](#)
 - BEvent1 TypeNone, [190](#)
- BEvent1 TypeError
 - BEvent1.h, [190](#)
- BEvent1 TypeInt
 - BEvent1.h, [190](#)
- BEvent1 TypeNone
 - BEvent1.h, [190](#)
- BEventTypeNone
 - BEvent.h, [189](#)
- BFifoCirc
 - defaultSize, [72](#)
- BMutex
 - Normal, [88](#)
 - Recursive, [88](#)
- BSocket
 - DGRAM, [136](#)
 - PriorityHigh, [136](#)
 - PriorityLow, [136](#)
 - PriorityNormal, [136](#)

- STREAM, [136](#)
- BSpi
 - Mode0, [141](#)
 - Mode1, [141](#)
 - Mode2, [141](#)
 - Mode3, [141](#)
- BTypeBool
 - BTypes.h, [220](#)
- BTypeChar
 - BTypes.h, [220](#)
- BTypeCompArray
 - BTypes.h, [220](#)
- BTypeCompArrayFixed
 - BTypes.h, [220](#)
- BTypeCompDict
 - BTypes.h, [220](#)
- BTypeCompList
 - BTypes.h, [220](#)
- BTypeCompSingle
 - BTypes.h, [220](#)
- BTypeError
 - BTypes.h, [220](#)
- BTypeFloat32
 - BTypes.h, [220](#)
- BTypeFloat64
 - BTypes.h, [220](#)
- BTypeInt16
 - BTypes.h, [220](#)
- BTypeInt32
 - BTypes.h, [220](#)
- BTypeInt64
 - BTypes.h, [220](#)
- BTypeInt8
 - BTypes.h, [220](#)
- BTypeNone
 - BTypes.h, [220](#)
- BTypeObj
 - BTypes.h, [220](#)
- BTypeString
 - BTypes.h, [220](#)
- BTypeTime
 - BTypes.h, [220](#)
- BTypeUInt16
 - BTypes.h, [220](#)
- BTypeUInt32
 - BTypes.h, [220](#)
- BTypeUInt64
 - BTypes.h, [220](#)
- BTypeUInt8
 - BTypes.h, [220](#)
- BTypes.h
 - BTypeBool, [220](#)
 - BTypeChar, [220](#)
 - BTypeCompArray, [220](#)
 - BTypeCompArrayFixed, [220](#)
 - BTypeCompDict, [220](#)
 - BTypeCompList, [220](#)
 - BTypeCompSingle, [220](#)
 - BTypeError, [220](#)
 - BTypeFloat32, [220](#)
 - BTypeFloat64, [220](#)
 - BTypeInt16, [220](#)
 - BTypeInt32, [220](#)
 - BTypeInt64, [220](#)
 - BTypeInt8, [220](#)
 - BTypeNone, [220](#)
 - BTypeObj, [220](#)
 - BTypeString, [220](#)
 - BTypeTime, [220](#)
 - BTypeUInt16, [220](#)
 - BTypeUInt32, [220](#)
 - BTypeUInt64, [220](#)
 - BTypeUInt8, [220](#)
- BArray
 - append, [18](#)
 - BArray, [18](#)
 - BArray, [18](#)
 - del, [18](#)
 - insert, [18](#)
 - number, [18](#)
 - rear, [18](#)
 - sort, [18](#)
 - SortFunc, [18](#)
- BArray< T >, [17](#)
- BArray.h, [171](#)
 - BArrayLoop, [171](#)
- BArrayDouble
 - BTypes.h, [219](#)
- BArrayFloat
 - BTypes.h, [219](#)
- BArrayLoop
 - BArray.h, [171](#)
- BAtomic
 - add, [19](#)
 - BAtomic, [19](#)
 - BAtomic, [19](#)
 - getValue, [19](#)
 - operator Type, [19](#)
 - operator++, [19](#)
 - operator--, [19](#)
 - ovalue, [19](#)
- BAtomic< Type >, [18](#)
- BAtomic.h, [171](#)
 - BAtomicInt32, [172](#)
 - BAtomicInt64, [172](#)
 - BAtomicUInt32, [172](#)
 - BAtomicUInt64, [172](#)
- BAtomicCount, [19](#)
 - add, [20](#)
 - BAtomicCount, [20](#)
 - BAtomicCount, [20](#)
 - getValue, [20](#)
 - operator long, [20](#)
 - operator++, [20](#)
 - operator--, [20](#)
 - ovalue, [20](#)

- BAtomicCount.h, [172](#)
- BAtomicInt32
 - BAtomic.h, [172](#)
- BAtomicInt64
 - BAtomic.h, [172](#)
- BAtomicUInt32
 - BAtomic.h, [172](#)
- BAtomicUInt64
 - BAtomic.h, [172](#)
- BBigEndian
 - BBuffer.h, [173](#)
- BBuffer, [20](#)
 - ~BBuffer, [21](#)
 - BBuffer, [21](#)
 - BBuffer, [21](#)
 - data, [21](#)
 - odata, [22](#)
 - odataSize, [22](#)
 - osize, [22](#)
 - resize, [21](#)
 - setData, [21](#)
 - setSize, [21](#)
 - size, [21](#)
 - writeData, [21](#)
- BBuffer.cpp, [172](#)
 - roundSize, [172](#)
- BBuffer.h, [172](#)
 - BBigEndian, [173](#)
- BBufferStore, [22](#)
 - ~BBufferStore, [23](#)
 - BBufferStore, [23](#)
 - BBufferStore, [23](#)
 - getHexString, [23](#)
 - getPos, [23](#)
 - opos, [24](#)
 - oswapBytes, [24](#)
 - pop, [23](#), [24](#)
 - push, [24](#)
 - setHexString, [24](#)
 - setPos, [24](#)
- BChar
 - BTypes.h, [219](#)
- BComms, [25](#)
 - ~BComms, [26](#)
 - BComms, [26](#)
 - BComms, [26](#)
 - eventQueue, [26](#)
 - init, [26](#)
 - oevent, [26](#)
 - oeventNum, [26](#)
 - oeventQueue, [26](#)
 - opacketMode, [26](#)
 - otimeout, [26](#)
 - packetMode, [26](#)
 - read, [26](#)
 - readAvailable, [26](#)
 - setPacketMode, [26](#)
 - setTimeout, [26](#)
 - Wait, [25](#)
 - wait, [26](#)
 - write, [26](#)
 - writeAvailable, [26](#)
- BComms.cpp, [173](#)
- BComms.h, [173](#)
- BComplex
 - BComplex.h, [173](#)
- BComplex.h, [173](#)
 - BComplex, [173](#)
 - BComplex32, [173](#)
 - BComplex64, [173](#)
- BComplex32
 - BComplex.h, [173](#)
- BComplex64
 - BComplex.h, [173](#)
- BCond, [27](#)
 - ~BCond, [27](#)
 - BCond, [27](#)
 - BCond, [27](#)
 - ocond, [27](#)
 - omutex, [27](#)
 - signal, [27](#)
 - timedWait, [27](#)
 - wait, [27](#)
- BCond.cpp, [174](#)
- BCond.h, [174](#)
- BCondBool, [27](#)
 - ~BCondBool, [28](#)
 - BCondBool, [28](#)
 - BCondBool, [28](#)
 - clear, [28](#)
 - ocond, [29](#)
 - omutex, [29](#)
 - operator int, [28](#)
 - ovalue, [29](#)
 - set, [28](#)
 - timedWait, [28](#)
 - value, [28](#)
 - wait, [28](#)
- BCondInt, [29](#)
 - ~BCondInt, [30](#)
 - BCondInt, [30](#)
 - BCondInt, [30](#)
 - decrement, [30](#)
 - increment, [30](#)
 - ocond, [31](#)
 - omutex, [31](#)
 - operator++, [30](#)
 - operator+==, [30](#)
 - operator--, [30](#)
 - operator-=, [30](#)
 - ovalue, [31](#)
 - setValue, [30](#)
 - value, [30](#)
 - waitLessThan, [30](#)
 - waitLessThanOrEqual, [31](#)
 - waitMoreThanOrEqual, [31](#)

- BCondInt.cpp, 174
 - getTimeout, 174
- BCondInt.h, 174
- BCondResource, 31
 - ~BCondResource, 32
 - BCondResource, 32
 - BCondResource, 32
 - end, 32
 - inUse, 32
 - lock, 32
 - locked, 32
 - ocond, 32
 - oclock, 32
 - omutex, 32
 - ouse, 32
 - start, 32
 - unlock, 32
- BCondValue, 32
 - ~BCondValue, 33
 - BCondValue, 33
 - BCondValue, 33
 - decrement, 33
 - increment, 33
 - ocond, 34
 - omutex, 34
 - operator++, 33
 - operator+==, 34
 - operator--, 34
 - operator-=, 34
 - ovalue, 34
 - setValue, 34
 - value, 34
 - waitLessThan, 34
 - waitLessThanOrEqual, 34
 - waitMoreThanOrEqual, 34
- BCondWrap, 35
 - ~BCondWrap, 35
 - BCondWrap, 35
 - BCondWrap, 35
 - decrement, 35
 - diff, 35
 - increment, 36
 - ocond, 36
 - omutex, 36
 - operator++, 36
 - operator+==, 36
 - operator--, 36
 - operator-=, 36
 - ovalue, 36
 - setValue, 36
 - value, 36
 - waitLessThan, 36
 - waitLessThanOrEqual, 36
 - waitMoreThanOrEqual, 36
- BConfig, 37
 - close, 37
 - fileName, 37
 - findValue, 37
 - ofile, 38
 - ofileName, 38
 - oclock, 38
 - open, 37
 - read, 38
 - write, 38
- BConfig.cpp, 175
- BConfig.h, 175
- BCrc16.cpp, 175
 - brc16, 175
 - table_crc_hi, 175
 - table_crc_lo, 176
- BCrc16.h, 176
 - brc16, 176
- BDate, 38
 - ~BDate, 39
 - BDate, 39
 - BDate, 39
 - clear, 39
 - compare, 39
 - day, 39
 - daysInMonth, 39
 - getDate, 39
 - getString, 39
 - getStringFormatted, 39
 - isLeap, 40
 - isSet, 40
 - month, 40
 - operator BString, 40
 - operator<, 40
 - operator<=, 40
 - operator>, 40
 - operator>=, 40
 - operator==, 40
 - oyday, 41
 - oyear, 41
 - set, 40
 - setFirst, 40
 - setLast, 40
 - setNow, 40
 - setString, 40
 - setYDay, 40
 - yday, 40
 - year, 40
- BDate.cpp, 176
 - fromBString, 177
 - mon_yday, 177
 - toBString, 177
- BDate.h, 177
 - fromBString, 177
 - toBString, 177
- BDebug.cpp, 178
 - BTRACE_SIZE, 178
 - bdebug, 179
 - getTime, 178
 - gettid, 178
 - hd32, 178
 - hd8, 178

- hd8a, [178](#)
- hda32, [178](#)
- hda8, [179](#)
- STRBUF_SIZE, [179](#)
- setDebug, [179](#)
- tprintf, [179](#)
- BDebug.h, [179](#)
 - BDebug_STD, [180](#)
 - bdebug, [180](#)
 - dprintf, [180](#)
 - eprintf, [180](#)
 - getTime, [180](#)
 - gettid, [180](#)
 - hd32, [180](#)
 - hd8, [180](#)
 - hd8a, [180](#)
 - hda8, [180](#)
 - hds32, [180](#)
 - nprintf, [180](#)
 - setDebug, [180](#)
 - tprintf, [180](#)
 - wprintf, [180](#)
- BDebug_STD
 - BDebug.h, [180](#)
- BDebugBacktrace, [41](#)
 - ~BDebugBacktrace, [41](#)
 - BDebugBacktrace, [41](#)
 - BDebugBacktrace, [41](#)
 - dumpBacktrace, [41](#)
 - dumpBacktraceFile, [41](#)
 - dumpBacktraceStdout, [41](#)
 - dumpBacktraceSyslog, [41](#)
- BDict
 - append, [43](#)
 - BDict, [43](#)
 - BDict, [43](#)
 - clear, [43](#)
 - del, [43](#)
 - find, [43](#)
 - hasKey, [43](#)
 - hashAdd, [43](#)
 - hashDelete, [43](#)
 - hashFind, [43](#)
 - hashPrint, [43](#)
 - insert, [43](#)
 - iterator, [43](#)
 - key, [43](#)
 - ohashLists, [44](#)
 - ohashSize, [44](#)
 - operator+, [43](#)
 - operator=, [43](#)
- BDict< Type >, [42](#)
- BDict.cpp, [180](#)
 - bdictStringToString, [181](#)
 - fromBString, [181](#)
 - toBString, [181](#)
- BDict.h, [181](#)
 - BDictString, [181](#)
- bdictStringToString, [181](#)
- fromBString, [181](#)
- toBString, [181](#)
- BDictItem
 - BDictItem, [44](#)
 - BDictItem, [44](#)
 - key, [44](#)
 - value, [44](#)
- BDictItem< Type >, [44](#)
- BDictMap
 - clear, [45](#)
 - del, [45](#)
 - hasKey, [45](#)
 - isEnd, [45](#)
 - iterator, [45](#)
 - key, [45](#)
 - next, [46](#)
 - size, [46](#)
 - start, [46](#)
- BDictMap< Value >, [45](#)
- BDictMap.h, [181](#)
 - BDictMapString, [182](#)
- BDictMapString
 - BDictMap.h, [182](#)
- BDictString
 - BDict.h, [181](#)
- BDir, [46](#)
 - ~BDir, [47](#)
 - BDir, [47](#)
 - BDir, [47](#)
 - clear, [47](#)
 - entryName, [47](#)
 - entryStat, [47](#)
 - entryStat64, [47](#)
 - error, [47](#)
 - odirname, [48](#)
 - oerror, [48](#)
 - open, [47](#)
 - osort, [48](#)
 - owild, [48](#)
 - read, [47](#)
 - setSort, [47](#)
 - setWild, [48](#)
- BDir.cpp, [182](#)
 - wild, [182](#)
 - wildString, [182](#)
- BDir.h, [182](#)
- BDouble
 - BTypes.h, [219](#)
- BDuration, [48](#)
 - ~BDuration, [49](#)
 - addMicroSeconds, [49](#)
 - addMilliSeconds, [49](#)
 - addSeconds, [49](#)
 - BDuration, [49](#)
 - BDuration, [49](#)
 - clear, [49](#)
 - getMicroSeconds, [49](#)

- getSeconds, [49](#)
- getString, [49](#)
- hour, [49](#)
- microSecond, [50](#)
- minute, [50](#)
- ohour, [50](#)
- omicroSecond, [50](#)
- omminute, [50](#)
- osecond, [50](#)
- ospare, [50](#)
- second, [50](#)
- set, [50](#)
- setString, [50](#)
- BDuration.cpp, [182](#)
- BDuration.h, [183](#)
- BEndian.cpp, [183](#)
 - bswap_copy, [183](#)
- BEndian.h, [183](#)
 - be16toh, [184](#)
 - be32toh, [184](#)
 - be64toh, [184](#)
 - betoh, [185](#)
 - bswap_copy, [185](#)
 - bswap_p16, [185](#)
 - bswap_p32, [185](#)
 - bswap_p64, [185](#)
 - bswap_p8, [185](#)
 - htobe, [185](#)
 - htobe16, [184](#)
 - htobe32, [184](#)
 - htobe64, [184](#)
 - htole, [185](#), [186](#)
 - htole16, [184](#)
 - htole32, [184](#)
 - htole64, [184](#)
 - le16toh, [184](#)
 - le32toh, [185](#)
 - le64toh, [185](#)
 - letoh, [186](#)
- BEntry, [50](#)
 - BEntry, [51](#)
 - BEntry, [51](#)
 - getName, [51](#)
 - getValue, [51](#)
 - line, [51](#)
 - oname, [52](#)
 - ovalue, [52](#)
 - print, [52](#)
 - setLine, [52](#)
 - setName, [52](#)
 - setValue, [52](#)
- BEntry.cpp, [186](#)
- BEntry.h, [186](#)
- BEntryFile, [52](#)
 - ~BEntryFile, [53](#)
 - BEntryFile, [53](#)
 - BEntryFile, [53](#)
 - clear, [53](#)
 - filename, [53](#)
 - ocomments, [54](#)
 - ofilename, [54](#)
 - open, [53](#)
 - read, [53](#)
 - write, [54](#)
 - writeList, [54](#)
- BEntryList, [54](#)
 - BEntryList, [55](#)
 - BEntryList, [55](#)
 - clear, [55](#)
 - del, [55](#)
 - deleteEntry, [55](#)
 - find, [55](#)
 - findValue, [55](#)
 - getString, [56](#)
 - insert, [56](#)
 - isSet, [56](#)
 - olastPos, [56](#)
 - operator=, [56](#)
 - print, [56](#)
 - setValue, [56](#)
 - setValueRaw, [56](#)
- BError, [56](#)
 - BError, [57](#)
 - BError, [57](#)
 - clear, [58](#)
 - copy, [58](#)
 - getErrorNo, [58](#)
 - getNumber, [58](#)
 - getString, [58](#)
 - num, [58](#)
 - oerrNo, [59](#)
 - oerrStr, [59](#)
 - operator int, [58](#)
 - set, [58](#)
 - setError, [58](#)
 - str, [58](#)
- BError.cpp, [187](#)
- BError.h, [187](#)
 - BErrorNum, [187](#)
- BErrorNum
 - BError.h, [187](#)
- BErrorTime, [59](#)
 - BErrorTime, [60](#)
 - BErrorTime, [60](#)
 - clear, [60](#)
 - copy, [60](#)
 - getErrorNo, [60](#)
 - getString, [60](#)
 - getTime, [60](#)
 - oerrNo, [60](#)
 - oerrStr, [60](#)
 - oerrTime, [60](#)
 - operator int, [60](#)
 - set, [60](#)
 - Type, [60](#)
- BErrorTime.cpp, [188](#)

BErrorTime.h, [188](#)
 BEvent, [61](#)
 arg, [61](#)
 BEvent, [61](#)
 BEvent, [61](#)
 oarg, [61](#)
 otype, [61](#)
 type, [61](#)
 BEvent.cpp, [188](#)
 BEvent.h, [188](#)
 BEventQueue, [189](#)
 BEventType, [189](#)
 BEvent1, [62](#)
 ~BEvent1, [62](#)
 BEvent1, [62](#)
 BEvent1, [62](#)
 getBinary, [62](#)
 getType, [62](#)
 otype, [63](#)
 setBinary, [62](#)
 BEvent1.cpp, [189](#)
 BEvent1.h, [189](#)
 BEvent1Type, [190](#)
 BEvent1Error, [63](#)
 BEvent1Error, [63](#)
 BEvent1Error, [63](#)
 getBinary, [63](#)
 setBinary, [63](#)
 BEvent1Int, [63](#)
 ~BEvent1Int, [64](#)
 BEvent1Int, [64](#)
 BEvent1Int, [64](#)
 clear, [64](#)
 getEvent, [64](#)
 getFd, [64](#)
 ofds, [64](#)
 sendEvent, [64](#)
 BEvent1Pipe, [65](#)
 ~BEvent1Pipe, [65](#)
 BEvent1Pipe, [65](#)
 BEvent1Pipe, [65](#)
 clear, [65](#)
 getEvent, [65](#)
 getReceiveFd, [65](#)
 ofds, [66](#)
 sendEvent, [66](#)
 BEvent1Type
 BEvent1.h, [190](#)
 BEventPipe, [66](#)
 ~BEventPipe, [67](#)
 BEventPipe, [67](#)
 BEventPipe, [67](#)
 clear, [67](#)
 getFd, [67](#)
 ofds, [67](#)
 read, [67](#)
 readAvailable, [67](#)
 write, [67](#)
 writeAvailable, [67](#)
 BEventQueue
 BEvent.h, [189](#)
 BEventType
 BEvent.h, [189](#)
 BFifo
 ~BFifo, [69](#)
 BFifo, [69](#)
 BFifo, [69](#)
 clear, [69](#)
 odata, [70](#)
 oclock, [70](#)
 oreadPos, [70](#)
 osize, [70](#)
 owritePos, [70](#)
 read, [69](#)
 readAvailable, [69](#)
 readAvailableChunk, [69](#)
 readData, [69](#)
 readDone, [69](#)
 readPos, [69](#)
 resize, [69](#)
 size, [69](#)
 write, [69](#), [70](#)
 writeAvailable, [70](#)
 writeAvailableChunk, [70](#)
 writeBackup, [70](#)
 writeData, [70](#)
 writeDone, [70](#)
 BFifo< Type >, [67](#)
 BFifo.h, [190](#)
 BFifo.inc, [190](#)
 BFifoCirc
 ~BFifoCirc, [72](#)
 BFifoCirc, [72](#)
 BFifoCirc, [72](#)
 clear, [72](#)
 mapCircularBuffer, [72](#)
 odata, [73](#)
 oclock, [73](#)
 oreadPos, [73](#)
 osize, [73](#)
 ovmSize, [74](#)
 owriteNumFifoSamples, [74](#)
 owritePos, [74](#)
 read, [72](#)
 readAvailable, [72](#)
 readData, [72](#)
 readDone, [73](#)
 readWaitAvailable, [73](#)
 size, [73](#)
 unmapCircularBuffer, [73](#)
 write, [73](#)
 writeAvailable, [73](#)
 writeData, [73](#)
 writeDone, [73](#)
 writeWaitAvailable, [73](#)
 BFifoCirc< Type >, [71](#)

- BFifoCirc.cpp, 190
 - dprintf, 190
- BFifoCirc.h, 190
- BFifoCirc.inc, 191
- BFifoCircPos, 74
 - BFifoCircPos, 75
 - BFifoCircPos, 75
 - difference, 75
 - increment, 75
 - operator int, 75
 - operator+=, 75
 - operator==, 75
 - opos, 75
 - osize, 75
 - pos, 75
 - set, 75
 - setSize, 75
- BFile, 75
 - ~BFile, 77
 - BFile, 77
 - BFile, 77
 - close, 77
 - fgets, 77
 - fileName, 77
 - flush, 77
 - getFd, 77
 - isEnd, 77
 - isOpen, 77
 - length, 77
 - ofile, 79
 - ofileName, 79
 - omode, 79
 - open, 77, 78
 - operator=, 78
 - position, 78
 - printf, 78
 - read, 78
 - readString, 78
 - seek, 78
 - setVBuf, 78
 - truncate, 78
 - write, 78
 - writeString, 78
- BFile.cpp, 191
 - STRBUF, 191
- BFile.h, 191
- BFileCsv, 79
 - BFileCsv, 79
 - BFileCsv, 79
 - oseparator, 79
 - readCsv, 79
 - writeCsv, 79
- BFileCsv.cpp, 191
- BFileCsv.h, 192
- BFileData, 80
 - del, 80
 - find, 80
 - getNextId, 80
 - ofilename, 80
 - open, 80
 - read, 80
 - write, 80
- BFileData.cpp, 192
- BFileData.h, 192
- BFloat
 - BTypes.h, 219
- BFloat32
 - BTypes.h, 219
- BFloat64
 - BTypes.h, 219
- BInt
 - BTypes.h, 219
- BInt16
 - BTypes.h, 219
- BInt32
 - BTypes.h, 219
- BInt64
 - BTypes.h, 219
- BInt8
 - BTypes.h, 219
- BIter, 81
 - BIter, 81
 - BIter, 81
 - oi, 81
 - operator BNode *, 81
 - operator==, 81
 - valid, 81
- BList
 - ~BList, 84
 - append, 84
 - BList, 84
 - begin, 84
 - BList, 84
 - clear, 84
 - del, 84
 - deleteFirst, 84
 - deleteLast, 85
 - end, 85
 - front, 85
 - get, 85
 - goTo, 85
 - has, 85
 - insert, 85
 - insertAfter, 85
 - isEnd, 85
 - next, 85
 - nodeCreate, 86
 - nodeGet, 86
 - number, 86
 - olength, 87
 - onodes, 87
 - operator+, 86
 - operator=, 86
 - pop, 86
 - position, 86
 - prev, 86

- push, [86](#)
- queueAdd, [86](#)
- queueGet, [86](#)
- rear, [87](#)
- size, [87](#)
- sort, [87](#)
- SortFunc, [84](#)
- start, [87](#)
- swap, [87](#)
- BList< T >, [81](#)
- BList< T >::Node, [169](#)
- BList.h, [192](#)
 - BListLoop, [193](#)
- BList::Node
 - item, [169](#)
 - Node, [169](#)
- BList_func.h, [193](#)
- BListLoop
 - BList.h, [193](#)
- BMutex, [87](#)
 - ~BMutex, [88](#)
 - BMutex, [88](#)
 - BMutex, [88](#)
 - lock, [88](#)
 - omutex, [89](#)
 - operator=, [88](#)
 - timedLock, [89](#)
 - tryLock, [89](#)
 - Type, [88](#)
 - unlock, [89](#)
- BMutex.cpp, [193](#)
 - MDEBUG, [193](#)
- BMutex.h, [193](#)
- BMutexLock, [89](#)
 - ~BMutexLock, [89](#)
 - BMutexLock, [89](#)
 - BMutexLock, [89](#)
 - lock, [89](#)
 - olock, [90](#)
 - unlock, [89](#)
- BMySQL, [90](#)
 - ~BMySQL, [90](#)
 - BMySQL, [90](#)
 - BMySQL, [90](#)
 - close, [90](#)
 - db, [90](#)
 - del, [90](#)
 - escapeString, [90](#)
 - flush, [91](#)
 - get, [91](#)
 - insert, [91](#)
 - odb, [91](#)
 - odebug, [91](#)
 - olock, [91](#)
 - opened, [91](#)
 - open, [91](#)
 - query, [91](#)
 - setDebug, [91](#)
 - update, [91](#)
- BMySQL.cpp, [193](#)
- BMySQL.h, [193](#)
- BNameValue
 - BNameValue, [92](#)
 - BNameValue, [92](#)
 - getName, [92](#)
 - getValue, [92](#)
 - oname, [92](#)
 - ovalue, [92](#)
- BNameValue< T >, [91](#)
- BNameValue.h, [194](#)
- BNameValueList
 - find, [92](#)
 - findPos, [92](#)
- BNameValueList< T >, [92](#)
- BNode, [93](#)
 - BNode, [93](#)
 - BNode, [93](#)
 - next, [93](#)
 - prev, [93](#)
- BObj, [120](#)
 - ~BObj, [120](#)
 - BObj, [120](#)
 - BObj, [120](#)
 - getDebugString, [120](#)
 - getMember, [120](#)
 - getMembers, [120](#)
 - getType, [120](#)
 - membersPrint, [120](#)
 - setMember, [120](#)
 - setMembers, [121](#)
- BObj.cpp, [201](#)
- BObj.h, [201](#)
- BObjMember, [121](#)
 - dataOffset, [121](#)
 - name, [121](#)
 - size, [121](#)
 - type, [121](#)
 - typeComp, [121](#)
 - typeName, [121](#)
- BObjStringFormat.cpp, [202](#)
 - toBDictStringFromJson, [203](#)
 - toBString, [203](#)
 - toBStringJson, [203](#), [204](#)
- BObjStringFormat.h, [204](#)
 - base64_decode, [205](#)
 - base64_encode, [205](#)
 - toBDictStringFromJson, [205](#)
 - toBString, [205](#)
 - toBStringJson, [205](#), [206](#)
- BPoll, [121](#)
 - ~BPoll, [122](#)
 - append, [122](#)
 - BPoll, [122](#)
 - BPoll, [122](#)
 - clear, [122](#)
 - delFd, [123](#)

- doPoll, 123
- doPollEvents, 123
- getPollFds, 123
- getPollFdsNum, 123
- nextFd, 123
- ofds, 123
- ofdsNext, 123
- ofdsNum, 123
- PollFd, 122
- BPoll.cpp, 206
- BPoll.h, 206
- BQueue
 - ~BQueue, 124
 - BQueue, 124
 - BQueue, 124
 - clear, 124
 - lock, 125
 - onumber, 125
 - osize, 125
 - read, 124
 - readAvailable, 124
 - write, 124
 - writeAvailable, 124
- BQueue< T >, 123
- BQueue.h, 207
 - BQueueInt, 207
- BQueueInt
 - BQueue.h, 207
- BRWLock, 129
 - ~BRWLock, 130
 - BRWLock, 130
 - BRWLock, 130
 - lock, 130
 - operator=, 130
 - rdLock, 130
 - tryRdLock, 130
 - tryWrLock, 130
 - unlock, 130
 - wrLock, 130
- BRWLock.cpp, 208
- BRWLock.h, 208
- BRefData, 125
 - ~BRefData, 126
 - addRef, 126
 - BRefData, 126
 - BRefData, 126
 - copy, 126
 - data, 126
 - deleteRef, 126
 - len, 126
 - odata, 126
 - olen, 126
 - operator=, 126
 - orefCount, 126
 - setLen, 126
- BRefData.cpp, 207
 - CHUNK, 207
- BRefData.h, 207
- BRtc, 127
 - ~BRtc, 127
 - BRtc, 127
 - BRtc, 127
 - init, 127
 - ofd, 128
 - orate, 128
 - wait, 127
- BRtc.cpp, 208
- BRtc.h, 208
- BRtcThreaded, 128
 - ~BRtcThreaded, 129
 - BRtcThreaded, 129
 - BRtcThreaded, 129
 - function, 129
 - init, 129
 - ocond, 129
 - orate, 129
 - ortc, 129
 - wait, 129
- BSema, 131
 - ~BSema, 131
 - BSema, 131
 - BSema, 131
 - getValue, 131
 - operator=, 131
 - osema, 132
 - post, 131
 - timedWait, 131
 - tryWait, 132
 - wait, 132
- BSema.cpp, 208
- BSema.h, 209
- BSemaphore, 132
 - ~BSemaphore, 133
 - BSemaphore, 133
 - BSemaphore, 133
 - getValue, 133
 - operator=, 133
 - osema, 133
 - set, 133
 - wait, 133
- BSemaphore.cpp, 209
- BSemaphore.h, 209
- BSemaphoreCount, 133
 - ~BSemaphoreCount, 134
 - add, 134
 - BSemaphoreCount, 134
 - BSemaphoreCount, 134
 - lock, 134
 - operator=, 134
 - osema, 134
 - ovalue, 134
 - setValue, 134
 - take, 134
 - value, 134
 - wait, 134
- BSize

- BTypes.h, 219
- BSocket, 134
 - ~BSocket, 136
 - accept, 136
 - BSocket, 136
 - bind, 136
 - BSocket, 136
 - close, 136
 - connect, 136
 - getAddress, 136
 - getFd, 136
 - getMTU, 136
 - getSockOpt, 136
 - init, 136
 - listen, 136
 - NType, 136
 - osocket, 137
 - Priority, 136
 - recv, 136
 - recvFrom, 137
 - recvFromWithTimeout, 137
 - recvWithTimeout, 137
 - send, 137
 - sendTo, 137
 - setBroadCast, 137
 - setFd, 137
 - setPriority, 137
 - setReuseAddress, 137
 - setSockOpt, 137
 - shutdown, 137
- BSocket.cpp, 209
 - IP_MTU, 210
- BSocket.h, 210
- BSocketAddress, 137
 - ~BSocketAddress, 138
 - BSocketAddress, 138
 - BSocketAddress, 138
 - len, 138
 - oaddress, 139
 - olen, 139
 - operator const SockAddr *, 138
 - operator=, 138
 - operator==, 138
 - raw, 138
 - set, 138
 - SockAddr, 138
- BSocketAddressINET, 139
 - address, 140
 - getHostName, 140
 - getIpAddressList, 140
 - getIpAddressListAll, 140
 - getIpAddresses, 140
 - getString, 140
 - port, 140
 - set, 140
 - setPort, 140
 - SockAddrIP, 140
- BSpi, 141
 - BSpi, 141
 - init, 141
 - Mode, 141
 - odev, 141
 - odevName, 141
 - transact, 141
- BSpi.cpp, 210
- BSpi.h, 210
- BString, 142
 - ~BString, 145
 - add, 145
 - append, 145
 - BString, 145
 - base64Decode, 145
 - base64Encode, 145
 - basename, 145
 - BString, 145
 - clear, 146
 - compare, 146
 - compareRegex, 146
 - compareWild, 146
 - compareWildExpression, 146
 - convert, 146
 - convertHex, 146
 - copy, 146
 - csvDecode, 147
 - csvEncode, 147
 - del, 147
 - dirname, 147
 - extension, 147
 - field, 147
 - fields, 147
 - find, 147
 - findReverse, 147
 - firstLine, 147
 - fixedLen, 147
 - get, 147
 - getTokenList, 147
 - hash, 148
 - inString, 148
 - init, 148
 - insert, 148
 - isSpace, 148
 - justify, 148
 - len, 148
 - lowerFirst, 148
 - operator const char *, 148
 - operator<, 148
 - operator<=, 148
 - operator>, 149
 - operator>=, 149
 - operator+, 148
 - operator+=, 148
 - operator=, 149
 - operator==, 149
 - ostr, 150
 - pad, 149

- printf, 149
- pullLine, 149
- pullSeparators, 149
- pullToken, 149
- pullWord, 149
- removeNL, 149
- removeSeparators, 149
- retDouble, 149
- retInt, 149
- retStr, 150
- retStrDup, 150
- retUInt, 150
- reverse, 150
- split, 150
- subString, 150
- toLower, 150
- toUpper, 150
- translateChar, 150
- truncate, 150
- BString.cpp, 211
 - barrayToString, 212
 - base64_decode_table, 212
 - blistToString, 212
 - bstringListinList, 212
 - bstringToArray, 212
 - bstringToList, 212
 - charToArray, 212
 - charToList, 212
 - fromBString, 212
 - gmatch, 212
 - MINUS, 211
 - operator<<, 212
 - operator>>, 212
 - STRIP, 211
 - toBString, 212
- BString.h, 213
 - fromBString, 213
 - operator<<, 213
 - operator>>, 213
 - toBString, 213, 214
- BStringLocked, 151
 - BStringLocked, 151
 - BStringLocked, 151
 - len, 151
 - lock, 151
 - operator BString, 151
 - operator+, 151
 - operator=, 151
 - ostr, 151
- BStringLocked.h, 214
- BStringMutex, 152
 - BStringMutex, 152
 - BStringMutex, 152
- BTRACE_SIZE
 - BDebug.cpp, 178
- BTable, 152
 - ~BTable, 153
 - addRow, 153
 - BTable, 153
 - BTable, 153
 - calculateWidths, 153
 - clear, 153
 - ocolumnWidths, 153
 - odata, 153
 - otitle, 153
 - print, 153
 - printLine, 153
 - setTitle, 153
- BTable.cpp, 214
- BTable.h, 214
- BThread, 153
 - ~BThread, 154
 - BThread, 154
 - BThread, 154
 - cancel, 154
 - function, 154
 - getThread, 154
 - opolicy, 154
 - opriority, 155
 - oreult, 155
 - orunning, 155
 - ostackSize, 155
 - othread, 155
 - result, 154
 - running, 154
 - setInitPriority, 154
 - setInitStackSize, 154
 - setPriority, 154
 - start, 154
 - startFunc, 154
 - waitForCompletion, 154
- BThread.cpp, 214
- BThread.h, 215
- BTime, 155
 - addSeconds, 156
 - BTime, 156
 - BTime, 156
 - getDate, 156
 - getSeconds, 156
 - getString, 156
 - getTime, 156
 - isLeapYear, 156
 - isSet, 156
 - operator<, 156
 - operator<=, 156
 - operator>, 156
 - operator>=, 157
 - operator+, 156
 - operator+=, 156
 - operator==, 156
 - otime, 157
 - set, 157
 - setString, 157
 - setYearDay, 157
- BTime.cpp, 215
 - monDays, 215

- yearDays, [215](#)
- yearIsLeap, [215](#)
- BTime.h, [215](#)
- BTimeStamp, [159](#)
 - ~BTimeStamp, [161](#)
 - addMicroSeconds, [161](#)
 - addMilliSeconds, [161](#)
 - addSeconds, [161](#)
 - BTimeStamp, [161](#)
 - BTimeStamp, [161](#)
 - clear, [161](#)
 - compare, [161](#)
 - day, [161](#)
 - difference, [162](#)
 - getDate, [162](#)
 - getString, [162](#)
 - getStringFormatted, [162](#)
 - getStringNoMs, [162](#)
 - getYearMicroSeconds, [162](#)
 - getYearSeconds, [162](#)
 - hour, [162](#)
 - isLeap, [162](#)
 - isSet, [162](#)
 - microSecond, [162](#)
 - minute, [162](#)
 - month, [162](#)
 - ohour, [163](#)
 - omicroSecond, [163](#)
 - omminute, [163](#)
 - operator BString, [162](#)
 - operator<, [162](#)
 - operator<=, [162](#)
 - operator>, [162](#)
 - operator>=, [162](#)
 - operator=, [162](#)
 - operator==, [162](#)
 - osecond, [163](#)
 - ospare, [164](#)
 - oyday, [164](#)
 - oyear, [164](#)
 - second, [162](#)
 - set, [163](#)
 - setFirst, [163](#)
 - setLast, [163](#)
 - setNow, [163](#)
 - setString, [163](#)
 - setTime, [163](#)
 - setYDay, [163](#)
 - yday, [163](#)
 - year, [163](#)
- BTimeStamp.cpp, [216](#)
 - fromBString, [216](#)
 - mon_yday, [216](#)
 - toBString, [216](#)
- BTimeStamp.h, [217](#)
 - fromBString, [217](#)
 - toBString, [217](#)
- BTimeStampMs, [164](#)
 - ~BTimeStampMs, [165](#)
 - addMilliSeconds, [165](#)
 - addSeconds, [165](#)
 - BTimeStampMs, [165](#)
 - BTimeStampMs, [165](#)
 - clear, [166](#)
 - compare, [166](#)
 - difference, [166](#)
 - getDate, [166](#)
 - getDurationString, [166](#)
 - getDurationStringNoMs, [166](#)
 - getString, [166](#)
 - getStringNoMs, [166](#)
 - getStringRaw, [166](#)
 - getYearMilliSeconds, [166](#)
 - getYearSeconds, [166](#)
 - hour, [167](#)
 - isLeap, [166](#)
 - milliSecond, [167](#)
 - minute, [167](#)
 - operator<, [166](#)
 - operator<=, [166](#)
 - operator>, [166](#)
 - operator>=, [167](#)
 - sampleNumber, [167](#)
 - second, [167](#)
 - setDurationString, [167](#)
 - setNow, [167](#)
 - setString, [167](#)
 - subMilliSeconds, [167](#)
 - subSeconds, [167](#)
 - yday, [167](#)
 - year, [167](#)
- BTimeStampMs.cpp, [217](#)
 - mon_yday, [217](#)
- BTimeStampMs.h, [217](#)
- BTimeout
 - BTypes.h, [219](#)
- BTimeoutForever
 - BTypes.h, [220](#)
- BTimer, [157](#)
 - ~BTimer, [158](#)
 - add, [158](#)
 - average, [158](#)
 - BTimer, [158](#)
 - BTimer, [158](#)
 - clear, [158](#)
 - getElapsedTime, [158](#)
 - getTime, [158](#)
 - oaverage, [159](#)
 - oendTime, [159](#)
 - oclock, [159](#)
 - onum, [159](#)
 - opeak, [159](#)
 - ostartTime, [159](#)
 - peak, [158](#)
 - start, [158](#)
 - stop, [159](#)

- BTimer.cpp, 216
- BTimer.h, 216
- BType
 - BTypes.h, 220
- BTypeComp
 - BTypes.h, 220
- BTypes.h, 218
 - BArrayDouble, 219
 - BArrayFloat, 219
 - BChar, 219
 - BDouble, 219
 - BFloat, 219
 - BFloat32, 219
 - BFloat64, 219
 - BInt, 219
 - BInt16, 219
 - BInt32, 219
 - BInt64, 219
 - BInt8, 219
 - BSize, 219
 - BTimeout, 219
 - BTimeoutForever, 220
 - BType, 220
 - BTypeComp, 220
 - BUInt, 219
 - BUInt16, 219
 - BUInt32, 219
 - BUInt64, 219
 - BUInt8, 219
 - Bool, 219
 - byteSwap16, 220
 - byteSwap32, 220
 - byteSwap64, 220
 - byteSwap8, 220
 - timeoutTicks, 220
- BUInt
 - BTypes.h, 219
- BUInt16
 - BTypes.h, 219
- BUInt32
 - BTypes.h, 219
- BUInt64
 - BTypes.h, 219
- BUInt8
 - BTypes.h, 219
- BUrl, 168
 - ~BUrl, 168
 - BUrl, 168
 - BUrl, 168
 - oinit, 169
 - ores, 169
 - readString, 168
 - writeData, 168
- BUrl.cpp, 221
- BUrl.h, 221
- barrayToString
 - BString.cpp, 212
- base64_decode
 - BObjStringFormat.h, 205
- base64_decode_table
 - BString.cpp, 212
- base64_encode
 - BObjStringFormat.h, 205
- base64Decode
 - BString, 145
- base64Encode
 - BString, 145
- basename
 - BString, 145
- bcrc16
 - BCrc16.cpp, 175
 - BCrc16.h, 176
- bdebug
 - BDebug.cpp, 179
 - BDebug.h, 180
- bdictStringToString
 - BDict.cpp, 181
 - BDict.h, 181
- be16toh
 - BEndian.h, 184
- be32toh
 - BEndian.h, 184
- be64toh
 - BEndian.h, 184
- begin
 - BList, 84
- betoh
 - BEndian.h, 185
- bind
 - BSocket, 136
- blistToString
 - BString.cpp, 212
- Boap.h
 - BoapPriorityHigh, 196
 - BoapPriorityLow, 196
 - BoapPriorityNormal, 196
 - BoapTypeRpc, 196
 - BoapTypeRpcError, 196
 - BoapTypeRpcReply, 196
 - BoapTypeSignal, 196
- Boap.cpp, 194
 - APIVERSION_TEST, 194
 - boapPort, 195
 - DEBUG, 194
 - dprintf, 194
 - IS_BIG_ENDIAN, 195
- Boap.h, 195
 - BoapFunc, 196
 - BoapMagic, 196
 - BoapPriority, 196
 - BoapService, 196
 - BoapType, 196
- BoapMc.h
 - BoapMcTypeReply, 197
 - BoapMcTypeRequest, 197
- BoapMcTypeReply

- BoapMc.h, [197](#)
- BoapMcTypeRequest
 - BoapMc.h, [197](#)
- BoapPriorityHigh
 - Boap.h, [196](#)
- BoapPriorityLow
 - Boap.h, [196](#)
- BoapPriorityNormal
 - Boap.h, [196](#)
- BoapServer
 - NOTHEADS, [113](#)
 - THREADED, [113](#)
- BoapSimple.h
 - BoapTypeRpc, [201](#)
 - BoapTypeRpcError, [201](#)
 - BoapTypeRpcReply, [201](#)
 - BoapTypeSignal, [201](#)
- BoapTypeRpc
 - Boap.h, [196](#)
 - BoapSimple.h, [201](#)
- BoapTypeRpcError
 - Boap.h, [196](#)
 - BoapSimple.h, [201](#)
- BoapTypeRpcReply
 - Boap.h, [196](#)
 - BoapSimple.h, [201](#)
- BoapTypeSignal
 - Boap.h, [196](#)
 - BoapSimple.h, [201](#)
- BoapClientObject, [93](#)
 - ~BoapClientObject, [95](#)
 - BoapClientObject, [95](#)
 - BoapClientObject, [95](#)
 - checkApiVersion, [95](#)
 - connectService, [95](#)
 - disconnectService, [95](#)
 - getServiceName, [95](#)
 - handleReconnect, [95](#)
 - oapiVersion, [96](#)
 - oconnected, [96](#)
 - oclock, [96](#)
 - omaxLength, [96](#)
 - oname, [96](#)
 - opriority, [96](#)
 - oreconnect, [96](#)
 - orx, [96](#)
 - oservice, [96](#)
 - otimeout, [96](#)
 - otx, [96](#)
 - performCall, [95](#)
 - performRecv, [95](#)
 - performSend, [95](#)
 - ping, [95](#)
 - pingLocked, [96](#)
 - setConnectionPriority, [96](#)
 - setMaxLength, [96](#)
 - setTimeout, [96](#)
- BoapEntry
 - Boapns::BoapEntry, [97](#)
- BoapFunc
 - Boap.h, [196](#)
 - BoapSimple.h, [201](#)
- BoapFuncEntry, [97](#)
 - BoapFuncEntry, [98](#)
 - BoapFuncEntry, [98](#)
 - ocmd, [98](#)
 - ofunc, [98](#)
- BoapMagic
 - Boap.h, [196](#)
- BoapMc.cpp, [196](#)
 - DEBUG_LOCAL, [197](#)
 - DEBUG_LOCAL1, [197](#)
 - dl1printf, [197](#)
 - dlprintf, [197](#)
- BoapMc.h, [197](#)
 - __attribute__, [198](#)
 - addressFrom, [198](#)
 - addressTo, [198](#)
 - BoapMcType, [197](#)
 - checksum, [198](#)
 - cmd, [198](#)
 - error, [198](#)
 - length, [198](#)
- BoapMcClientObject, [98](#)
 - ~BoapMcClientObject, [99](#)
 - BoapMcClientObject, [99](#)
 - BoapMcClientObject, [99](#)
 - getApiVersion, [99](#)
 - oaddressFrom, [99](#)
 - oaddressTo, [99](#)
 - oapiVersion, [99](#)
 - ocomms, [99](#)
 - opacket, [99](#)
 - performCall, [99](#)
 - performRecv, [99](#)
 - performSend, [99](#)
 - setAddress, [99](#)
- BoapMcComms, [100](#)
 - ~BoapMcComms, [101](#)
 - BoapMcComms, [101](#)
 - BoapMcComms, [101](#)
 - getApiVersion, [101](#)
 - oaddressFrom, [102](#)
 - oaddressTo, [102](#)
 - oapiVersion, [102](#)
 - ocomms, [102](#)
 - oclockCall, [103](#)
 - oclockTx, [103](#)
 - opacket, [103](#)
 - opacketReqQueue, [103](#)
 - opacketReqRx, [103](#)
 - opacketReqTx, [103](#)
 - opacketRx, [103](#)
 - opacketRxSema, [103](#)
 - opacketTx, [103](#)
 - opacketTxQueue, [103](#)

- opacketTxQueueWriteNum, 103
- opacketTxSema, 103
- oslave, 103
- othreaded, 104
- otimeout, 104
- packetRecv, 101
- packetSend, 101
- performCall, 101
- performSend, 101
- processPacket, 102
- processRequest, 102
- processRequests, 102
- processRx, 102
- setAddress, 102
- setComms, 102
- setCommsMode, 102
- setTimeout, 102
- BoapMcPacket, 104
 - data, 104
 - head, 104
- BoapMcPacketHead, 104
 - addressFrom, 105
 - addressTo, 105
 - checksum, 105
 - cmd, 105
 - error, 105
 - length, 105
- BoapMcServiceObject, 105
 - ~BoapMcServiceObject, 105
 - BoapMcServiceObject, 105
 - BoapMcServiceObject, 105
 - oapiVersion, 106
 - process, 105
 - processEvent, 105
 - sendEvent, 105
- BoapMcSignalObject, 106
 - BoapMcSignalObject, 106
 - BoapMcSignalObject, 106
 - ocomms, 106
 - performSend, 106
- BoapMcType
 - BoapMc.h, 197
- BoapPacket, 107
 - ~BoapPacket, 109
 - BoapPacket, 109
 - BoapPacket, 109
 - data, 109
 - getCmd, 109
 - nbytes, 109
 - odata, 110
 - onbytes, 110
 - opos, 110
 - osize, 110
 - peekHead, 109
 - pop, 109
 - popHead, 109
 - push, 109, 110
 - pushHead, 110
 - resize, 110
 - setData, 110
 - updateHead, 110
 - updateLen, 110
- BoapPacketHead, 110
 - cmd, 111
 - length, 111
 - reserved, 111
 - service, 111
 - type, 111
- boapPort
 - Boap.cpp, 195
- BoapPriority
 - Boap.h, 196
- BoapServer, 111
 - ~BoapServer, 113
 - addObject, 113
 - BoapServer, 113
 - BoapServer, 113
 - clientGone, 113
 - function, 113
 - getConnectionsNumber, 113
 - getEventSocket, 113
 - getHostName, 113
 - getSocket, 113
 - init, 113
 - newConnection, 113
 - oboapNs, 114
 - oboapns, 114
 - oclientGoneEvent, 114
 - oclients, 114
 - ohostName, 114
 - oisBoapns, 114
 - onet, 114
 - onetEvent, 114
 - onetEventAddress, 114
 - onumOperations, 114
 - opoll, 114
 - orx, 114
 - oservices, 114
 - othreaded, 114
 - otx, 114
 - process, 113
 - processEvent, 113, 114
 - run, 114
 - sendEvent, 114
- BoapServerConnection, 115
 - ~BoapServerConnection, 115
 - BoapServerConnection, 115
 - BoapServerConnection, 115
 - function, 115
 - getHead, 115
 - getSocket, 116
 - init, 116
 - oboapServer, 116
 - omaxLength, 116
 - orx, 116
 - osocket, 116

- otx, [116](#)
 - process, [116](#)
 - setMaxLength, [116](#)
 - validate, [116](#)
- BoapService
 - Boap.h, [196](#)
 - BoapSimple.h, [201](#)
- BoapServiceEntry, [116](#)
 - BoapServiceEntry, [116](#)
 - BoapServiceEntry, [116](#)
- oobject, [117](#)
- oservice, [117](#)
- BoapServiceObject, [117](#)
 - ~BoapServiceObject, [118](#)
 - BoapServiceObject, [118](#)
 - BoapServiceObject, [118](#)
 - doConnectionPriority, [118](#)
 - doPing, [118](#)
 - name, [118](#)
 - oapiVersion, [118](#)
 - ofuncList, [118](#)
 - oname, [118](#)
 - oserver, [118](#)
 - process, [118](#)
 - processEvent, [118](#)
 - sendEvent, [118](#)
 - setName, [118](#)
- BoapSignalObject, [119](#)
 - BoapSignalObject, [119](#)
 - BoapSignalObject, [119](#)
- orx, [119](#)
- otx, [119](#)
- performSend, [119](#)
- BoapSimple.cc, [199](#)
 - DEBUG, [200](#)
 - dprintf, [200](#)
 - roundSize, [200](#)
- BoapSimple.h, [200](#)
 - BoapFunc, [201](#)
 - BoapService, [201](#)
 - BoapType, [201](#)
 - Double, [201](#)
 - Int16, [201](#)
 - Int32, [201](#)
 - Int8, [201](#)
 - UInt16, [201](#)
 - UInt32, [201](#)
 - UInt8, [201](#)
- BoapType
 - Boap.h, [196](#)
 - BoapSimple.h, [201](#)
- Boapns, [15](#)
 - apiVersion, [15](#)
 - Boapns::Boapns, [107](#)
- Boapns::BoapEntry, [97](#)
 - addressList, [97](#)
 - BoapEntry, [97](#)
 - hostName, [97](#)
 - name, [97](#)
 - port, [97](#)
 - service, [97](#)
- Boapns::Boapns, [106](#)
 - addEntry, [107](#)
 - Boapns, [107](#)
 - delEntry, [107](#)
 - getEntry, [107](#)
 - getEntryList, [107](#)
 - getNewName, [107](#)
 - getVersion, [107](#)
- BoapnsC.cpp, [198](#)
- BoapnsC.h, [198](#)
- BoapnsD.cpp, [199](#)
- BoapnsD.h, [199](#)
- Bool
 - BTypes.h, [219](#)
- bstringListinList
 - BString.cpp, [212](#)
- bstringToArray
 - BString.cpp, [212](#)
- bstringToList
 - BString.cpp, [212](#)
- bswap_copy
 - BEndian.cpp, [183](#)
 - BEndian.h, [185](#)
- bswap_p16
 - BEndian.h, [185](#)
- bswap_p32
 - BEndian.h, [185](#)
- bswap_p64
 - BEndian.h, [185](#)
- bswap_p8
 - BEndian.h, [185](#)
- byteSwap16
 - BTypes.h, [220](#)
- byteSwap32
 - BTypes.h, [220](#)
- byteSwap64
 - BTypes.h, [220](#)
- byteSwap8
 - BTypes.h, [220](#)
- CHUNK
 - BRefData.cpp, [207](#)
- calculateWidths
 - BTable, [153](#)
- cancel
 - BThread, [154](#)
- charToArray
 - BString.cpp, [212](#)
- charToList
 - BString.cpp, [212](#)
- checkApiVersion
 - BoapClientObject, [95](#)
- checksum
 - BoapMc.h, [198](#)
 - BoapMcPacketHead, [105](#)
- clear

- BCondBool, 28
- BDate, 39
- BDict, 43
- BDictMap, 45
- BDir, 47
- BDuration, 49
- BEntryFile, 53
- BEntryList, 55
- BError, 58
- BErrorTime, 60
- BEvent1Int, 64
- BEvent1Pipe, 65
- BEventPipe, 67
- BFifo, 69
- BFifoCirc, 72
- BList, 84
- BPoll, 122
- BQueue, 124
- BString, 146
- BTable, 153
- BTimer, 158
- BTimeStamp, 161
- BTimeStampMs, 166
- clientGone
 - BoapServer, 113
- close
 - BConfig, 37
 - BFile, 77
 - BMysql, 90
 - BSocket, 136
- cmd
 - BoapMc.h, 198
 - BoapMcPacketHead, 105
 - BoapPacketHead, 111
- compare
 - BDate, 39
 - BString, 146
 - BTimeStamp, 161
 - BTimeStampMs, 166
- compareRegex
 - BString, 146
- compareWild
 - BString, 146
- compareWildExpression
 - BString, 146
- connect
 - BSocket, 136
- connectService
 - BoapClientObject, 95
- convert
 - BString, 146
- convertHex
 - BString, 146
- copy
 - BError, 58
 - BErrorTime, 60
 - BRefData, 126
 - BString, 146
- csvDecode
 - BString, 147
- csvEncode
 - BString, 147
- DGRAM
 - BSocket, 136
- DEBUG
 - Boap.cpp, 194
 - BoapSimple.cc, 200
- DEBUG_LOCAL
 - BoapMc.cpp, 197
- DEBUG_LOCAL1
 - BoapMc.cpp, 197
- data
 - BBuffer, 21
 - BoapMcPacket, 104
 - BoapPacket, 109
 - BRefData, 126
- dataOffset
 - BObjMember, 121
- day
 - BDate, 39
 - BTimeStamp, 161
- daysInMonth
 - BDate, 39
- db
 - BMysql, 90
- decrement
 - BCondInt, 30
 - BCondValue, 33
 - BCondWrap, 35
- defaultSize
 - BFifoCirc, 72
- del
 - BArray, 18
 - BDict, 43
 - BDictMap, 45
 - BEntryList, 55
 - BFileData, 80
 - BList, 84
 - BMysql, 90
 - BString, 147
- delEntry
 - Boapns::Boapns, 107
- delFd
 - BPoll, 123
- deleteEntry
 - BEntryList, 55
- deleteFirst
 - BList, 84
- deleteLast
 - BList, 85
- deleteRef
 - BRefData, 126
- diff
 - BCondWrap, 35
- difference
 - BFifoCircPos, 75

- BTimeStamp, [162](#)
- BTimeStampMs, [166](#)
- dirname
 - BString, [147](#)
- disconnectService
 - BoapClientObject, [95](#)
- dl1printf
 - BoapMc.cpp, [197](#)
- dlprintf
 - BoapMc.cpp, [197](#)
- doConnectionPriority
 - BoapServiceObject, [118](#)
- doPing
 - BoapServiceObject, [118](#)
- doPoll
 - BPoll, [123](#)
- doPollEvents
 - BPoll, [123](#)
- Double
 - BoapSimple.h, [201](#)
- dprintf
 - BDebug.h, [180](#)
 - BFifoCirc.cpp, [190](#)
 - Boap.cpp, [194](#)
 - BoapSimple.cc, [200](#)
- dumpBacktrace
 - BDebugBacktrace, [41](#)
- dumpBacktraceFile
 - BDebugBacktrace, [41](#)
- dumpBacktraceStdout
 - BDebugBacktrace, [41](#)
- dumpBacktraceSyslog
 - BDebugBacktrace, [41](#)
- end
 - BCondResource, [32](#)
 - BList, [85](#)
- entryName
 - BDir, [47](#)
- entryStat
 - BDir, [47](#)
- entryStat64
 - BDir, [47](#)
- eprintf
 - BDebug.h, [180](#)
- Error
 - BErrorTime, [60](#)
- error
 - BDir, [47](#)
 - BoapMc.h, [198](#)
 - BoapMcPacketHead, [105](#)
- ErrorAccessDenied
 - BError.h, [187](#)
- ErrorAppBase
 - BError.h, [188](#)
- ErrorChecksum
 - BError.h, [187](#)
- ErrorComms
 - BError.h, [187](#)
- ErrorConfig
 - BError.h, [187](#)
- ErrorData
 - BError.h, [187](#)
- ErrorDataPresent
 - BError.h, [188](#)
- ErrorEndOfData
 - BError.h, [188](#)
- ErrorEndOfFile
 - BError.h, [187](#)
- ErrorFile
 - BError.h, [187](#)
- ErrorFormat
 - BError.h, [187](#)
- ErrorInit
 - BError.h, [187](#)
- ErrorMisc
 - BError.h, [187](#)
- ErrorNoData
 - BError.h, [188](#)
- ErrorNotAvailable
 - BError.h, [187](#)
- ErrorNotImplemented
 - BError.h, [187](#)
- ErrorOk
 - BError.h, [187](#)
- ErrorOverrun
 - BError.h, [187](#)
- ErrorParam
 - BError.h, [187](#)
- ErrorResourceLimit
 - BError.h, [187](#)
- ErrorTimeout
 - BError.h, [187](#)
- ErrorUnderrun
 - BError.h, [187](#)
- ErrorWarning
 - BError.h, [187](#)
- escapeString
 - BMySQL, [90](#)
- eventQueue
 - BComms, [26](#)
- extension
 - BString, [147](#)
- fgets
 - BFile, [77](#)
- field
 - BString, [147](#)
- fields
 - BString, [147](#)
- fileName
 - BConfig, [37](#)
 - BFile, [77](#)
- filename
 - BEntryFile, [53](#)
- find
 - BDict, [43](#)
 - BEntryList, [55](#)

- BFileData, 80
- BNameValueList, 92
- BString, 147
- findPos
 - BNameValueList, 92
- findReverse
 - BString, 147
- findValue
 - BConfig, 37
 - BEntryList, 55
- firstLine
 - BString, 147
- fixedLen
 - BString, 147
- flush
 - BFile, 77
 - BMySQL, 91
- fromBString
 - BDate.cpp, 177
 - BDate.h, 177
 - BDict.cpp, 181
 - BDict.h, 181
 - BString.cpp, 212
 - BString.h, 213
 - BTimeStamp.cpp, 216
 - BTimeStamp.h, 217
- front
 - BList, 85
- function
 - BoapServer, 113
 - BoapServerConnection, 115
 - BRtcThreaded, 129
 - BThread, 154
- get
 - BList, 85
 - BMySQL, 91
 - BString, 147
- getAddress
 - BSocket, 136
- getApiVersion
 - BoapMcClientObject, 99
 - BoapMcComms, 101
- getBinary
 - BEvent1, 62
 - BEvent1Error, 63
- getCmd
 - BoapPacket, 109
- getConnectionsNumber
 - BoapServer, 113
- getDate
 - BDate, 39
 - BTime, 156
 - BTimeStamp, 162
 - BTimeStampMs, 166
- getDebugString
 - BObj, 120
- getDurationString
 - BTimeStampMs, 166
- getElapsedTime
 - BTimer, 158
- getEntry
 - Boapns::Boapns, 107
- getEntryList
 - Boapns::Boapns, 107
- getErrorNo
 - BError, 58
 - BErrorTime, 60
- getEvent
 - BEvent1Int, 64
 - BEvent1Pipe, 65
- getEventSocket
 - BoapServer, 113
- getFd
 - BEvent1Int, 64
 - BEventPipe, 67
 - BFile, 77
 - BSocket, 136
- getHead
 - BoapServerConnection, 115
- getHexString
 - BBufferStore, 23
- getHostName
 - BoapServer, 113
 - BSocketAddressINET, 140
- getIpAddressList
 - BSocketAddressINET, 140
- getIpAddressListAll
 - BSocketAddressINET, 140
- getIpAddresses
 - BSocketAddressINET, 140
- getMTU
 - BSocket, 136
- getMember
 - BObj, 120
- getMembers
 - BObj, 120
- getMicroSeconds
 - BDuration, 49
- getName
 - BEntry, 51
 - BNameValue, 92
- getNewName
 - Boapns::Boapns, 107
- getNextId
 - BFileData, 80
- getNumber
 - BError, 58
- getPollFds
 - BPoll, 123
- getPollFdsNum
 - BPoll, 123
- getPos
 - BBufferStore, 23
- getReceiveFd

- BEvent1Pipe, 65
- getSeconds
 - BDuration, 49
 - BTime, 156
- getServiceName
 - BoapClientObject, 95
- getSockOpt
 - BSocket, 136
- getSocket
 - BoapServer, 113
 - BoapServerConnection, 116
- getString
 - BDate, 39
 - BDuration, 49
 - BEntryList, 56
 - BError, 58
 - BErrorTime, 60
 - BSocketAddressINET, 140
 - BTime, 156
 - BTimeStamp, 162
 - BTimeStampMs, 166
- getStringFormatted
 - BDate, 39
 - BTimeStamp, 162
- getStringNoMs
 - BTimeStamp, 162
 - BTimeStampMs, 166
- getStringRaw
 - BTimeStampMs, 166
- getThread
 - BThread, 154
- getTime
 - BDebug.cpp, 178
 - BDebug.h, 180
 - BErrorTime, 60
 - BTime, 156
 - BTimer, 158
- getTimeout
 - BCondInt.cpp, 174
- getTokenList
 - BString, 147
- getType
 - BEvent1, 62
 - BObj, 120
- getValue
 - BAtomic, 19
 - BAtomicCount, 20
 - BEntry, 51
 - BNameValue, 92
 - BSema, 131
 - BSemaphore, 133
- getVersion
 - Boapns::Boapns, 107
- getYearMicroSeconds
 - BTimeStamp, 162
- getYearMilliSeconds
 - BTimeStampMs, 166
- getYearSeconds
 - BTimeStamp, 162
 - BTimeStampMs, 166
- gettid
 - BDebug.cpp, 178
 - BDebug.h, 180
- gmatch
 - BString.cpp, 212
- goTo
 - BList, 85
- handleReconnect
 - BoapClientObject, 95
- has
 - BList, 85
- hasKey
 - BDict, 43
 - BDictMap, 45
- hash
 - BString, 148
- hashAdd
 - BDict, 43
- hashDelete
 - BDict, 43
- hashFind
 - BDict, 43
- hashPrint
 - BDict, 43
- hd32
 - BDebug.cpp, 178
 - BDebug.h, 180
- hd8
 - BDebug.cpp, 178
 - BDebug.h, 180
- hd8a
 - BDebug.cpp, 178
 - BDebug.h, 180
- hda32
 - BDebug.cpp, 178
- hda8
 - BDebug.cpp, 179
 - BDebug.h, 180
- hds32
 - BDebug.h, 180
- head
 - BoapMcPacket, 104
- hostName
 - Boapns::BoapEntry, 97
- hour
 - BDuration, 49
 - BTimeStamp, 162
 - BTimeStampMs, 167
- htobe
 - BEndian.h, 185
- htobe16
 - BEndian.h, 184
- htobe32
 - BEndian.h, 184
- htobe64
 - BEndian.h, 184

- htole
 - BEndian.h, [185](#), [186](#)
- htole16
 - BEndian.h, [184](#)
- htole32
 - BEndian.h, [184](#)
- htole64
 - BEndian.h, [184](#)
- IP_MTU
 - BSocket.cpp, [210](#)
- IS_BIG_ENDIAN
 - Boap.cpp, [195](#)
- inString
 - BString, [148](#)
- inUse
 - BCondResource, [32](#)
- increment
 - BCondInt, [30](#)
 - BCondValue, [33](#)
 - BCondWrap, [36](#)
 - BFifoCircPos, [75](#)
- init
 - BComms, [26](#)
 - BoapServer, [113](#)
 - BoapServerConnection, [116](#)
 - BRtc, [127](#)
 - BRtcThreaded, [129](#)
 - BSocket, [136](#)
 - BSpi, [141](#)
 - BString, [148](#)
- insert
 - BArray, [18](#)
 - BDict, [43](#)
 - BEntryList, [56](#)
 - BList, [85](#)
 - BMysql, [91](#)
 - BString, [148](#)
- insertAfter
 - BList, [85](#)
- Int16
 - BoapSimple.h, [201](#)
- Int32
 - BoapSimple.h, [201](#)
- Int8
 - BoapSimple.h, [201](#)
- isEnd
 - BDictMap, [45](#)
 - BFile, [77](#)
 - BList, [85](#)
- isLeap
 - BDate, [40](#)
 - BTimeStamp, [162](#)
 - BTimeStampMs, [166](#)
- isLeapYear
 - BTime, [156](#)
- isOpen
 - BFile, [77](#)
- isSet
 - BDate, [40](#)
 - BEntryList, [56](#)
 - BTime, [156](#)
 - BTimeStamp, [162](#)
- isSpace
 - BString, [148](#)
- item
 - BList::Node, [169](#)
- iterator
 - BDict, [43](#)
 - BDictMap, [45](#)
- justify
 - BString, [148](#)
- key
 - BDict, [43](#)
 - BDictItem, [44](#)
 - BDictMap, [45](#)
- le16toh
 - BEndian.h, [184](#)
- le32toh
 - BEndian.h, [185](#)
- le64toh
 - BEndian.h, [185](#)
- len
 - BRefData, [126](#)
 - BSocketAddress, [138](#)
 - BString, [148](#)
 - BStringLocked, [151](#)
- length
 - BFile, [77](#)
 - BoapMc.h, [198](#)
 - BoapMcPacketHead, [105](#)
 - BoapPacketHead, [111](#)
- letoh
 - BEndian.h, [186](#)
- line
 - BEntry, [51](#)
- listen
 - BSocket, [136](#)
- lock
 - BCondResource, [32](#)
 - BMutex, [88](#)
 - BMutexLock, [89](#)
- locked
 - BCondResource, [32](#)
- lowerFirst
 - BString, [148](#)
- MDEBUG
 - BMutex.cpp, [193](#)
- MINUS
 - BString.cpp, [211](#)
- mapCircularBuffer
 - BFifoCirc, [72](#)
- membersPrint
 - BObj, [120](#)

- microSecond
 - BDuration, [50](#)
 - BTimeStamp, [162](#)
- milliSecond
 - BTimeStampMs, [167](#)
- minute
 - BDuration, [50](#)
 - BTimeStamp, [162](#)
 - BTimeStampMs, [167](#)
- Mode
 - BSpi, [141](#)
- Mode0
 - BSpi, [141](#)
- Mode1
 - BSpi, [141](#)
- Mode2
 - BSpi, [141](#)
- Mode3
 - BSpi, [141](#)
- mon_yday
 - BDate.cpp, [177](#)
 - BTimeStamp.cpp, [216](#)
 - BTimeStampMs.cpp, [217](#)
- monDays
 - BTime.cpp, [215](#)
- month
 - BDate, [40](#)
 - BTimeStamp, [162](#)
- NOTHEADS
 - BoapServer, [113](#)
- NType
 - BSocket, [136](#)
- name
 - Boapns::BoapEntry, [97](#)
 - BoapServiceObject, [118](#)
 - BObjMember, [121](#)
- nbytes
 - BoapPacket, [109](#)
- newConnection
 - BoapServer, [113](#)
- next
 - BDictMap, [46](#)
 - BList, [85](#)
 - BNode, [93](#)
- nextFd
 - BPoll, [123](#)
- Node
 - BList::Node, [169](#)
- nodeCreate
 - BList, [86](#)
- nodeGet
 - BList, [86](#)
- None
 - BErrorTime, [60](#)
- Normal
 - BMutex, [88](#)
- nprintf
 - BDebug.h, [180](#)
- num
 - BError, [58](#)
- number
 - BArray, [18](#)
 - BList, [86](#)
- oaddress
 - BSocketAddress, [139](#)
- oaddressFrom
 - BoapMcClientObject, [99](#)
 - BoapMcComms, [102](#)
- oaddressTo
 - BoapMcClientObject, [99](#)
 - BoapMcComms, [102](#)
- oapiVersion
 - BoapClientObject, [96](#)
 - BoapMcClientObject, [99](#)
 - BoapMcComms, [102](#)
 - BoapMcServiceObject, [106](#)
 - BoapServiceObject, [118](#)
- oarg
 - BEvent, [61](#)
- oaverage
 - BTimer, [159](#)
- oboapNs
 - BoapServer, [114](#)
- oboapServer
 - BoapServerConnection, [116](#)
- oboapns
 - BoapServer, [114](#)
- oclientGoneEvent
 - BoapServer, [114](#)
- oclients
 - BoapServer, [114](#)
- ocmd
 - BoapFuncEntry, [98](#)
- ocolumnWidths
 - BTable, [153](#)
- ocomments
 - BEntryFile, [54](#)
- ocomms
 - BoapMcClientObject, [99](#)
 - BoapMcComms, [102](#)
 - BoapMcSignalObject, [106](#)
- ocond
 - BCond, [27](#)
 - BCondBool, [29](#)
 - BCondInt, [31](#)
 - BCondResource, [32](#)
 - BCondValue, [34](#)
 - BCondWrap, [36](#)
 - BRtcThreaded, [129](#)
- oconnected
 - BoapClientObject, [96](#)
- odata
 - BBuffer, [22](#)
 - BFifo, [70](#)
 - BFifoCirc, [73](#)
 - BoapPacket, [110](#)

- BRefData, 126
- BTable, 153
- odataSize
 - BBuffer, 22
- odb
 - BMySQL, 91
- odebug
 - BMySQL, 91
- odev
 - BSpi, 141
- odevName
 - BSpi, 141
- odirname
 - BDir, 48
- oendTime
 - BTimer, 159
- oerrNo
 - BError, 59
 - BErrorTime, 60
- oerrStr
 - BError, 59
 - BErrorTime, 60
- oerrTime
 - BErrorTime, 60
- oerror
 - BDir, 48
- oevent
 - BComms, 26
- oeventNum
 - BComms, 26
- oeventQueue
 - BComms, 26
- ofd
 - BRtc, 128
- ofds
 - BEvent1Int, 64
 - BEvent1Pipe, 66
 - BEventPipe, 67
 - BPoll, 123
- ofdsNext
 - BPoll, 123
- ofdsNum
 - BPoll, 123
- ofile
 - BConfig, 38
 - BFile, 79
- ofileName
 - BConfig, 38
 - BFile, 79
- ofilename
 - BEntryFile, 54
 - BFileData, 80
- ofunc
 - BoapFuncEntry, 98
- ofuncList
 - BoapServiceObject, 118
- ohashLists
 - BDict, 44
- ohashSize
 - BDict, 44
- ohostName
 - BoapServer, 114
- ohour
 - BDuration, 50
 - BTimeStamp, 163
- oi
 - BIter, 81
- oinit
 - BUrl, 169
- oisBoapns
 - BoapServer, 114
- olastPos
 - BEntryList, 56
- olen
 - BRefData, 126
 - BSocketAddress, 139
- olength
 - BList, 87
- olock
 - BCondResource, 32
 - BConfig, 38
 - BFifo, 70
 - BFifoCirc, 73
 - BMutexLock, 90
 - BMySQL, 91
 - BoapClientObject, 96
 - BQueue, 125
 - BRWLock, 130
 - BSemaphoreCount, 134
 - BStringLocked, 151
 - BTimer, 159
- olockCall
 - BoapMcComms, 103
- olockTx
 - BoapMcComms, 103
- omaxLength
 - BoapClientObject, 96
 - BoapServerConnection, 116
- omicroSecond
 - BDuration, 50
 - BTimeStamp, 163
- ominute
 - BDuration, 50
 - BTimeStamp, 163
- omode
 - BFile, 79
- omutex
 - BCond, 27
 - BCondBool, 29
 - BCondInt, 31
 - BCondResource, 32
 - BCondValue, 34
 - BCondWrap, 36
 - BMutex, 89
- oname
 - BEntry, 52

- BNameValue, [92](#)
- BoapClientObject, [96](#)
- BoapServiceObject, [118](#)
- onbytes
 - BoapPacket, [110](#)
- onet
 - BoapServer, [114](#)
- onetEvent
 - BoapServer, [114](#)
- onetEventAddress
 - BoapServer, [114](#)
- onodes
 - BList, [87](#)
- onum
 - BTimer, [159](#)
- onumOperations
 - BoapServer, [114](#)
- onumber
 - BQueue, [125](#)
- oobject
 - BoapServiceEntry, [117](#)
- oopened
 - BMysql, [91](#)
- opacket
 - BoapMcClientObject, [99](#)
 - BoapMcComms, [103](#)
- opacketMode
 - BComms, [26](#)
- opacketReqQueue
 - BoapMcComms, [103](#)
- opacketReqRx
 - BoapMcComms, [103](#)
- opacketReqTx
 - BoapMcComms, [103](#)
- opacketRx
 - BoapMcComms, [103](#)
- opacketRxSema
 - BoapMcComms, [103](#)
- opacketTx
 - BoapMcComms, [103](#)
- opacketTxQueue
 - BoapMcComms, [103](#)
- opacketTxQueueWriteNum
 - BoapMcComms, [103](#)
- opacketTxSema
 - BoapMcComms, [103](#)
- opeak
 - BTimer, [159](#)
- open
 - BConfig, [37](#)
 - BDir, [47](#)
 - BEntryFile, [53](#)
 - BFile, [77](#), [78](#)
 - BFileData, [80](#)
 - BMysql, [91](#)
- operator BNode *
 - BIter, [81](#)
- operator BString
 - BDate, [40](#)
 - BStringLocked, [151](#)
 - BTimeStamp, [162](#)
- operator const char *
 - BString, [148](#)
- operator const SockAddr *
 - BSocketAddress, [138](#)
- operator int
 - BCondBool, [28](#)
 - BError, [58](#)
 - BErrorTime, [60](#)
 - BFifoCircPos, [75](#)
- operator long
 - BAtomicCount, [20](#)
- operator Type
 - BAtomic, [19](#)
- operator<
 - BDate, [40](#)
 - BString, [148](#)
 - BTime, [156](#)
 - BTimeStamp, [162](#)
 - BTimeStampMs, [166](#)
- operator<<
 - BString.cpp, [212](#)
 - BString.h, [213](#)
- operator<=
 - BDate, [40](#)
 - BString, [148](#)
 - BTime, [156](#)
 - BTimeStamp, [162](#)
 - BTimeStampMs, [166](#)
- operator>
 - BDate, [40](#)
 - BString, [149](#)
 - BTime, [156](#)
 - BTimeStamp, [162](#)
 - BTimeStampMs, [166](#)
- operator>>
 - BString.cpp, [212](#)
 - BString.h, [213](#)
- operator>=
 - BDate, [40](#)
 - BString, [149](#)
 - BTime, [157](#)
 - BTimeStamp, [162](#)
 - BTimeStampMs, [167](#)
- operator+
 - BDict, [43](#)
 - BList, [86](#)
 - BString, [148](#)
 - BStringLocked, [151](#)
 - BTime, [156](#)
- operator++
 - BAtomic, [19](#)
 - BAtomicCount, [20](#)
 - BCondInt, [30](#)
 - BCondValue, [33](#)
 - BCondWrap, [36](#)

- operator+=
 - BCondInt, [30](#)
 - BCondValue, [34](#)
 - BCondWrap, [36](#)
 - BFifoCircPos, [75](#)
 - BString, [148](#)
 - BTime, [156](#)
- operator--
 - BAtomic, [19](#)
 - BAtomicCount, [20](#)
 - BCondInt, [30](#)
 - BCondValue, [34](#)
 - BCondWrap, [36](#)
- operator-=
 - BCondInt, [30](#)
 - BCondValue, [34](#)
 - BCondWrap, [36](#)
- operator=
 - BDict, [43](#)
 - BEntryList, [56](#)
 - BFile, [78](#)
 - BList, [86](#)
 - BMutex, [88](#)
 - BRefData, [126](#)
 - BRWLock, [130](#)
 - BSema, [131](#)
 - BSemaphore, [133](#)
 - BSemaphoreCount, [134](#)
 - BSocketAddress, [138](#)
 - BString, [149](#)
 - BStringLocked, [151](#)
 - BTimeStamp, [162](#)
- operator==
 - BDate, [40](#)
 - BFifoCircPos, [75](#)
 - BIter, [81](#)
 - BSocketAddress, [138](#)
 - BString, [149](#)
 - BTime, [156](#)
 - BTimeStamp, [162](#)
- opolicy
 - BThread, [154](#)
- opoll
 - BoapServer, [114](#)
- opos
 - BBufferStore, [24](#)
 - BFifoCircPos, [75](#)
 - BoapPacket, [110](#)
- opriority
 - BoapClientObject, [96](#)
 - BThread, [155](#)
- orate
 - BRtc, [128](#)
 - BRtcThreaded, [129](#)
- oreadPos
 - BFifo, [70](#)
 - BFifoCirc, [73](#)
- oreconnect
 - BoapClientObject, [96](#)
- orefCount
 - BRefData, [126](#)
- ores
 - BUrl, [169](#)
- oreult
 - BThread, [155](#)
- ortc
 - BRtcThreaded, [129](#)
- orunning
 - BThread, [155](#)
- orx
 - BoapClientObject, [96](#)
 - BoapServer, [114](#)
 - BoapServerConnection, [116](#)
 - BoapSignalObject, [119](#)
- osecond
 - BDuration, [50](#)
 - BTimeStamp, [163](#)
- osema
 - BSema, [132](#)
 - BSemaphore, [133](#)
 - BSemaphoreCount, [134](#)
- oseparator
 - BFileCsv, [79](#)
- oserver
 - BoapServiceObject, [118](#)
- oservice
 - BoapClientObject, [96](#)
 - BoapServiceEntry, [117](#)
- oservices
 - BoapServer, [114](#)
- osize
 - BBuffer, [22](#)
 - BFifo, [70](#)
 - BFifoCirc, [73](#)
 - BFifoCircPos, [75](#)
 - BoapPacket, [110](#)
 - BQueue, [125](#)
- oslave
 - BoapMcComms, [103](#)
- osocket
 - BoapServerConnection, [116](#)
 - BSocket, [137](#)
- osort
 - BDir, [48](#)
- ospare
 - BDuration, [50](#)
 - BTimeStamp, [164](#)
- ostackSize
 - BThread, [155](#)
- ostartTime
 - BTimer, [159](#)
- ostr
 - BString, [150](#)
 - BStringLocked, [151](#)
- oswapBytes
 - BBufferStore, [24](#)

- othread
 - BThread, [155](#)
- othreaded
 - BoapMcComms, [104](#)
 - BoapServer, [114](#)
- otime
 - BTime, [157](#)
- otimeout
 - BComms, [26](#)
 - BoapClientObject, [96](#)
 - BoapMcComms, [104](#)
- otitle
 - BTable, [153](#)
- otx
 - BoapClientObject, [96](#)
 - BoapServer, [114](#)
 - BoapServerConnection, [116](#)
 - BoapSignalObject, [119](#)
- otype
 - BEvent, [61](#)
 - BEvent1, [63](#)
- ouse
 - BCondResource, [32](#)
- ovalue
 - BAtomic, [19](#)
 - BAtomicCount, [20](#)
 - BCondBool, [29](#)
 - BCondInt, [31](#)
 - BCondValue, [34](#)
 - BCondWrap, [36](#)
 - BEntry, [52](#)
 - BNameValue, [92](#)
 - BSemaphoreCount, [134](#)
- ovmSize
 - BFifoCirc, [74](#)
- owild
 - BDir, [48](#)
- owriteNumFifoSamples
 - BFifoCirc, [74](#)
- owritePos
 - BFifo, [70](#)
 - BFifoCirc, [74](#)
- oyday
 - BDate, [41](#)
 - BTimeStamp, [164](#)
- oyear
 - BDate, [41](#)
 - BTimeStamp, [164](#)
- packetMode
 - BComms, [26](#)
- packetRecv
 - BoapMcComms, [101](#)
- packetSend
 - BoapMcComms, [101](#)
- pad
 - BString, [149](#)
- peak
 - BTimer, [158](#)
- peekHead
 - BoapPacket, [109](#)
- performCall
 - BoapClientObject, [95](#)
 - BoapMcClientObject, [99](#)
 - BoapMcComms, [101](#)
- performRecv
 - BoapClientObject, [95](#)
 - BoapMcClientObject, [99](#)
- performSend
 - BoapClientObject, [95](#)
 - BoapMcClientObject, [99](#)
 - BoapMcComms, [101](#)
 - BoapMcSignalObject, [106](#)
 - BoapSignalObject, [119](#)
- ping
 - BoapClientObject, [95](#)
- pingLocked
 - BoapClientObject, [96](#)
- PollFd
 - BPoll, [122](#)
- pop
 - BBufferStore, [23, 24](#)
 - BList, [86](#)
 - BoapPacket, [109](#)
- popHead
 - BoapPacket, [109](#)
- port
 - Boapns::BoapEntry, [97](#)
 - BSocketAddressINET, [140](#)
- pos
 - BFifoCircPos, [75](#)
- position
 - BFile, [78](#)
 - BList, [86](#)
- post
 - BSema, [131](#)
- prev
 - BList, [86](#)
 - BNode, [93](#)
- print
 - BEntry, [52](#)
 - BEntryList, [56](#)
 - BTable, [153](#)
- printLine
 - BTable, [153](#)
- printf
 - BFile, [78](#)
 - BString, [149](#)
- Priority
 - BSocket, [136](#)
- PriorityHigh
 - BSocket, [136](#)
- PriorityLow
 - BSocket, [136](#)
- PriorityNormal
 - BSocket, [136](#)
- process

- BoapMcServiceObject, 105
- BoapServer, 113
- BoapServerConnection, 116
- BoapServiceObject, 118
- processEvent
 - BoapMcServiceObject, 105
 - BoapServer, 113, 114
 - BoapServiceObject, 118
- processPacket
 - BoapMcComms, 102
- processRequest
 - BoapMcComms, 102
- processRequests
 - BoapMcComms, 102
- processRx
 - BoapMcComms, 102
- pullLine
 - BString, 149
- pullSeparators
 - BString, 149
- pullToken
 - BString, 149
- pullWord
 - BString, 149
- push
 - BBufferStore, 24
 - BList, 86
 - BoapPacket, 109, 110
- pushHead
 - BoapPacket, 110
- query
 - BMySQL, 91
- queueAdd
 - BList, 86
- queueGet
 - BList, 86
- raw
 - BSocketAddress, 138
- rdLock
 - BRWLock, 130
- read
 - BComms, 26
 - BConfig, 38
 - BDir, 47
 - BEntryFile, 53
 - BEventPipe, 67
 - BFifo, 69
 - BFifoCirc, 72
 - BFile, 78
 - BFileData, 80
 - BQueue, 124
- readAvailable
 - BComms, 26
 - BEventPipe, 67
 - BFifo, 69
 - BFifoCirc, 72
 - BQueue, 124
- readAvailableChunk
 - BFifo, 69
- readCsv
 - BFileCsv, 79
- readData
 - BFifo, 69
 - BFifoCirc, 72
- readDone
 - BFifo, 69
 - BFifoCirc, 73
- readPos
 - BFifo, 69
- readString
 - BFile, 78
 - BUrl, 168
- readWaitAvailable
 - BFifoCirc, 73
- rear
 - BArray, 18
 - BList, 87
- Recursive
 - BMutex, 88
- recv
 - BSocket, 136
- recvFrom
 - BSocket, 137
- recvFromWithTimeout
 - BSocket, 137
- recvWithTimeout
 - BSocket, 137
- removeNL
 - BString, 149
- removeSeparators
 - BString, 149
- reserved
 - BoapPacketHead, 111
- resize
 - BBuffer, 21
 - BFifo, 69
 - BoapPacket, 110
- result
 - BThread, 154
- retDouble
 - BString, 149
- retInt
 - BString, 149
- retStr
 - BString, 150
- retStrDup
 - BString, 150
- retUInt
 - BString, 150
- reverse
 - BString, 150
- roundSize
 - BBuffer.cpp, 172
 - BoapSimple.cc, 200
- run

- BoapServer, [114](#)
- running
 - BThread, [154](#)
- STREAM
 - BSocket, [136](#)
- STRBUF
 - BFile.cpp, [191](#)
- STRBUF_SIZE
 - BDebug.cpp, [179](#)
- STRIP
 - BString.cpp, [211](#)
- sampleNumber
 - BTimeStampMs, [167](#)
- second
 - BDuration, [50](#)
 - BTimeStamp, [162](#)
 - BTimeStampMs, [167](#)
- seek
 - BFile, [78](#)
- send
 - BSocket, [137](#)
- sendEvent
 - BEvent1Int, [64](#)
 - BEvent1Pipe, [66](#)
 - BoapMcServiceObject, [105](#)
 - BoapServer, [114](#)
 - BoapServiceObject, [118](#)
- sendTo
 - BSocket, [137](#)
- service
 - Boapns::BoapEntry, [97](#)
 - BoapPacketHead, [111](#)
- set
 - BCondBool, [28](#)
 - BDate, [40](#)
 - BDuration, [50](#)
 - BError, [58](#)
 - BErrorTime, [60](#)
 - BFifoCircPos, [75](#)
 - BSemaphore, [133](#)
 - BSocketAddress, [138](#)
 - BSocketAddressINET, [140](#)
 - BTime, [157](#)
 - BTimeStamp, [163](#)
- setAddress
 - BoapMcClientObject, [99](#)
 - BoapMcComms, [102](#)
- setBinary
 - BEvent1, [62](#)
 - BEvent1Error, [63](#)
- setBroadCast
 - BSocket, [137](#)
- setComms
 - BoapMcComms, [102](#)
- setCommsMode
 - BoapMcComms, [102](#)
- setConnectionPriority
 - BoapClientObject, [96](#)
- setData
 - BBuffer, [21](#)
 - BoapPacket, [110](#)
- setDebug
 - BDebug.cpp, [179](#)
 - BDebug.h, [180](#)
 - BMysql, [91](#)
- setDurationString
 - BTimeStampMs, [167](#)
- setError
 - BError, [58](#)
- setFd
 - BSocket, [137](#)
- setFirst
 - BDate, [40](#)
 - BTimeStamp, [163](#)
- setHexString
 - BBufferStore, [24](#)
- setInitPriority
 - BThread, [154](#)
- setInitStackSize
 - BThread, [154](#)
- setLast
 - BDate, [40](#)
 - BTimeStamp, [163](#)
- setLen
 - BRefData, [126](#)
- setLine
 - BEntry, [52](#)
- setMaxLength
 - BoapClientObject, [96](#)
 - BoapServerConnection, [116](#)
- setMember
 - BObj, [120](#)
- setMembers
 - BObj, [121](#)
- setName
 - BEntry, [52](#)
 - BoapServiceObject, [118](#)
- setNow
 - BDate, [40](#)
 - BTimeStamp, [163](#)
 - BTimeStampMs, [167](#)
- setPacketMode
 - BComms, [26](#)
- setPort
 - BSocketAddressINET, [140](#)
- setPos
 - BBufferStore, [24](#)
- setPriority
 - BSocket, [137](#)
 - BThread, [154](#)
- setReuseAddress
 - BSocket, [137](#)
- setSize
 - BBuffer, [21](#)
 - BFifoCircPos, [75](#)
- setSockOpt

- BSocket, 137
- setSort
 - BDir, 47
- setString
 - BDate, 40
 - BDuration, 50
 - BTime, 157
 - BTimeStamp, 163
 - BTimeStampMs, 167
- setTime
 - BTimeStamp, 163
- setTimeout
 - BComms, 26
 - BoapClientObject, 96
 - BoapMcComms, 102
- setTitle
 - BTable, 153
- setVBuf
 - BFile, 78
- setValue
 - BCondInt, 30
 - BCondValue, 34
 - BCondWrap, 36
 - BEntry, 52
 - BEntryList, 56
 - BSemaphoreCount, 134
- setValueRaw
 - BEntryList, 56
- setWild
 - BDir, 48
- setYDay
 - BDate, 40
 - BTimeStamp, 163
- setYearDay
 - BTime, 157
- shutdown
 - BSocket, 137
- signal
 - BCond, 27
- size
 - BBuffer, 21
 - BDictMap, 46
 - BFifo, 69
 - BFifoCirc, 73
 - BList, 87
 - BObjMember, 121
- SockAddr
 - BSocketAddress, 138
- SockAddrIP
 - BSocketAddressINET, 140
- sort
 - BArray, 18
 - BList, 87
- SortFunc
 - BArray, 18
 - BList, 84
- split
 - BString, 150
- start
 - BCondResource, 32
 - BDictMap, 46
 - BList, 87
 - BThread, 154
 - BTimer, 158
- startFunc
 - BThread, 154
- stop
 - BTimer, 159
- str
 - BError, 58
- subMilliseconds
 - BTimeStampMs, 167
- subSeconds
 - BTimeStampMs, 167
- subString
 - BString, 150
- swap
 - BList, 87
- THREADED
 - BoapServer, 113
- table_crc_hi
 - BCrc16.cpp, 175
- table_crc_lo
 - BCrc16.cpp, 176
- take
 - BSemaphoreCount, 134
- timedLock
 - BMutex, 89
- timedWait
 - BCond, 27
 - BCondBool, 28
 - BSema, 131
- timeoutTicks
 - BTypes.h, 220
- toBDictStringFromJson
 - BObjStringFormat.cpp, 203
 - BObjStringFormat.h, 205
- toBString
 - BDate.cpp, 177
 - BDate.h, 177
 - BDict.cpp, 181
 - BDict.h, 181
 - BObjStringFormat.cpp, 203
 - BObjStringFormat.h, 205
 - BString.cpp, 212
 - BString.h, 213, 214
 - BTimeStamp.cpp, 216
 - BTimeStamp.h, 217
- toBStringJson
 - BObjStringFormat.cpp, 203, 204
 - BObjStringFormat.h, 205, 206
- toLower
 - BString, 150
- toUpper
 - BString, 150
- fprintf

- BDebug.cpp, [179](#)
- BDebug.h, [180](#)
- transact
 - BSpi, [141](#)
- translateChar
 - BString, [150](#)
- truncate
 - BFile, [78](#)
 - BString, [150](#)
- tryLock
 - BMutex, [89](#)
- tryRdLock
 - BRWLock, [130](#)
- tryWait
 - BSema, [132](#)
- tryWrLock
 - BRWLock, [130](#)
- Type
 - BErrorTime, [60](#)
 - BMutex, [88](#)
- type
 - BEvent, [61](#)
 - BoapPacketHead, [111](#)
 - BObjMember, [121](#)
- typeComp
 - BObjMember, [121](#)
- typeName
 - BObjMember, [121](#)
- UInt16
 - BoapSimple.h, [201](#)
- UInt32
 - BoapSimple.h, [201](#)
- UInt8
 - BoapSimple.h, [201](#)
- unlock
 - BCondResource, [32](#)
 - BMutex, [89](#)
 - BMutexLock, [89](#)
 - BRWLock, [130](#)
- unmapCircularBuffer
 - BFifoCirc, [73](#)
- update
 - BMysql, [91](#)
- updateHead
 - BoapPacket, [110](#)
- updateLen
 - BoapPacket, [110](#)
- valid
 - Blter, [81](#)
- validate
 - BoapServerConnection, [116](#)
- value
 - BCondBool, [28](#)
 - BCondInt, [30](#)
 - BCondValue, [34](#)
 - BCondWrap, [36](#)
 - BDictItem, [44](#)
 - BSemaphoreCount, [134](#)
- Wait
 - BComms, [25](#)
- wait
 - BComms, [26](#)
 - BCond, [27](#)
 - BCondBool, [28](#)
 - BRtc, [127](#)
 - BRtcThreaded, [129](#)
 - BSema, [132](#)
 - BSemaphore, [133](#)
 - BSemaphoreCount, [134](#)
- WaitError
 - BComms, [25](#)
- WaitNone
 - BComms, [25](#)
- WaitRead
 - BComms, [25](#)
- WaitWrite
 - BComms, [25](#)
- waitForCompletion
 - BThread, [154](#)
- waitLessThan
 - BCondInt, [30](#)
 - BCondValue, [34](#)
 - BCondWrap, [36](#)
- waitLessThanOrEqual
 - BCondInt, [31](#)
 - BCondValue, [34](#)
 - BCondWrap, [36](#)
- waitMoreThanOrEqual
 - BCondInt, [31](#)
 - BCondValue, [34](#)
 - BCondWrap, [36](#)
- wild
 - BDir.cpp, [182](#)
- wildString
 - BDir.cpp, [182](#)
- wprintf
 - BDebug.h, [180](#)
- wrLock
 - BRWLock, [130](#)
- write
 - BComms, [26](#)
 - BConfig, [38](#)
 - BEntryFile, [54](#)
 - BEventPipe, [67](#)
 - BFifo, [69](#), [70](#)
 - BFifoCirc, [73](#)
 - BFile, [78](#)
 - BFileData, [80](#)
 - BQueue, [124](#)
- writeAvailable
 - BComms, [26](#)
 - BEventPipe, [67](#)
 - BFifo, [70](#)
 - BFifoCirc, [73](#)
 - BQueue, [124](#)

- writeAvailableChunk
 - BFifo, [70](#)
- writeBackup
 - BFifo, [70](#)
- writeCsv
 - BFileCsv, [79](#)
- writeData
 - BBuffer, [21](#)
 - BFifo, [70](#)
 - BFifoCirc, [73](#)
 - BUrl, [168](#)
- writeDone
 - BFifo, [70](#)
 - BFifoCirc, [73](#)
- writeList
 - BEntryFile, [54](#)
- writeString
 - BFile, [78](#)
- writeWaitAvailable
 - BFifoCirc, [73](#)
- yday
 - BDate, [40](#)
 - BTimeStamp, [163](#)
 - BTimeStampMs, [167](#)
- year
 - BDate, [40](#)
 - BTimeStamp, [163](#)
 - BTimeStampMs, [167](#)
- yearDays
 - BTime.cpp, [215](#)
- yearIsLeap
 - BTime.cpp, [215](#)