

LibBeamApi

1.3.0

Generated by Doxygen 1.5.6

Wed Jun 3 13:59:47 2009

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class Hierarchy	3
3	Class Index	5
3.1	Class List	5
4	File Index	7
4.1	File List	7
5	Namespace Documentation	9
5.1	Boapns Namespace Reference	9
5.1.1	Function Documentation	10
5.1.1.1	addEntry	10
5.1.1.2	Boapns	10
5.1.1.3	delEntry	10
5.1.1.4	getEntry	10
5.1.1.5	getEntryList	10
5.1.1.6	getNewName	10
5.1.1.7	getVersion	10
5.1.2	Variable Documentation	10
5.1.2.1	apiVersion	10
6	Class Documentation	11
6.1	BArray< T > Class Template Reference	11
6.1.1	Detailed Description	11
6.1.2	Constructor & Destructor Documentation	11
6.1.2.1	BArray	11

6.1.2.2	BArray	11
6.1.2.3	BArray	11
6.1.3	Member Function Documentation	11
6.1.3.1	append	11
6.1.3.2	append	11
6.2	BAtomicCount Class Reference	13
6.2.1	Detailed Description	13
6.2.2	Constructor & Destructor Documentation	13
6.2.2.1	BAtomicCount	13
6.2.3	Member Function Documentation	13
6.2.3.1	getValue	13
6.2.3.2	add	13
6.2.3.3	operator++	13
6.2.3.4	operator++	13
6.2.3.5	operator--	13
6.2.3.6	operator--	13
6.2.3.7	operator long	13
6.2.4	Member Data Documentation	13
6.2.4.1	ovalue	13
6.3	BBuffer Class Reference	15
6.3.1	Constructor & Destructor Documentation	15
6.3.1.1	BBuffer	15
6.3.1.2	~BBuffer	16
6.3.2	Member Function Documentation	16
6.3.2.1	setSize	16
6.3.2.2	setData	16
6.3.2.3	writeData	16
6.3.2.4	data	16
6.3.2.5	size	16
6.3.2.6	resize	16
6.3.3	Member Data Documentation	16
6.3.3.1	odataSize	16
6.3.3.2	odata	16
6.3.3.3	osize	16
6.4	BBufferStore Class Reference	17
6.4.1	Constructor & Destructor Documentation	18

6.4.1.1	BBufferStore	18
6.4.1.2	~BBufferStore	18
6.4.2	Member Function Documentation	18
6.4.2.1	getPos	18
6.4.2.2	setPos	18
6.4.2.3	getHexString	18
6.4.2.4	setHexString	18
6.4.2.5	push	18
6.4.2.6	push	18
6.4.2.7	push	18
6.4.2.8	push	18
6.4.2.9	push	18
6.4.2.10	push	18
6.4.2.11	push	19
6.4.2.12	push	19
6.4.2.13	push	19
6.4.2.14	push	19
6.4.2.15	push	19
6.4.2.16	push	19
6.4.2.17	push	19
6.4.2.18	push	19
6.4.2.19	pop	19
6.4.2.20	pop	19
6.4.2.21	pop	19
6.4.2.22	pop	19
6.4.2.23	pop	19
6.4.2.24	pop	19
6.4.2.25	pop	20
6.4.2.26	pop	20
6.4.2.27	pop	20
6.4.2.28	pop	20
6.4.2.29	pop	20
6.4.2.30	pop	20
6.4.2.31	pop	20
6.4.2.32	pop	20
6.4.2.33	copyWithSwap	20

6.4.3	Member Data Documentation	20
6.4.3.1	opos	20
6.5	BCond Class Reference	21
6.5.1	Constructor & Destructor Documentation	21
6.5.1.1	BCond	21
6.5.1.2	~BCond	21
6.5.2	Member Function Documentation	21
6.5.2.1	signal	21
6.5.2.2	wait	21
6.5.2.3	timedWait	21
6.5.3	Member Data Documentation	21
6.5.3.1	omutex	21
6.5.3.2	ocond	21
6.6	BCondBool Class Reference	22
6.6.1	Detailed Description	22
6.6.2	Constructor & Destructor Documentation	22
6.6.2.1	BCondBool	22
6.6.2.2	~BCondBool	22
6.6.3	Member Function Documentation	22
6.6.3.1	set	22
6.6.3.2	clear	23
6.6.3.3	value	23
6.6.3.4	wait	23
6.6.3.5	timedWait	23
6.6.3.6	operator int	23
6.6.4	Member Data Documentation	23
6.6.4.1	omutex	23
6.6.4.2	ocond	23
6.6.4.3	ovalue	23
6.7	BCondInt Class Reference	24
6.7.1	Detailed Description	25
6.7.2	Constructor & Destructor Documentation	25
6.7.2.1	BCondInt	25
6.7.2.2	~BCondInt	25
6.7.3	Member Function Documentation	25
6.7.3.1	setValue	25

6.7.3.2	increment	25
6.7.3.3	decrement	25
6.7.3.4	value	25
6.7.3.5	wait	25
6.7.3.6	waitIncrement	25
6.7.3.7	waitNotZero	25
6.7.3.8	waitNotZeroDecrement	25
6.7.3.9	tryNotZeroDecrement	25
6.7.3.10	timedWait	26
6.7.3.11	operator++	26
6.7.3.12	operator--	26
6.7.4	Member Data Documentation	26
6.7.4.1	omutex	26
6.7.4.2	ocond	26
6.7.4.3	ovalue	26
6.8	BCondValue Class Reference	27
6.8.1	Detailed Description	28
6.8.2	Constructor & Destructor Documentation	28
6.8.2.1	BCondValue	28
6.8.2.2	~BCondValue	28
6.8.3	Member Function Documentation	28
6.8.3.1	setValue	28
6.8.3.2	value	28
6.8.3.3	increment	28
6.8.3.4	decrement	28
6.8.3.5	waitMoreThanOrEqual	28
6.8.3.6	waitLessThanOrEqual	28
6.8.3.7	waitLessThan	28
6.8.3.8	operator+=	28
6.8.3.9	operator-=	28
6.8.3.10	operator++	29
6.8.3.11	operator--	29
6.8.4	Member Data Documentation	29
6.8.4.1	omutex	29
6.8.4.2	ocond	29
6.8.4.3	ovalue	29

6.9	BCondWrap Class Reference	30
6.9.1	Constructor & Destructor Documentation	31
6.9.1.1	BCondWrap	31
6.9.1.2	~BCondWrap	31
6.9.2	Member Function Documentation	31
6.9.2.1	setValue	31
6.9.2.2	value	31
6.9.2.3	increment	31
6.9.2.4	decrement	31
6.9.2.5	waitMoreThanOrEqual	31
6.9.2.6	waitLessThanOrEqual	31
6.9.2.7	waitLessThan	31
6.9.2.8	operator+=	31
6.9.2.9	operator-=	32
6.9.2.10	operator++	32
6.9.2.11	operator--	32
6.9.2.12	diff	32
6.9.3	Member Data Documentation	32
6.9.3.1	omutex	32
6.9.3.2	ocond	32
6.9.3.3	ovalue	32
6.10	BConfig Class Reference	33
6.10.1	Detailed Description	33
6.10.2	Member Function Documentation	33
6.10.2.1	open	33
6.10.2.2	read	33
6.10.2.3	write	33
6.10.2.4	findValue	33
6.10.3	Member Data Documentation	33
6.10.3.1	olock	33
6.10.3.2	ofile	33
6.11	BDebugBacktrace Class Reference	35
6.11.1	Constructor & Destructor Documentation	35
6.11.1.1	BDebugBacktrace	35
6.11.1.2	~BDebugBacktrace	35
6.11.2	Member Function Documentation	35

6.11.2.1	dumpBacktraceStdout	35
6.11.2.2	dumpBacktraceFile	35
6.11.2.3	dumpBacktraceSyslog	35
6.11.2.4	dumpBacktrace	35
6.12	BDict< Type > Class Template Reference	36
6.12.1	Member Typedef Documentation	36
6.12.1.1	iterator	36
6.12.2	Member Function Documentation	36
6.12.2.1	hasKey	36
6.12.2.2	key	36
6.12.2.3	del	36
6.12.2.4	find	36
6.12.2.5	operator[36
6.12.2.6	operator[37
6.12.2.7	operator[37
6.12.2.8	operator+	37
6.12.2.9	operator=	37
6.13	BDictItem< Type > Class Template Reference	38
6.13.1	Detailed Description	38
6.13.2	Constructor & Destructor Documentation	38
6.13.2.1	BDictItem	38
6.13.3	Member Data Documentation	38
6.13.3.1	key	38
6.13.3.2	value	38
6.14	BDictMap< Value > Class Template Reference	39
6.14.1	Detailed Description	39
6.14.2	Member Typedef Documentation	40
6.14.2.1	iterator	40
6.14.3	Member Function Documentation	40
6.14.3.1	clear	40
6.14.3.2	hasKey	40
6.14.3.3	key	40
6.14.3.4	size	40
6.14.3.5	start	40
6.14.3.6	isEnd	40
6.14.3.7	next	40

6.14.3.8	del	40
6.14.3.9	del	40
6.14.3.10	operator[40
6.14.3.11	operator[40
6.15	BDir Class Reference	41
6.15.1	Detailed Description	42
6.15.2	Constructor & Destructor Documentation	42
6.15.2.1	BDir	42
6.15.2.2	BDir	42
6.15.2.3	~BDir	42
6.15.3	Member Function Documentation	42
6.15.3.1	open	42
6.15.3.2	error	42
6.15.3.3	read	42
6.15.3.4	clear	42
6.15.3.5	setWild	42
6.15.3.6	setSort	42
6.15.3.7	entryName	43
6.15.3.8	entryStat	43
6.15.3.9	entryStat64	43
6.15.4	Member Data Documentation	43
6.15.4.1	oerror	43
6.15.4.2	odirname	43
6.15.4.3	owild	43
6.15.4.4	osort	43
6.16	BEntry Class Reference	44
6.16.1	Detailed Description	44
6.16.2	Constructor & Destructor Documentation	45
6.16.2.1	BEntry	45
6.16.2.2	BEntry	45
6.16.2.3	BEntry	45
6.16.3	Member Function Documentation	45
6.16.3.1	getName	45
6.16.3.2	getValue	45
6.16.3.3	setLine	45
6.16.3.4	setName	45

6.16.3.5	setValue	45
6.16.3.6	line	45
6.16.3.7	print	45
6.16.4	Member Data Documentation	46
6.16.4.1	oname	46
6.16.4.2	ovalue	46
6.17	BEntryFile Class Reference	47
6.17.1	Detailed Description	47
6.17.2	Constructor & Destructor Documentation	48
6.17.2.1	BEntryFile	48
6.17.2.2	BEntryFile	48
6.17.2.3	~BEntryFile	48
6.17.3	Member Function Documentation	48
6.17.3.1	open	48
6.17.3.2	read	48
6.17.3.3	write	48
6.17.3.4	writeList	48
6.17.3.5	clear	48
6.17.4	Member Data Documentation	48
6.17.4.1	ofilename	48
6.17.4.2	ocomments	48
6.18	BEntryList Class Reference	49
6.18.1	Detailed Description	50
6.18.2	Constructor & Destructor Documentation	50
6.18.2.1	BEntryList	50
6.18.3	Member Function Documentation	50
6.18.3.1	isSet	50
6.18.3.2	find	50
6.18.3.3	findValue	50
6.18.3.4	setValue	50
6.18.3.5	setValueRaw	50
6.18.3.6	deleteEntry	50
6.18.3.7	print	50
6.18.3.8	getString	51
6.18.3.9	insert	51
6.18.3.10	del	51

6.18.3.11	clear	51
6.18.3.12	operator=	51
6.18.4	Member Data Documentation	51
6.18.4.1	olastPos	51
6.19	BError Class Reference	52
6.19.1	Detailed Description	53
6.19.2	Member Enumeration Documentation	53
6.19.2.1	Type	53
6.19.3	Constructor & Destructor Documentation	53
6.19.3.1	BError	53
6.19.3.2	BError	53
6.19.4	Member Function Documentation	53
6.19.4.1	copy	53
6.19.4.2	set	53
6.19.4.3	clear	53
6.19.4.4	setError	53
6.19.4.5	getString	54
6.19.4.6	getErrorNo	54
6.19.4.7	operator int	54
6.19.5	Member Data Documentation	54
6.19.5.1	oerrNo	54
6.19.5.2	oerrStr	54
6.20	BEvent Class Reference	55
6.20.1	Detailed Description	55
6.20.2	Constructor & Destructor Documentation	55
6.20.2.1	BEvent	55
6.20.2.2	~BEvent	55
6.20.3	Member Function Documentation	55
6.20.3.1	getType	55
6.20.3.2	getBinary	55
6.20.3.3	setBinary	55
6.20.4	Member Data Documentation	56
6.20.4.1	otype	56
6.21	BEventError Class Reference	57
6.21.1	Constructor & Destructor Documentation	57
6.21.1.1	BEventError	57

6.21.2	Member Function Documentation	57
6.21.2.1	getBinary	57
6.21.2.2	setBinary	57
6.22	BEventInt Class Reference	58
6.22.1	Detailed Description	58
6.22.2	Constructor & Destructor Documentation	58
6.22.2.1	BEventInt	58
6.22.2.2	~BEventInt	58
6.22.3	Member Function Documentation	58
6.22.3.1	clear	58
6.22.3.2	sendEvent	58
6.22.3.3	getEvent	59
6.22.3.4	getFd	59
6.22.4	Member Data Documentation	59
6.22.4.1	ofds	59
6.23	BEventPipe Class Reference	60
6.23.1	Detailed Description	60
6.23.2	Constructor & Destructor Documentation	60
6.23.2.1	BEventPipe	60
6.23.2.2	~BEventPipe	60
6.23.3	Member Function Documentation	60
6.23.3.1	clear	60
6.23.3.2	sendEvent	61
6.23.3.3	getEvent	61
6.23.3.4	getReceiveFd	61
6.23.4	Member Data Documentation	61
6.23.4.1	ofds	61
6.24	BFifo< Type > Class Template Reference	62
6.24.1	Detailed Description	63
6.24.2	Member Enumeration Documentation	63
6.24.2.1	"@0	63
6.24.3	Constructor & Destructor Documentation	63
6.24.3.1	BFifo	63
6.24.3.2	~BFifo	63
6.24.4	Member Function Documentation	63
6.24.4.1	size	63

6.24.4.2	clear	63
6.24.4.3	writeAvailable	64
6.24.4.4	writeWaitAvailable	64
6.24.4.5	write	64
6.24.4.6	writeData	64
6.24.4.7	writeDone	64
6.24.4.8	readAvailable	64
6.24.4.9	readWaitAvailable	64
6.24.4.10	read	64
6.24.4.11	readData	64
6.24.4.12	readDone	64
6.24.4.13	operator[64
6.24.4.14	mapCircularBuffer	65
6.24.4.15	unmapCircularBuffer	65
6.24.5	Member Data Documentation	65
6.24.5.1	oLock	65
6.24.5.2	ovmSize	65
6.24.5.3	osize	65
6.24.5.4	odata	65
6.24.5.5	owritePos	65
6.24.5.6	owriteNumFifoSamples	65
6.24.5.7	oreadPos	65
6.25	BFifoPos Class Reference	66
6.25.1	Detailed Description	66
6.25.2	Constructor & Destructor Documentation	66
6.25.2.1	BFifoPos	66
6.25.3	Member Function Documentation	66
6.25.3.1	setSize	66
6.25.3.2	set	66
6.25.3.3	pos	67
6.25.3.4	increment	67
6.25.3.5	difference	67
6.25.3.6	operator int	67
6.25.3.7	operator+=	67
6.25.3.8	operator==	67
6.25.3.9	operator"!="	67

6.25.4	Member Data Documentation	67
6.25.4.1	osize	67
6.25.4.2	opos	67
6.26	BFile Class Reference	68
6.26.1	Detailed Description	69
6.26.2	Constructor & Destructor Documentation	69
6.26.2.1	BFile	69
6.26.2.2	BFile	69
6.26.2.3	~BFile	69
6.26.3	Member Function Documentation	69
6.26.3.1	open	69
6.26.3.2	open	69
6.26.3.3	open	70
6.26.3.4	close	70
6.26.3.5	isEnd	70
6.26.3.6	getFd	70
6.26.3.7	length	70
6.26.3.8	setVBuf	70
6.26.3.9	read	70
6.26.3.10	readString	70
6.26.3.11	fgets	70
6.26.3.12	write	70
6.26.3.13	writeString	70
6.26.3.14	seek	70
6.26.3.15	position	71
6.26.3.16	printf	71
6.26.3.17	truncate	71
6.26.3.18	flush	71
6.26.3.19	operator=	71
6.26.4	Member Data Documentation	71
6.26.4.1	ofile	71
6.26.4.2	ofilename	71
6.26.4.3	omode	71
6.27	BIter Class Reference	72
6.27.1	Detailed Description	72
6.27.2	Constructor & Destructor Documentation	72

6.27.2.1	BIter	72
6.27.3	Member Function Documentation	72
6.27.3.1	operator BNode *	72
6.27.3.2	operator==	72
6.27.4	Member Data Documentation	72
6.27.4.1	oi	72
6.28	BList< T > Class Template Reference	73
6.28.1	Detailed Description	75
6.28.2	Member Typedef Documentation	76
6.28.2.1	SortFunc	76
6.28.3	Constructor & Destructor Documentation	76
6.28.3.1	BList	76
6.28.3.2	BList	76
6.28.3.3	~BList	76
6.28.4	Member Function Documentation	76
6.28.4.1	start	76
6.28.4.2	begin	76
6.28.4.3	end	76
6.28.4.4	end	76
6.28.4.5	next	76
6.28.4.6	prev	76
6.28.4.7	goTo	76
6.28.4.8	position	76
6.28.4.9	number	77
6.28.4.10	size	77
6.28.4.11	isEnd	77
6.28.4.12	front	77
6.28.4.13	rear	77
6.28.4.14	get	77
6.28.4.15	get	77
6.28.4.16	append	77
6.28.4.17	insert	77
6.28.4.18	insertAfter	77
6.28.4.19	clear	78
6.28.4.20	del	78
6.28.4.21	deleteLast	78

6.28.4.22	deleteFirst	78
6.28.4.23	push	78
6.28.4.24	pop	78
6.28.4.25	queueAdd	78
6.28.4.26	queueGet	78
6.28.4.27	append	78
6.28.4.28	swap	78
6.28.4.29	sort	78
6.28.4.30	sort	79
6.28.4.31	operator=	79
6.28.4.32	operator[79
6.28.4.33	operator[79
6.28.4.34	operator[79
6.28.4.35	operator[79
6.28.4.36	operator+	79
6.28.4.37	nodeGet	79
6.28.4.38	nodeGet	79
6.28.4.39	nodeCreate	79
6.28.4.40	nodeCreate	79
6.28.5	Member Data Documentation	79
6.28.5.1	onodes	79
6.28.5.2	olength	79
6.29	BList< T >::Node Class Reference	80
6.29.1	Constructor & Destructor Documentation	80
6.29.1.1	Node	80
6.29.2	Member Data Documentation	80
6.29.2.1	item	80
6.30	BMutex Class Reference	81
6.30.1	Detailed Description	81
6.30.2	Member Enumeration Documentation	82
6.30.2.1	Type	82
6.30.3	Constructor & Destructor Documentation	82
6.30.3.1	BMutex	82
6.30.3.2	BMutex	82
6.30.3.3	~BMutex	82
6.30.4	Member Function Documentation	82

6.30.4.1	lock	82
6.30.4.2	timedLock	82
6.30.4.3	unlock	82
6.30.4.4	tryLock	82
6.30.4.5	operator=	82
6.30.5	Member Data Documentation	82
6.30.5.1	omutex	82
6.31	BMutexLock Class Reference	83
6.31.1	Constructor & Destructor Documentation	83
6.31.1.1	BMutexLock	83
6.31.1.2	~BMutexLock	83
6.31.2	Member Data Documentation	83
6.31.2.1	olock	83
6.32	BMySQL Class Reference	84
6.32.1	Constructor & Destructor Documentation	85
6.32.1.1	BMySQL	85
6.32.1.2	~BMySQL	85
6.32.2	Member Function Documentation	85
6.32.2.1	open	85
6.32.2.2	get	85
6.32.2.3	insert	85
6.32.2.4	update	85
6.32.2.5	query	85
6.32.2.6	db	85
6.32.2.7	setDebug	85
6.32.3	Member Data Documentation	85
6.32.3.1	odb	85
6.32.3.2	oopened	85
6.32.3.3	odebug	85
6.32.3.4	olock	85
6.33	BNameValue< T > Class Template Reference	86
6.33.1	Constructor & Destructor Documentation	86
6.33.1.1	BNameValue	86
6.33.1.2	BNameValue	86
6.33.2	Member Function Documentation	86
6.33.2.1	getName	86

6.33.2.2	getValue	86
6.33.3	Member Data Documentation	86
6.33.3.1	oname	86
6.33.3.2	ovalue	86
6.34	BNameValueList< T > Class Template Reference	87
6.34.1	Member Function Documentation	87
6.34.1.1	find	87
6.34.1.2	findPos	87
6.35	BNode Class Reference	88
6.35.1	Constructor & Destructor Documentation	88
6.35.1.1	BNode	88
6.35.2	Member Data Documentation	88
6.35.2.1	next	88
6.35.2.2	prev	88
6.36	BoapClientObject Class Reference	89
6.36.1	Constructor & Destructor Documentation	90
6.36.1.1	BoapClientObject	90
6.36.1.2	BoapClientObject	90
6.36.2	Member Function Documentation	90
6.36.2.1	connectService	90
6.36.2.2	disconnectService	90
6.36.2.3	getServiceName	90
6.36.2.4	ping	90
6.36.2.5	setConnectionPriority	91
6.36.2.6	setMaxLength	91
6.36.2.7	setTimeout	91
6.36.2.8	pingLocked	91
6.36.2.9	checkApiVersion	91
6.36.2.10	performCall	91
6.36.2.11	performSend	91
6.36.2.12	performRecv	91
6.36.2.13	connectService	92
6.36.2.14	performSend	92
6.36.2.15	performRecv	92
6.36.2.16	performCall	92
6.36.3	Member Data Documentation	92

6.36.3.1	oname	92
6.36.3.2	oapiVersion	92
6.36.3.3	opriority	92
6.36.3.4	oservice	92
6.36.3.5	oconnected	92
6.36.3.6	omaxLength	92
6.36.3.7	otx	92
6.36.3.8	orx	92
6.36.3.9	olock	92
6.36.3.10	otimeout	92
6.36.3.11	oreconnect	92
6.37	Boapns::BoapEntry Class Reference	93
6.37.1	Constructor & Destructor Documentation	93
6.37.1.1	BoapEntry	93
6.37.2	Member Data Documentation	93
6.37.2.1	name	93
6.37.2.2	hostName	93
6.37.2.3	addressList	93
6.37.2.4	port	93
6.37.2.5	service	93
6.38	BoapFuncEntry Class Reference	94
6.38.1	Constructor & Destructor Documentation	94
6.38.1.1	BoapFuncEntry	94
6.38.1.2	BoapFuncEntry	94
6.38.2	Member Data Documentation	94
6.38.2.1	ocmd	94
6.38.2.2	ofunc	94
6.38.2.3	ocmd	94
6.39	Boapns::Boapns Class Reference	95
6.39.1	Constructor & Destructor Documentation	95
6.39.1.1	Boapns	95
6.39.2	Member Function Documentation	95
6.39.2.1	getVersion	95
6.39.2.2	getEntryList	95
6.39.2.3	getEntry	95
6.39.2.4	addEntry	95

6.39.2.5	delEntry	95
6.39.2.6	getNewName	95
6.40	BoapPacket Class Reference	96
6.40.1	Constructor & Destructor Documentation	97
6.40.1.1	BoapPacket	97
6.40.1.2	~BoapPacket	97
6.40.1.3	BoapPacket	97
6.40.1.4	~BoapPacket	97
6.40.2	Member Function Documentation	97
6.40.2.1	getCmd	97
6.40.2.2	peekHead	97
6.40.2.3	pushHead	97
6.40.2.4	popHead	97
6.40.2.5	updateHead	97
6.40.2.6	resize	97
6.40.2.7	setData	97
6.40.2.8	nbytes	97
6.40.2.9	data	97
6.40.2.10	pushHead	98
6.40.2.11	push	98
6.40.2.12	push	98
6.40.2.13	push	98
6.40.2.14	push	98
6.40.2.15	push	98
6.40.2.16	push	98
6.40.2.17	push	98
6.40.2.18	push	98
6.40.2.19	push	98
6.40.2.20	push	98
6.40.2.21	popHead	98
6.40.2.22	pop	98
6.40.2.23	pop	98
6.40.2.24	pop	99
6.40.2.25	pop	99
6.40.2.26	pop	99
6.40.2.27	pop	99

6.40.2.28	pop	99
6.40.2.29	pop	99
6.40.2.30	pop	99
6.40.2.31	pop	99
6.40.2.32	updateLen	99
6.40.3	Member Data Documentation	99
6.40.3.1	osize	99
6.40.3.2	onbytes	99
6.40.3.3	odata	99
6.40.3.4	opos	100
6.41	BoapPacketHead Struct Reference	101
6.41.1	Member Data Documentation	101
6.41.1.1	type	101
6.41.1.2	length	101
6.41.1.3	service	101
6.41.1.4	cmd	101
6.41.1.5	length	101
6.41.1.6	type	101
6.41.1.7	service	101
6.41.1.8	cmd	101
6.41.1.9	reserved	101
6.42	BoapServer Class Reference	102
6.42.1	Member Enumeration Documentation	103
6.42.1.1	"@1	103
6.42.2	Constructor & Destructor Documentation	104
6.42.2.1	BoapServer	104
6.42.2.2	~BoapServer	104
6.42.2.3	BoapServer	104
6.42.3	Member Function Documentation	104
6.42.3.1	init	104
6.42.3.2	run	104
6.42.3.3	processEvent	104
6.42.3.4	addObject	104
6.42.3.5	process	104
6.42.3.6	sendEvent	104
6.42.3.7	getSocket	104

6.42.3.8	getEventSocket	104
6.42.3.9	processEvent	104
6.42.3.10	getHostName	104
6.42.3.11	clientGone	104
6.42.3.12	getConnectionsNumber	104
6.42.3.13	newConnection	104
6.42.3.14	function	104
6.42.3.15	init	106
6.42.3.16	run	106
6.42.3.17	processEvent	106
6.42.3.18	addObject	106
6.42.3.19	process	106
6.42.3.20	sendEvent	106
6.42.3.21	getSocket	106
6.42.3.22	getEventSocket	106
6.42.3.23	processEvent	106
6.42.3.24	getHostName	106
6.42.4	Member Data Documentation	106
6.42.4.1	othreaded	106
6.42.4.2	oisBoapns	106
6.42.4.3	oboapns	106
6.42.4.4	oclients	106
6.42.4.5	oclientGoneEvent	106
6.42.4.6	oservices	106
6.42.4.7	opoll	106
6.42.4.8	onet	106
6.42.4.9	onetEvent	106
6.42.4.10	onetEventAddress	106
6.42.4.11	ohostName	106
6.42.4.12	onumOperations	106
6.42.4.13	oboapNs	106
6.42.4.14	orx	106
6.42.4.15	otx	106
6.43	BoapServerConnection Class Reference	108
6.43.1	Constructor & Destructor Documentation	108
6.43.1.1	BoapServerConnection	108

6.43.1.2	~BoapServerConnection	108
6.43.2	Member Function Documentation	108
6.43.2.1	init	108
6.43.2.2	process	109
6.43.2.3	getSocket	109
6.43.2.4	setMaxLength	109
6.43.2.5	getHead	109
6.43.2.6	validate	109
6.43.2.7	function	109
6.43.3	Member Data Documentation	109
6.43.3.1	oboapServer	109
6.43.3.2	osocket	109
6.43.3.3	orx	109
6.43.3.4	otx	109
6.43.3.5	omaxLength	109
6.44	BoapServiceEntry Class Reference	110
6.44.1	Constructor & Destructor Documentation	110
6.44.1.1	BoapServiceEntry	110
6.44.1.2	BoapServiceEntry	110
6.44.2	Member Data Documentation	110
6.44.2.1	oservice	110
6.44.2.2	oobject	110
6.45	BoapServiceObject Class Reference	111
6.45.1	Constructor & Destructor Documentation	113
6.45.1.1	BoapServiceObject	113
6.45.1.2	~BoapServiceObject	113
6.45.1.3	BoapServiceObject	113
6.45.1.4	~BoapServiceObject	113
6.45.2	Member Function Documentation	113
6.45.2.1	setName	113
6.45.2.2	sendEvent	113
6.45.2.3	processEvent	113
6.45.2.4	name	113
6.45.2.5	doPing	113
6.45.2.6	doConnectionPriority	113
6.45.2.7	process	113

6.45.2.8	processEvent	113
6.45.2.9	sendEvent	113
6.45.2.10	sendEvent	113
6.45.2.11	processEvent	113
6.45.2.12	name	113
6.45.2.13	process	113
6.45.2.14	processEvent	113
6.45.2.15	sendEvent	113
6.45.3	Member Data Documentation	113
6.45.3.1	oserver	113
6.45.3.2	oname	113
6.45.3.3	oapiVersion	113
6.45.3.4	ofuncList	113
6.46	BoapSignalObject Class Reference	115
6.46.1	Constructor & Destructor Documentation	115
6.46.1.1	BoapSignalObject	115
6.46.1.2	BoapSignalObject	115
6.46.2	Member Function Documentation	115
6.46.2.1	performSend	115
6.46.2.2	performSend	115
6.46.3	Member Data Documentation	115
6.46.3.1	otx	115
6.46.3.2	orx	115
6.47	BObj Class Reference	117
6.47.1	Constructor & Destructor Documentation	117
6.47.1.1	BObj	117
6.47.1.2	~BObj	117
6.47.2	Member Function Documentation	117
6.47.2.1	getType	117
6.47.2.2	getMembers	117
6.47.2.3	setMembers	117
6.47.2.4	getDebugString	117
6.48	BObject Class Reference	118
6.48.1	Constructor & Destructor Documentation	119
6.48.1.1	BObject	119
6.48.1.2	~BObject	119

6.48.2	Member Function Documentation	119
6.48.2.1	getBinary	119
6.48.2.2	setBinary	119
6.48.2.3	getString	119
6.48.2.4	setString	119
6.48.2.5	getMemberList	119
6.48.2.6	addMember	119
6.48.2.7	printIt	119
6.48.2.8	getType	119
6.48.2.9	createObj	119
6.48.3	Member Data Documentation	119
6.48.3.1	otype	119
6.49	BPoll Class Reference	120
6.49.1	Detailed Description	120
6.49.2	Member Typedef Documentation	121
6.49.2.1	PollFd	121
6.49.3	Constructor & Destructor Documentation	121
6.49.3.1	BPoll	121
6.49.3.2	~BPoll	121
6.49.4	Member Function Documentation	121
6.49.4.1	append	121
6.49.4.2	delFd	121
6.49.4.3	doPoll	121
6.49.4.4	getPollFdsNum	121
6.49.4.5	getPollFds	121
6.49.4.6	clear	121
6.49.4.7	nextFd	121
6.49.5	Member Data Documentation	121
6.49.5.1	ofdsNum	121
6.49.5.2	ofds	121
6.49.5.3	ofdsNext	121
6.50	BRefData Class Reference	123
6.50.1	Detailed Description	123
6.50.2	Constructor & Destructor Documentation	124
6.50.2.1	BRefData	124
6.50.2.2	BRefData	124

6.50.2.3	BRefData	124
6.50.2.4	~BRefData	124
6.50.3	Member Function Documentation	124
6.50.3.1	copy	124
6.50.3.2	addRef	124
6.50.3.3	deleteRef	124
6.50.3.4	data	124
6.50.3.5	len	124
6.50.3.6	operator=	124
6.50.3.7	setLen	124
6.50.4	Member Data Documentation	124
6.50.4.1	orefCount	124
6.50.4.2	olen	124
6.50.4.3	odata	125
6.51	BRtc Class Reference	126
6.51.1	Detailed Description	126
6.51.2	Constructor & Destructor Documentation	126
6.51.2.1	BRtc	126
6.51.2.2	~BRtc	126
6.51.3	Member Function Documentation	126
6.51.3.1	init	126
6.51.3.2	wait	126
6.51.4	Member Data Documentation	126
6.51.4.1	ofd	126
6.51.4.2	orate	126
6.52	BRtcThreaded Class Reference	128
6.52.1	Detailed Description	128
6.52.2	Constructor & Destructor Documentation	128
6.52.2.1	BRtcThreaded	128
6.52.2.2	~BRtcThreaded	128
6.52.3	Member Function Documentation	128
6.52.3.1	init	128
6.52.3.2	wait	129
6.52.3.3	function	129
6.52.4	Member Data Documentation	129
6.52.4.1	ortc	129

6.52.4.2	orate	129
6.52.4.3	ocond	129
6.53	BRWLock Class Reference	130
6.53.1	Detailed Description	130
6.53.2	Constructor & Destructor Documentation	130
6.53.2.1	BRWLock	130
6.53.2.2	BRWLock	130
6.53.2.3	~BRWLock	130
6.53.3	Member Function Documentation	130
6.53.3.1	rdLock	130
6.53.3.2	tryRdLock	131
6.53.3.3	wrLock	131
6.53.3.4	tryWrLock	131
6.53.3.5	unlock	131
6.53.3.6	operator=	131
6.53.4	Member Data Documentation	131
6.53.4.1	olock	131
6.54	BSema Class Reference	132
6.54.1	Detailed Description	132
6.54.2	Constructor & Destructor Documentation	132
6.54.2.1	BSema	132
6.54.2.2	BSema	132
6.54.2.3	~BSema	132
6.54.3	Member Function Documentation	132
6.54.3.1	post	132
6.54.3.2	wait	133
6.54.3.3	timedWait	133
6.54.3.4	tryWait	133
6.54.3.5	getValue	133
6.54.3.6	operator=	133
6.54.4	Member Data Documentation	133
6.54.4.1	osema	133
6.55	BSocket Class Reference	134
6.55.1	Member Enumeration Documentation	135
6.55.1.1	NType	135
6.55.1.2	Priority	135

6.55.2	Constructor & Destructor Documentation	137
6.55.2.1	BSocket	137
6.55.2.2	BSocket	137
6.55.2.3	BSocket	137
6.55.2.4	~BSocket	137
6.55.3	Member Function Documentation	137
6.55.3.1	init	137
6.55.3.2	getFd	137
6.55.3.3	bind	137
6.55.3.4	connect	137
6.55.3.5	shutdown	137
6.55.3.6	close	137
6.55.3.7	listen	137
6.55.3.8	accept	137
6.55.3.9	accept	137
6.55.3.10	send	137
6.55.3.11	sendTo	137
6.55.3.12	recv	137
6.55.3.13	recvFrom	137
6.55.3.14	recvWithTimeout	137
6.55.3.15	recvFromWithTimeout	137
6.55.3.16	setSockOpt	137
6.55.3.17	getSockOpt	137
6.55.3.18	setReuseAddress	137
6.55.3.19	setBroadCast	137
6.55.3.20	setPriority	137
6.55.3.21	getMTU	137
6.55.3.22	getAddress	137
6.55.4	Member Data Documentation	137
6.55.4.1	osocket	137
6.56	BSocketAddress Class Reference	139
6.56.1	Detailed Description	139
6.56.2	Member Typedef Documentation	140
6.56.2.1	SockAddr	140
6.56.3	Constructor & Destructor Documentation	140
6.56.3.1	BSocketAddress	140

6.56.3.2	BSocketAddress	140
6.56.3.3	BSocketAddress	140
6.56.3.4	~BSocketAddress	140
6.56.4	Member Function Documentation	140
6.56.4.1	set	140
6.56.4.2	raw	140
6.56.4.3	len	140
6.56.4.4	operator=	140
6.56.4.5	operator const SocketAddr *	140
6.56.4.6	operator==	140
6.56.4.7	operator"!="	140
6.56.5	Member Data Documentation	140
6.56.5.1	olen	140
6.56.5.2	oaddress	140
6.57	BSocketAddressINET Class Reference	141
6.57.1	Detailed Description	142
6.57.2	Member Typedef Documentation	142
6.57.2.1	SocketAddrIP	142
6.57.3	Member Function Documentation	142
6.57.3.1	set	142
6.57.3.2	set	142
6.57.3.3	set	142
6.57.3.4	setPort	142
6.57.3.5	address	142
6.57.3.6	port	142
6.57.3.7	getString	142
6.57.3.8	getHostName	142
6.57.3.9	getIpAddresses	142
6.57.3.10	getIpAddressList	142
6.57.3.11	getIpAddressListAll	142
6.58	BString Class Reference	144
6.58.1	Constructor & Destructor Documentation	147
6.58.1.1	BString	147
6.58.1.2	BString	147
6.58.1.3	BString	147
6.58.1.4	BString	147

6.58.1.5	BString	147
6.58.1.6	BString	147
6.58.1.7	BString	147
6.58.1.8	BString	147
6.58.1.9	BString	147
6.58.1.10	~BString	147
6.58.2	Member Function Documentation	147
6.58.2.1	convert	147
6.58.2.2	convert	148
6.58.2.3	convert	148
6.58.2.4	convert	148
6.58.2.5	convert	148
6.58.2.6	convert	148
6.58.2.7	convertHex	148
6.58.2.8	convertHex	148
6.58.2.9	copy	148
6.58.2.10	len	148
6.58.2.11	retStr	148
6.58.2.12	retStrDup	148
6.58.2.13	retInt	149
6.58.2.14	retUInt	149
6.58.2.15	retDouble	149
6.58.2.16	compare	149
6.58.2.17	compareWild	149
6.58.2.18	compareWildExpression	149
6.58.2.19	append	149
6.58.2.20	truncate	149
6.58.2.21	pad	149
6.58.2.22	toUpper	149
6.58.2.23	toLower	149
6.58.2.24	removeNL	150
6.58.2.25	justify	150
6.58.2.26	subString	150
6.58.2.27	del	150
6.58.2.28	insert	150
6.58.2.29	printf	150

6.58.2.30	find	150
6.58.2.31	find	150
6.58.2.32	findReverse	150
6.58.2.33	getTokenList	150
6.58.2.34	removeSeparators	150
6.58.2.35	pullToken	151
6.58.2.36	pullSeparators	151
6.58.2.37	pullWord	151
6.58.2.38	pullLine	151
6.58.2.39	dirname	151
6.58.2.40	basename	151
6.58.2.41	extension	151
6.58.2.42	operator=	151
6.58.2.43	operator[151
6.58.2.44	operator==	153
6.58.2.45	operator==	153
6.58.2.46	operator>	153
6.58.2.47	operator>	153
6.58.2.48	operator<	153
6.58.2.49	operator<	153
6.58.2.50	operator>=	153
6.58.2.51	operator<=	153
6.58.2.52	operator"!=	153
6.58.2.53	operator"!=	153
6.58.2.54	operator+	153
6.58.2.55	operator+	153
6.58.2.56	operator+=	153
6.58.2.57	operator+=	153
6.58.2.58	operator+	153
6.58.2.59	operator+	153
6.58.2.60	operator+	153
6.58.2.61	operator+	153
6.58.2.62	operator const char *	153
6.58.2.63	field	153
6.58.2.64	fields	153
6.58.2.65	Init	153

6.58.2.66	inString	153
6.58.2.67	isSpace	153
6.58.3	Member Data Documentation	153
6.58.3.1	ostr	153
6.59	BStringLocked Class Reference	155
6.59.1	Constructor & Destructor Documentation	155
6.59.1.1	BStringLocked	155
6.59.1.2	BStringLocked	155
6.59.1.3	BStringLocked	155
6.59.2	Member Function Documentation	155
6.59.2.1	len	155
6.59.2.2	operator BString	155
6.59.2.3	operator+	155
6.59.2.4	operator=	155
6.59.3	Member Data Documentation	155
6.59.3.1	olock	155
6.59.3.2	ostr	155
6.60	BStringMutex Class Reference	157
6.60.1	Constructor & Destructor Documentation	157
6.60.1.1	BStringMutex	157
6.61	BTable Class Reference	158
6.61.1	Constructor & Destructor Documentation	158
6.61.1.1	BTable	158
6.61.1.2	~BTable	158
6.61.2	Member Function Documentation	158
6.61.2.1	setTitle	158
6.61.2.2	addRow	158
6.61.2.3	print	158
6.61.2.4	calculateWidths	158
6.61.2.5	printLine	158
6.61.3	Member Data Documentation	158
6.61.3.1	otitle	158
6.61.3.2	odata	158
6.61.3.3	ocolumnWidths	158
6.62	BThread Class Reference	160
6.62.1	Constructor & Destructor Documentation	161

6.62.1.1	BThread	161
6.62.1.2	~BThread	161
6.62.2	Member Function Documentation	161
6.62.2.1	setInitPriority	161
6.62.2.2	setInitStackSize	161
6.62.2.3	start	161
6.62.2.4	result	161
6.62.2.5	running	161
6.62.2.6	setPriority	161
6.62.2.7	cancel	161
6.62.2.8	waitForCompletion	161
6.62.2.9	getThread	161
6.62.2.10	function	161
6.62.2.11	startFunc	161
6.62.3	Member Data Documentation	161
6.62.3.1	othread	161
6.62.3.2	ostackSize	161
6.62.3.3	opolicy	161
6.62.3.4	opriority	161
6.62.3.5	orunning	161
6.62.3.6	oresult	161
6.63	BTimer Class Reference	162
6.63.1	Detailed Description	162
6.63.2	Constructor & Destructor Documentation	163
6.63.2.1	BTimer	163
6.63.2.2	~BTimer	163
6.63.3	Member Function Documentation	163
6.63.3.1	start	163
6.63.3.2	stop	163
6.63.3.3	clear	163
6.63.3.4	getElapsedTime	163
6.63.3.5	add	163
6.63.3.6	average	163
6.63.3.7	peak	163
6.63.3.8	getTime	164
6.63.4	Member Data Documentation	164

6.63.4.1	oLock	164
6.63.4.2	oNum	164
6.63.4.3	oStartTime	164
6.63.4.4	oEndTime	164
6.63.4.5	oAverage	164
6.63.4.6	oPeak	164
6.64	BTimeStamp Class Reference	165
6.64.1	Constructor & Destructor Documentation	167
6.64.1.1	BTimeStamp	167
6.64.1.2	BTimeStamp	167
6.64.1.3	BTimeStamp	167
6.64.1.4	~BTimeStamp	167
6.64.2	Member Function Documentation	167
6.64.2.1	clear	167
6.64.2.2	setFirst	167
6.64.2.3	setLast	167
6.64.2.4	set	167
6.64.2.5	set	167
6.64.2.6	set	167
6.64.2.7	setYDay	168
6.64.2.8	setTime	168
6.64.2.9	setNow	168
6.64.2.10	year	168
6.64.2.11	yday	168
6.64.2.12	month	168
6.64.2.13	day	168
6.64.2.14	hour	168
6.64.2.15	minute	168
6.64.2.16	second	168
6.64.2.17	microSecond	168
6.64.2.18	getDate	168
6.64.2.19	getString	168
6.64.2.20	setString	168
6.64.2.21	getStringNoMs	168
6.64.2.22	getStringFormatted	168
6.64.2.23	addMilliseconds	168

6.64.2.24	addMicroSeconds	169
6.64.2.25	addSeconds	169
6.64.2.26	getYearSeconds	169
6.64.2.27	getYearMicroSeconds	169
6.64.2.28	isSet	169
6.64.2.29	compare	169
6.64.2.30	operator BString	169
6.64.2.31	operator=	169
6.64.2.32	operator==	169
6.64.2.33	operator"!="	169
6.64.2.34	operator>	169
6.64.2.35	operator>=	169
6.64.2.36	operator<	169
6.64.2.37	operator<=	169
6.64.2.38	isLeap	169
6.64.2.39	difference	169
6.64.3	Member Data Documentation	169
6.64.3.1	oyear	169
6.64.3.2	oyday	170
6.64.3.3	ohour	170
6.64.3.4	ominute	170
6.64.3.5	osecond	170
6.64.3.6	ospare	170
6.64.3.7	omicroSecond	170
6.65	BTimeStampMs Class Reference	171
6.65.1	Constructor & Destructor Documentation	173
6.65.1.1	BTimeStampMs	173
6.65.1.2	~BTimeStampMs	173
6.65.2	Member Function Documentation	173
6.65.2.1	clear	173
6.65.2.2	setNow	173
6.65.2.3	addMilliseconds	173
6.65.2.4	subMilliseconds	173
6.65.2.5	addSeconds	173
6.65.2.6	subSeconds	173
6.65.2.7	getYearSeconds	173

6.65.2.8	getYearMilliseconds	173
6.65.2.9	getString	173
6.65.2.10	getStringNoMs	173
6.65.2.11	setString	174
6.65.2.12	getDurationString	174
6.65.2.13	getDurationStringNoMs	174
6.65.2.14	setDurationString	174
6.65.2.15	getStringRaw	174
6.65.2.16	getDate	174
6.65.2.17	compare	174
6.65.2.18	operator>	174
6.65.2.19	operator>=	174
6.65.2.20	operator<	174
6.65.2.21	operator<=	174
6.65.2.22	isLeap	174
6.65.2.23	difference	174
6.65.3	Member Data Documentation	174
6.65.3.1	year	174
6.65.3.2	yday	174
6.65.3.3	hour	175
6.65.3.4	minute	175
6.65.3.5	second	175
6.65.3.6	milliSecond	175
6.65.3.7	sampleNumber	175
6.66	BUrl Class Reference	176
6.66.1	Detailed Description	176
6.66.2	Constructor & Destructor Documentation	176
6.66.2.1	BUrl	176
6.66.2.2	~BUrl	176
6.66.3	Member Function Documentation	176
6.66.3.1	readString	176
6.66.3.2	writeData	177
6.66.4	Member Data Documentation	177
6.66.4.1	oinit	177
6.66.4.2	ores	177
6.67	vector Class Reference	178

7	File Documentation	179
7.1	BArray.h File Reference	179
7.1.1	Define Documentation	179
7.1.1.1	BArray_H	179
7.2	BAtomicCount.h File Reference	180
7.2.1	Define Documentation	180
7.2.1.1	BAtomicCount_H	180
7.3	BBuffer.cpp File Reference	181
7.3.1	Variable Documentation	181
7.3.1.1	roundSize	181
7.4	BBuffer.h File Reference	182
7.4.1	Define Documentation	182
7.4.1.1	BBUFFER_H	182
7.5	BCond.cpp File Reference	183
7.6	BCond.h File Reference	184
7.6.1	Define Documentation	184
7.6.1.1	BCOND_H	184
7.7	BCondInt.cpp File Reference	185
7.8	BCondInt.h File Reference	186
7.8.1	Define Documentation	186
7.8.1.1	BCONDINT_H	186
7.9	BConfig.cpp File Reference	187
7.10	BConfig.h File Reference	188
7.11	BDebug.cpp File Reference	189
7.11.1	Define Documentation	190
7.11.1.1	BTRACE_SIZE	190
7.11.2	Function Documentation	190
7.11.2.1	gettid	190
7.11.2.2	getTime	190
7.11.2.3	hd32	190
7.11.2.4	hd8	190
7.11.2.5	hda32	190
7.11.2.6	hda8	190
7.11.2.7	setDebug	190
7.11.2.8	tprintf	190
7.11.3	Variable Documentation	190

7.11.3.1	bdebug	190
7.11.3.2	STRBUF_SIZE	190
7.12	BDebug.h File Reference	191
7.12.1	Define Documentation	191
7.12.1.1	BDebug_STD	191
7.12.1.2	dprintf	191
7.12.1.3	eprintf	192
7.12.1.4	nprintf	192
7.12.1.5	wprintf	192
7.12.2	Function Documentation	192
7.12.2.1	gettid	192
7.12.2.2	getTime	192
7.12.2.3	hd32	192
7.12.2.4	hd8	192
7.12.2.5	hda8	192
7.12.2.6	hds32	192
7.12.2.7	setDebug	192
7.12.2.8	tprintf	192
7.12.3	Variable Documentation	192
7.12.3.1	bdebug	192
7.13	BDict.cpp File Reference	193
7.13.1	Function Documentation	193
7.13.1.1	fromBString	193
7.13.1.2	toBString	193
7.14	BDict.h File Reference	194
7.14.1	Define Documentation	194
7.14.1.1	BDict_H	194
7.14.2	Typedef Documentation	194
7.14.2.1	BDictString	194
7.14.3	Function Documentation	194
7.14.3.1	fromBString	194
7.14.3.2	toBString	194
7.15	BDictMap.h File Reference	195
7.15.1	Define Documentation	195
7.15.1.1	BDictMap_H	195
7.15.2	Typedef Documentation	195

7.15.2.1	BDictMapString	195
7.16	BDir.cpp File Reference	196
7.16.1	Function Documentation	196
7.16.1.1	wild	196
7.16.2	Variable Documentation	196
7.16.2.1	wildString	196
7.17	BDir.h File Reference	197
7.17.1	Define Documentation	197
7.17.1.1	BDIR_H	197
7.18	BEntry.cpp File Reference	198
7.19	BEntry.h File Reference	199
7.20	BError.cpp File Reference	200
7.21	BError.h File Reference	201
7.21.1	Define Documentation	201
7.21.1.1	BERROR_H	201
7.22	BEvent.cpp File Reference	202
7.23	BEvent.h File Reference	203
7.23.1	Define Documentation	203
7.23.1.1	BEvent_H	203
7.23.2	Enumeration Type Documentation	203
7.23.2.1	BEventType	203
7.24	BFifo.cpp File Reference	204
7.24.1	Define Documentation	204
7.24.1.1	DEBUG	204
7.24.1.2	dprintf	204
7.25	BFifo.h File Reference	205
7.25.1	Define Documentation	205
7.25.1.1	BFIFO_H	205
7.26	BFifo.inc File Reference	206
7.27	BFile.cpp File Reference	207
7.27.1	Define Documentation	207
7.27.1.1	STRBUF	207
7.28	BFile.h File Reference	208
7.28.1	Define Documentation	208
7.28.1.1	BFILE_H	208
7.29	BList.h File Reference	209

7.29.1	Define Documentation	209
7.29.1.1	BLIST_H	209
7.29.1.2	BListLoop	209
7.30	BList_func.h File Reference	210
7.31	BMutex.cpp File Reference	211
7.31.1	Define Documentation	211
7.31.1.1	MDEBUG	211
7.32	BMutex.h File Reference	212
7.32.1	Define Documentation	212
7.32.1.1	BMUTEX_H	212
7.33	BMySQL.cpp File Reference	213
7.34	BMySQL.h File Reference	214
7.34.1	Define Documentation	214
7.34.1.1	BMySQL_H	214
7.35	BNameValue.h File Reference	215
7.35.1	Define Documentation	215
7.35.1.1	BNAMEVALUE_H	215
7.36	Boap.cpp File Reference	216
7.36.1	Define Documentation	216
7.36.1.1	APIVERSION_TEST	216
7.36.1.2	DEBUG	216
7.36.1.3	dprintf	216
7.36.1.4	IS_BIG_ENDIAN	216
7.36.2	Variable Documentation	216
7.36.2.1	boapPort	216
7.37	Boap.h File Reference	217
7.37.1	Typedef Documentation	218
7.37.1.1	BoapFunc	218
7.37.1.2	BoapService	218
7.37.2	Enumeration Type Documentation	218
7.37.2.1	BoapPriority	218
7.37.2.2	BoapType	218
7.37.3	Variable Documentation	218
7.37.3.1	BoapMagic	218
7.38	BoapnsC.cc File Reference	219
7.39	BoapnsC.h File Reference	220

7.39.1	Define Documentation	220
7.39.1.1	BOAPNSC_H	220
7.40	BoapnsD.cc File Reference	221
7.41	BoapnsD.h File Reference	222
7.41.1	Define Documentation	222
7.41.1.1	BOAPNSD_H	222
7.42	BoapSimple.cc File Reference	223
7.42.1	Define Documentation	223
7.42.1.1	DEBUG	223
7.42.1.2	dprintf	223
7.42.2	Variable Documentation	223
7.42.2.1	roundSize	223
7.43	BoapSimple.h File Reference	224
7.43.1	Typedef Documentation	225
7.43.1.1	BoapFunc	225
7.43.1.2	BoapService	225
7.43.1.3	Double	225
7.43.1.4	Int16	225
7.43.1.5	Int32	225
7.43.1.6	Int8	225
7.43.1.7	UInt16	225
7.43.1.8	UInt32	225
7.43.1.9	UInt8	225
7.43.2	Enumeration Type Documentation	225
7.43.2.1	BoapType	225
7.44	BObj.cpp File Reference	226
7.45	BObj.h File Reference	227
7.45.1	Define Documentation	227
7.45.1.1	BObj_H	227
7.46	BObject.cc File Reference	228
7.46.1	Define Documentation	228
7.46.1.1	DEBUG	228
7.47	BObject.h File Reference	229
7.47.1	Define Documentation	229
7.47.1.1	BOBJECT_H	229
7.47.2	Typedef Documentation	229

7.47.2.1	BMember	229
7.47.2.2	BMemberList	229
7.48	BPoll.cpp File Reference	230
7.49	BPoll.h File Reference	231
7.49.1	Define Documentation	231
7.49.1.1	B POLL_H	231
7.50	BRefData.cpp File Reference	232
7.50.1	Define Documentation	232
7.50.1.1	CHUNK	232
7.50.1.2	DEBUG	232
7.51	BRefData.h File Reference	233
7.51.1	Define Documentation	233
7.51.1.1	BREFDATA_H	233
7.52	BRtc.cpp File Reference	234
7.53	BRtc.h File Reference	235
7.54	BRWLock.cpp File Reference	236
7.55	BRWLock.h File Reference	237
7.55.1	Define Documentation	237
7.55.1.1	BRWLOCK_H	237
7.56	BSema.cpp File Reference	238
7.57	BSema.h File Reference	239
7.57.1	Define Documentation	239
7.57.1.1	BSEMA_H	239
7.58	BSocket.cpp File Reference	240
7.58.1	Define Documentation	240
7.58.1.1	IP_MTU	240
7.59	BSocket.h File Reference	241
7.59.1	Define Documentation	241
7.59.1.1	BSOCKET_H	241
7.60	BString.cpp File Reference	242
7.60.1	Define Documentation	244
7.60.1.1	DEBUG	244
7.60.1.2	MINUS	244
7.60.1.3	STRIP	244
7.60.2	Function Documentation	244
7.60.2.1	barrayToString	244

7.60.2.2	blistToString	244
7.60.2.3	bstringListinList	244
7.60.2.4	bstringToArray	244
7.60.2.5	bstringToList	244
7.60.2.6	charToArray	244
7.60.2.7	charToList	244
7.60.2.8	fromBString	244
7.60.2.9	fromBString	244
7.60.2.10	fromBString	244
7.60.2.11	fromBString	244
7.60.2.12	fromBString	244
7.60.2.13	fromBString	244
7.60.2.14	gmatch	244
7.60.2.15	operator<<	244
7.60.2.16	operator>>	244
7.60.2.17	toBString	244
7.60.2.18	toBString	244
7.60.2.19	toBString	244
7.60.2.20	toBString	244
7.60.2.21	toBString	244
7.60.2.22	toBString	244
7.61	BString.h File Reference	245
7.61.1	Define Documentation	246
7.61.1.1	BSTRING_H	246
7.61.2	Function Documentation	246
7.61.2.1	fromBString	246
7.61.2.2	fromBString	246
7.61.2.3	fromBString	246
7.61.2.4	fromBString	246
7.61.2.5	fromBString	246
7.61.2.6	fromBString	246
7.61.2.7	operator<<	246
7.61.2.8	operator>>	246
7.61.2.9	toBString	246
7.61.2.10	toBString	246
7.61.2.11	toBString	246

7.61.2.12	toBString	246
7.61.2.13	toBString	246
7.61.2.14	toBString	246
7.62	BStringLocked.h File Reference	247
7.62.1	Define Documentation	247
7.62.1.1	BStringLocked_H	247
7.63	BTable.cpp File Reference	248
7.64	BTable.h File Reference	249
7.65	BThread.cpp File Reference	250
7.66	BThread.h File Reference	251
7.66.1	Define Documentation	251
7.66.1.1	BTHREAD_H	251
7.67	BTimer.cpp File Reference	252
7.68	BTimer.h File Reference	253
7.69	BTimeStamp.cpp File Reference	254
7.69.1	Function Documentation	254
7.69.1.1	fromBString	254
7.69.1.2	toBString	254
7.69.2	Variable Documentation	254
7.69.2.1	mon_yday	254
7.70	BTimeStamp.h File Reference	255
7.70.1	Define Documentation	255
7.70.1.1	BTimeStamp_H	255
7.70.2	Function Documentation	255
7.70.2.1	fromBString	255
7.70.2.2	toBString	255
7.71	BTimeStampMs.cpp File Reference	256
7.71.1	Variable Documentation	256
7.71.1.1	mon_yday	256
7.72	BTimeStampMs.h File Reference	257
7.72.1	Define Documentation	257
7.72.1.1	BTimeStampMs_H	257
7.73	BTypes.h File Reference	258
7.73.1	Define Documentation	259
7.73.1.1	BTYPES_H	259
7.73.2	Typedef Documentation	259

7.73.2.1	BArrayDouble	259
7.73.2.2	BArrayFloat	259
7.73.2.3	BDouble	259
7.73.2.4	BFloat	259
7.73.2.5	BFloat32	259
7.73.2.6	BFloat64	259
7.73.2.7	BInt	259
7.73.2.8	BInt16	259
7.73.2.9	BInt32	259
7.73.2.10	BInt64	259
7.73.2.11	BInt8	259
7.73.2.12	BSize	259
7.73.2.13	BUInt	259
7.73.2.14	BUInt16	259
7.73.2.15	BUInt32	259
7.73.2.16	BUInt64	259
7.73.2.17	BUInt8	259
7.73.3	Function Documentation	259
7.73.3.1	byteSwap16	259
7.73.3.2	byteSwap32	259
7.73.3.3	byteSwap64	259
7.73.3.4	byteSwap8	259
7.74	BUrl.cpp File Reference	260
7.75	BUrl.h File Reference	261
7.75.1	Define Documentation	261
7.75.1.1	BURL_H	261

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Boapns	9
------------------------------	---

Chapter 2

Class Index

2.1 Class Hierarchy

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

BArray< T >	11
BAtomicCount	13
BBuffer	15
BBufferStore	17
BoapPacket	96
BCond	21
BCondBool	22
BCondInt	24
BCondValue	27
BCondWrap	30
BDebugBacktrace	35
BDictItem< Type >	38
BDictMap< Value >	39
BEntry	44
BError	52
BEventError	57
BEvent	55
BEventError	57
BEventInt	58
BEventPipe	60
BFifo< Type >	62
BFifoPos	66
BFile	68
BIter	72
BList< T >	73
BDir	41
BList< BDictItem< BString > >	73
BDict< BString >	36
BConfig	33
BList< BDictItem< Type > >	73
BDict< Type >	36

BList< BEntry >	73
BEntryList	49
BEntryFile	47
BList< BNameValue< T > >	73
BNameValueList< T >	87
BList< dirent * >	73
BMutex	81
BStringMutex	157
BMutexLock	83
BMysql	84
BNameValue< T >	86
BNode	88
BList< T >::Node	80
Boapns::BoapEntry	93
BoapFuncEntry	94
BoapPacketHead	101
BoapServiceEntry	110
BoapServiceObject	111
BObj	117
BObject	118
BPoll	120
BRefData	123
BRtc	126
BRWLock	130
BSema	132
BSocket	134
BoapClientObject	89
Boapns::Boapns	95
BoapClientObject	89
BoapSignalObject	115
BoapSignalObject	115
BSocketAddress	139
BSocketAddressINET	141
BString	144
BStringLocked	155
BTable	158
BThread	160
BoapServer	102
BoapServerConnection	108
BRtcThreaded	128
BTimer	162
BTimeStamp	165
BTimeStampMs	171
BUrl	176
vector	178

Chapter 3

Class Index

3.1 Class List

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

BArray< T >	11
BAtomicCount (BAtomicCount class)	13
BBuffer	15
BBufferStore	17
BCond	21
BCondBool (Thread conditional boolean)	22
BCondInt (Thread conditional integer)	24
BCondValue (Thread conditional value)	27
BCondWrap	30
BConfig (This class implements the configuration file access)	33
BDebugBacktrace	35
BDict< Type >	36
BDictItem< Type > (Template based Dictionary class)	38
BDictMap< Value >	39
BDir (File system directory class)	41
BEntry (Manipulate a name value pair)	44
BEntryFile (File of Entries)	47
BEntryList (List of Entries. Where an entry is a name value pair)	49
BError (Error return class)	52
BEvent (This class provides a base class for all event objects that can be sent over the events interface)	55
BEventError	57
BEventInt (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)	58
BEventPipe (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)	60
BFifo< Type > (This class implements a thread safe FIFO buffer)	62
BFifoPos (This class implements a pointer into the Fifo's circular buffer)	66
BFile (File operations class)	68
BIter (Iterator for BList)	72
BList< T > (Template based list class)	73
BList< T >::Node	80
BMutex (Mutex class)	81

BMutexLock	83
BMySQL	84
BNameValue< T >	86
BNameValueList< T >	87
BNode	88
BoapClientObject	89
Boapns::BoapEntry	93
BoapFuncEntry	94
Boapns::Boapns	95
BoapPacket	96
BoapPacketHead	101
BoapServer	102
BoapServerConnection	108
BoapServiceEntry	110
BoapServiceObject	111
BoapSignalObject	115
BObj	117
BObject	118
BPoll (This class provides an interface for polling a number of file descriptors. It uses round robin polling)	120
BRefData	123
BRtc (Realtime clock)	126
BRtcThreaded (Threaded real time clock)	128
BRWLock (Thread read-write locks)	130
BSema (Semaphore class)	132
BSocket	134
BSocketAddress (Socket Address)	139
BSocketAddressINET (IP aware socket address)	141
BString	144
BStringLocked	155
BStringMutex	157
BTable	158
BThread	160
BTimer (Stopwatch style timer)	162
BTimeStamp	165
BTimeStampMs	171
BUrl (Basic access to a Url)	176
vector	178

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

BArray.h	179
BAtomicCount.h	180
BBuffer.cpp	181
BBuffer.h	182
BCond.cpp	183
BCond.h	184
BCondInt.cpp	185
BCondInt.h	186
BConfig.cpp	187
BConfig.h	188
BDebug.cpp	189
BDebug.h	191
BDict.cpp	193
BDict.h	194
BDictMap.h	195
BDir.cpp	196
BDir.h	197
BEntry.cpp	198
BEntry.h	199
BError.cpp	200
BError.h	201
BEvent.cpp	202
BEvent.h	203
BFifo.cpp	204
BFifo.h	205
BFifo.inc	206
BFile.cpp	207
BFile.h	208
BList.h	209
BList_func.h	210
BMutex.cpp	211
BMutex.h	212
BMysql.cpp	213

BMySQL.h	214
BNameValue.h	215
Boap.cpp	216
Boap.h	217
BoapnsC.cc	219
BoapnsC.h	220
BoapnsD.cc	221
BoapnsD.h	222
BoapSimple.cc	223
BoapSimple.h	224
BObj.cpp	226
BObj.h	227
BObject.cc	228
BObject.h	229
BPoll.cpp	230
BPoll.h	231
BRefData.cpp	232
BRefData.h	233
BRtc.cpp	234
BRtc.h	235
BRWLock.cpp	236
BRWLock.h	237
BSema.cpp	238
BSema.h	239
BSocket.cpp	240
BSocket.h	241
BString.cpp	242
BString.h	245
BStringLocked.h	247
BTable.cpp	248
BTable.h	249
BThread.cpp	250
BThread.h	251
BTimer.cpp	252
BTimer.h	253
BTimeStamp.cpp	254
BTimeStamp.h	255
BTimeStampMs.cpp	256
BTimeStampMs.h	257
BTypes.h	258
BUrl.cpp	260
BUrl.h	261

Chapter 5

Namespace Documentation

5.1 Boapns Namespace Reference

Classes

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

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)

Variables

- const [BUInt32 apiVersion](#) = 0

5.1.1 Function Documentation

5.1.1.1 BError Boapns::addEntry (BoapEntry *entry*)

5.1.1.2 Boapns::Boapns (BString *name*)

5.1.1.3 BError Boapns::delEntry (BString *name*)

5.1.1.4 BError Boapns::getEntry (BString *name*, BoapEntry & *entry*)

5.1.1.5 BError Boapns::getEntryList (BList< BoapEntry > & *entryList*)

5.1.1.6 BError Boapns::getNewName (BString & *name*)

5.1.1.7 BError Boapns::getVersion (BString & *version*)

5.1.2 Variable Documentation

5.1.2.1 const BUInt32 Boapns::apiVersion = 0

Chapter 6

Class Documentation

6.1 BArray< T > Class Template Reference

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

Public Member Functions

- [BArray \(\)](#)
- [BArray \(BSize size, T value=T\(\)\)](#)
- [BArray \(const BArray &array\)](#)
- [void append \(T value\)](#)
- [void append \(const BArray< T > &array\)](#)

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 Constructor & Destructor Documentation

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

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

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

6.1.3 Member Function Documentation

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

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

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

- [BArray.h](#)

6.2 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](#)

6.2.1 Detailed Description

[BAtomicCount](#) class.

6.2.2 Constructor & Destructor Documentation

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

6.2.3 Member Function Documentation

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

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

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

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

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

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

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

6.2.4 Member Data Documentation

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

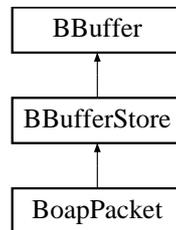
The documentation for this class was generated from the following file:

- [BAtomicCount.h](#)

6.3 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](#)

6.3.1 Constructor & Destructor Documentation

6.3.1.1 BBuffer::BBuffer (BUInt size = 0)

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

6.3.1.2 BBuffer::~~BBuffer ()

6.3.2 Member Function Documentation

6.3.2.1 int BBuffer::setSize (BUInt32 *size*)

Sets the bufer size.

6.3.2.2 int BBuffer::setData (const void * *data*, BUInt32 *size*)

Sets buffer data resized to contain the data.

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

Writes data into buffer from offset pos.

6.3.2.4 char * BBuffer::data ()

The data.

Reimplemented in [BoapPacket](#).

6.3.2.5 BUInt32 BBuffer::size ()

Size of the buffer in bytes.

6.3.2.6 int BBuffer::resize (BUInt32 *size*) [inline]

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

6.3.3 Member Data Documentation

6.3.3.1 BUInt32 BBuffer::odataSize [protected]

6.3.3.2 char* BBuffer::odata [protected]

Reimplemented in [BoapPacket](#).

6.3.3.3 BUInt32 BBuffer::osize [protected]

Reimplemented in [BoapPacket](#).

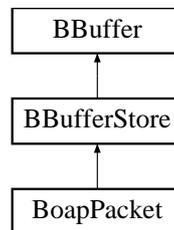
The documentation for this class was generated from the following files:

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

6.4 BBufferStore Class Reference

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

Inheritance diagram for BBufferStore::



Public Member Functions

- [BBufferStore \(\)](#)
- [~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 \(BUInt32 nBytes, const void *data, char *swapType="1"\)](#)
- [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 \(BUInt32 nBytes, void *data, char *swapType="1"\)](#)

Protected Member Functions

- void [copyWithSwap](#) (void *dst, const void *src, [BUInt32](#) nBytes, char *swapType)

Protected Attributes

- [BUInt32](#) opos

6.4.1 Constructor & Destructor Documentation

6.4.1.1 [BBufferStore::BBufferStore \(\)](#)

6.4.1.2 [BBufferStore::~~BBufferStore \(\)](#)

6.4.2 Member Function Documentation

6.4.2.1 [BUInt32 BBufferStore::getPos \(\)](#)

6.4.2.2 [void BBufferStore::setPos \(BUInt32 pos\)](#)

6.4.2.3 [BString BBufferStore::getHexString \(\)](#)

6.4.2.4 [void BBufferStore::setHexString \(BString s\)](#)

6.4.2.5 [int BBufferStore::push \(BInt8 v\)](#)

Reimplemented in [BoapPacket](#).

6.4.2.6 [int BBufferStore::push \(BUInt8 v\)](#)

Reimplemented in [BoapPacket](#).

6.4.2.7 [int BBufferStore::push \(BInt16 v\)](#)

Reimplemented in [BoapPacket](#).

6.4.2.8 [int BBufferStore::push \(BUInt16 v\)](#)

Reimplemented in [BoapPacket](#).

6.4.2.9 [int BBufferStore::push \(BInt32 v\)](#)

Reimplemented in [BoapPacket](#).

6.4.2.10 [int BBufferStore::push \(BUInt32 v\)](#)

Reimplemented in [BoapPacket](#).

6.4.2.11 int BBufferStore::push (BInt64 *v*)

6.4.2.12 int BBufferStore::push (BUInt64 *v*)

6.4.2.13 int BBufferStore::push (BFloat32 *v*)

6.4.2.14 int BBufferStore::push (BFloat64 *v*)

Reimplemented in [BoapPacket](#).

6.4.2.15 int BBufferStore::push (const BString & *v*)

6.4.2.16 int BBufferStore::push (const BError & *v*)

6.4.2.17 int BBufferStore::push (const BTimeStamp & *v*)

6.4.2.18 int BBufferStore::push (BUInt32 *nBytes*, const void * *data*, char * *swapType* = "1")

6.4.2.19 int BBufferStore::pop (BInt8 & *v*)

Reimplemented in [BoapPacket](#).

6.4.2.20 int BBufferStore::pop (BUInt8 & *v*)

Reimplemented in [BoapPacket](#).

6.4.2.21 int BBufferStore::pop (BInt16 & *v*)

Reimplemented in [BoapPacket](#).

6.4.2.22 int BBufferStore::pop (BUInt16 & *v*)

Reimplemented in [BoapPacket](#).

6.4.2.23 int BBufferStore::pop (BInt32 & *v*)

Reimplemented in [BoapPacket](#).

6.4.2.24 int BBufferStore::pop (BUInt32 & *v*)

Reimplemented in [BoapPacket](#).

6.4.2.25 `int BBufferStore::pop (BInt64 & v)`

6.4.2.26 `int BBufferStore::pop (BUInt64 & v)`

6.4.2.27 `int BBufferStore::pop (BFloat32 & v)`

6.4.2.28 `int BBufferStore::pop (BFloat64 & v)`

Reimplemented in [BoapPacket](#).

6.4.2.29 `int BBufferStore::pop (BString & v)`

Reimplemented in [BoapPacket](#).

6.4.2.30 `int BBufferStore::pop (BError & v)`

Reimplemented in [BoapPacket](#).

6.4.2.31 `int BBufferStore::pop (BTimeStamp & v)`

6.4.2.32 `int BBufferStore::pop (BUInt32 nBytes, void * data, char * swapType = "1")`

6.4.2.33 `void BBufferStore::copyWithSwap (void * dst, const void * src, BUInt32 nBytes, char * swapType) [protected]`

6.4.3 Member Data Documentation

6.4.3.1 `BUInt32 BBufferStore::opos [protected]`

Reimplemented in [BoapPacket](#).

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

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

6.5 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](#)

6.5.1 Constructor & Destructor Documentation

6.5.1.1 BCond::BCond ()

Thread conditional variable.

6.5.1.2 BCond::~~BCond ()

6.5.2 Member Function Documentation

6.5.2.1 int BCond::signal ()

6.5.2.2 int BCond::wait ()

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

6.5.3 Member Data Documentation

6.5.3.1 pthread_mutex_t BCond::omutex [private]

6.5.3.2 pthread_cond_t BCond::ocond [private]

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

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

6.6 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](#)

6.6.1 Detailed Description

Thread conditional boolean.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 BCondBool::BCondBool ()

6.6.2.2 BCondBool::~~BCondBool ()

6.6.3 Member Function Documentation

6.6.3.1 int BCondBool::set ()

Set value. Wakes waiting.

6.6.3.2 int BCondBool::clear ()

Clear Value.

6.6.3.3 int BCondBool::value ()

Current value.

6.6.3.4 int BCondBool::wait ()

Wait until value is true.

6.6.3.5 int BCondBool::timedWait (int *timeOutUs*)

Wait until set, with timeout.

6.6.3.6 BCondBool::operator int () [inline]

6.6.4 Member Data Documentation

6.6.4.1 pthread_mutex_t BCondBool::omutex [private]

6.6.4.2 pthread_cond_t BCondBool::ocond [private]

6.6.4.3 int BCondBool::ovalue [private]

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

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

6.7 BCondInt Class Reference

Thread conditional integer.

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

Public Member Functions

- [BCondInt \(\)](#)
- [~BCondInt \(\)](#)
- void [setValue](#) (int value)
Set value.
- int [increment](#) ()
Increment.
- int [decrement](#) ()
Decrement.
- int [value](#) ()
Current value.
- int [wait](#) ()
Wait until value is 0.
- int [waitIncrement](#) (int timeOutUs=0)
Wait until value is 0 then increment.
- int [waitNotZero](#) ()
Wait until value is not 0.
- int [waitNotZeroDecrement](#) ()
Wait until value is not 0 and then decrement.
- int [tryNotZeroDecrement](#) ()
Test if value is not 0, if not zero then decrement.
- int [timedWait](#) (int timeOutUs)
Wait for the condition, with timeout.
- void [operator++](#) (int)
- void [operator--](#) (int)

Private Attributes

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

6.7.1 Detailed Description

Thread conditional integer.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 BCondInt::BCondInt ()

6.7.2.2 BCondInt::~~BCondInt ()

6.7.3 Member Function Documentation

6.7.3.1 void BCondInt::setValue (int *value*)

Set value.

6.7.3.2 int BCondInt::increment ()

Increment.

6.7.3.3 int BCondInt::decrement ()

Decrement.

6.7.3.4 int BCondInt::value ()

Current value.

6.7.3.5 int BCondInt::wait ()

Wait until value is 0.

6.7.3.6 int BCondInt::waitIncrement (int *timeOutUs* = 0)

Wait until value is 0 then increment.

6.7.3.7 int BCondInt::waitNotZero ()

Wait until value is not 0.

6.7.3.8 int BCondInt::waitNotZeroDecrement ()

Wait until value is not 0 and then decrement.

6.7.3.9 int BCondInt::tryNotZeroDecrement ()

Test if value is not 0, if not zero then decrement.

6.7.3.10 `int BCondInt::timedWait (int timeOutUs)`

Wait for the condition, with timeout.

6.7.3.11 `void BCondInt::operator++ (int) [inline]`**6.7.3.12** `void BCondInt::operator- (int) [inline]`**6.7.4 Member Data Documentation****6.7.4.1** `pthread_mutex_t BCondInt::omutex [private]`**6.7.4.2** `pthread_cond_t BCondInt::ocond [private]`**6.7.4.3** `int BCondInt::ovalue [private]`

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

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

6.8 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++](#) (int)
Increment value. Wakes waiting.
- void [operator--](#) (int)
Decrement value. Wakes waiting.

Private Attributes

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

6.8.1 Detailed Description

Thread conditional value.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 BCondValue::BCondValue ()

6.8.2.2 BCondValue::~~BCondValue ()

6.8.3 Member Function Documentation

6.8.3.1 void BCondValue::setValue (int *value*)

Set the value. Wakes waiting.

6.8.3.2 int BCondValue::value ()

Current value.

6.8.3.3 int BCondValue::increment (int *v* = 1)

Increment. Wakes waiting.

6.8.3.4 int BCondValue::decrement (int *v* = 1)

Decrement. Wakes waiting.

6.8.3.5 int BCondValue::waitMoreThanOrEqual (int *v*, int *decrement* = 0, int *timeOutUs* = 0)

Wait until value is at least the value given.

6.8.3.6 int BCondValue::waitLessThanOrEqual (int *v*, int *increment* = 0, int *timeOutUs* = 0)

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

6.8.3.7 int BCondValue::waitLessThan (int *v*, int *timeOutUs* = 0)

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

6.8.3.8 void BCondValue::operator+= (int *v*) [inline]

Add to value. Wakes waiting.

6.8.3.9 void BCondValue::operator-= (int *v*) [inline]

Subtract from value. Wakes waiting.

6.8.3.10 void BCondValue::operator++ (int) [inline]

Increment value. Wakes waiting.

6.8.3.11 void BCondValue::operator-- (int) [inline]

Decrement value. Wakes waiting.

6.8.4 Member Data Documentation**6.8.4.1 pthread_mutex_t BCondValue::omutex [private]****6.8.4.2 pthread_cond_t BCondValue::ocond [private]****6.8.4.3 int BCondValue::ovalue [private]**

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

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

6.9 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](#)

6.9.1 Constructor & Destructor Documentation

6.9.1.1 BCondWrap::BCondWrap ()

6.9.1.2 BCondWrap::~~BCondWrap ()

6.9.2 Member Function Documentation

6.9.2.1 void BCondWrap::setValue (uint32_t *value*)

Set the value. Wakes waiting.

6.9.2.2 uint32_t BCondWrap::value ()

Current value.

6.9.2.3 uint32_t BCondWrap::increment (uint32_t *v* = 1)

Increment. Wakes waiting.

6.9.2.4 uint32_t BCondWrap::decrement (uint32_t *v* = 1)

Decrement. Wakes waiting.

6.9.2.5 int BCondWrap::waitMoreThanOrEqual (uint32_t *v*, uint32_t *decrement* = 0, uint32_t *timeOutUs* = 0)

Wait until value is at least the value given.

6.9.2.6 int BCondWrap::waitLessThanOrEqual (uint32_t *v*, uint32_t *increment* = 0, uint32_t *timeOutUs* = 0)

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

6.9.2.7 int BCondWrap::waitLessThan (uint32_t *v*, uint32_t *timeOutUs* = 0)

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

6.9.2.8 void BCondWrap::operator+= (int *v*) [inline]

Add to value. Wakes waiting.

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

Subtract from value. Wakes waiting.

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

Increment value. Wakes waiting.

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

Decrement value. Wakes waiting.

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

6.9.3 Member Data Documentation

6.9.3.1 `pthread_mutex_t BCondWrap::omutex` [private]

6.9.3.2 `pthread_cond_t BCondWrap::ocond` [private]

6.9.3.3 `uint32_t BCondWrap::ovalue` [private]

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

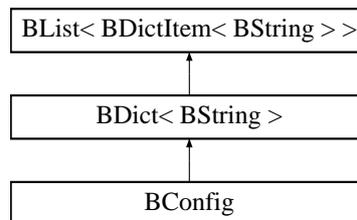
- [BCondInt.h](#)
- [BCondInt.cpp](#)

6.10 BConfig Class Reference

This class implements the configuration file access.

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

Inheritance diagram for BConfig::



Public Member Functions

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

Private Attributes

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

6.10.1 Detailed Description

This class implements the configuration file access.

6.10.2 Member Function Documentation

6.10.2.1 [BError BConfig::open \(BString *fileName*, BString *mode* = "r"\)](#)

6.10.2.2 [BError BConfig::read \(\)](#)

6.10.2.3 [BError BConfig::write \(\)](#)

6.10.2.4 [BString BConfig::findValue \(BString *name*\)](#)

6.10.3 Member Data Documentation

6.10.3.1 [BMutex BConfig::olock](#) [private]

6.10.3.2 [BFile BConfig::ofile](#) [private]

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

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

6.11 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)

6.11.1 Constructor & Destructor Documentation

6.11.1.1 [BDebugBacktrace::BDebugBacktrace \(\)](#)

6.11.1.2 [BDebugBacktrace::~~BDebugBacktrace \(\)](#)

6.11.2 Member Function Documentation

6.11.2.1 void [BDebugBacktrace::dumpBacktraceStdout](#) (char * *comment*)

6.11.2.2 int [BDebugBacktrace::dumpBacktraceFile](#) (char * *fileName*, char * *comment*)

6.11.2.3 void [BDebugBacktrace::dumpBacktraceSyslog](#) (char * *comment*)

6.11.2.4 void [BDebugBacktrace::dumpBacktrace](#) (char * *strBuf*, int *strBufLen*, char * *comment*)

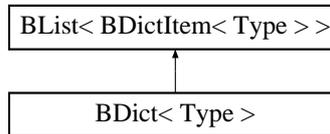
The documentation for this class was generated from the following files:

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

6.12 BDict< Type > Class Template Reference

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

Inheritance diagram for BDict< Type >::



Public Types

- typedef [BIter](#) iterator

Public Member Functions

- int [hasKey](#) (const [BString](#) &k) const
- [BString](#) [key](#) (iterator &i) const
- void [del](#) (const [BString](#) &k)
- [BIter](#) [find](#) (const [BString](#) &k) const
- Type & [operator\[\]](#) (const [BString](#) &i)
- Type & [operator\[\]](#) (const [BIter](#) &i)
- const Type & [operator\[\]](#) (const [BIter](#) &i) const
- [BDict](#)< Type > [operator+](#) (const [BDict](#)< Type > &dict) const
- [BDict](#)< Type > & [operator=](#) (const [BDict](#)< Type > &dict)

```
template<class Type> class BDict< Type >
```

6.12.1 Member Typedef Documentation

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

6.12.2 Member Function Documentation

6.12.2.1 `template<class Type> int BDict< Type >::hasKey (const BString & k) const` [inline]

6.12.2.2 `template<class Type> BString BDict< Type >::key (iterator & i) const` [inline]

6.12.2.3 `template<class Type> void BDict< Type >::del (const BString & k)` [inline]

6.12.2.4 `template<class Type> BIter BDict< Type >::find (const BString & k) const` [inline]

6.12.2.5]

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

6.12.2.6]

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

6.12.2.7]

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

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

6.12.2.8 `template<class Type> BDict< Type > BDict< Type >::operator+ (const BDict< Type > & dict) const [inline]`

6.12.2.9 `template<class Type> BDict< Type > & BDict< Type >::operator= (const BDict< Type > & dict) [inline]`

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

- [BDict.h](#)

6.13 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

6.13.1 Detailed Description

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

Template based Dictionary class.

6.13.2 Constructor & Destructor Documentation

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

6.13.3 Member Data Documentation

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

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

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

- [BDict.h](#)

6.14 BDictMap< Value > Class Template Reference

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

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)

6.14.1 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.

6.14.2 Member Typedef Documentation

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

6.14.3 Member Function Documentation

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

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

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

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

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

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

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

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

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

6.14.3.10]

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

6.14.3.11]

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

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

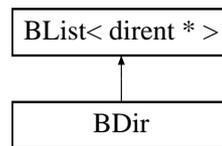
- [BDictMap.h](#)

6.15 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](#)

6.15.1 Detailed Description

File system directory class.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 `BDir::BDir ()`

6.15.2.2 `BDir::BDir (BString name)`

6.15.2.3 `BDir::~~BDir ()`

6.15.3 Member Function Documentation

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

Reads named directory.

6.15.3.2 `BError BDir::error ()`

Current value of error.

6.15.3.3 `BError BDir::read ()`

read/re-reads directory

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

Clears list.

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

6.15.3.5 `void BDir::setWild (BString wild)`

Set wildcard filter string used on read.

6.15.3.6 `void BDir::setSort (int on)`

Set alpha sort on/off.

6.15.3.7 BString BDir::entryName (BIter *i*)

Get filename.

6.15.3.8 struct stat BDir::entryStat (BIter *i*) [read]

Get file stats.

6.15.3.9 struct stat64 BDir::entryStat64 (BIter *i*) [read]

Get file stats 64.

6.15.4 Member Data Documentation

6.15.4.1 BError BDir::oerror [private]

6.15.4.2 BString BDir::odirname [private]

6.15.4.3 BString BDir::owild [private]

6.15.4.4 int BDir::osort [private]

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

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

6.16 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](#)

6.16.1 Detailed Description

Manipulate a name value pair.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 BEntry::BEntry ()

6.16.2.2 BEntry::BEntry (BString *name*, BString *value*)

Set name and value.

6.16.2.3 BEntry::BEntry (BString *line*)

Set name and value from white space delimited string.

6.16.3 Member Function Documentation

6.16.3.1 BString BEntry::getName ()

Get the name.

6.16.3.2 BString BEntry::getValue ()

Get the value.

6.16.3.3 void BEntry::setLine (BString *line*)

Set name and value from white space delimited string.

6.16.3.4 void BEntry::setName (BString *name*)

Set the name.

6.16.3.5 void BEntry::setValue (BString *value*)

Set the value.

6.16.3.6 BString BEntry::line ()

Return name and value as padded single string.

6.16.3.7 void BEntry::print ()

Print name and value.

6.16.4 Member Data Documentation

6.16.4.1 `BString BEntry::oname` [private]

6.16.4.2 `BString BEntry::ovalue` [private]

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

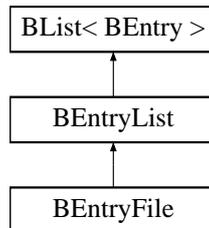
- [BEntry.h](#)
- [BEntry.cpp](#)

6.17 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.

Private Attributes

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

6.17.1 Detailed Description

File of Entries.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 BEntryFile::BEntryFile ()

6.17.2.2 BEntryFile::BEntryFile (BString *filename*)

Opens entryfile.

6.17.2.3 BEntryFile::~~BEntryFile ()

6.17.3 Member Function Documentation

6.17.3.1 int BEntryFile::open (BString *filename*)

Opens entryfile.

6.17.3.2 int BEntryFile::read ()

Reads entry file and builds list.

6.17.3.3 int BEntryFile::write ()

Writes list to entryfile.

6.17.3.4 int BEntryFile::writeList (BEntryList & *l*)

Writes specified list to file.

6.17.3.5 void BEntryFile::clear () [virtual]

Clears current list.

Reimplemented from [BEntryList](#).

6.17.4 Member Data Documentation

6.17.4.1 BString BEntryFile::ofilename [private]

6.17.4.2 BString BEntryFile::ocomments [private]

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

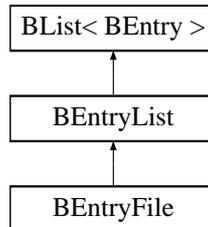
- [BEntry.h](#)
- [BEntry.cpp](#)

6.18 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`

6.18.1 Detailed Description

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

6.18.2 Constructor & Destructor Documentation

6.18.2.1 `BEntryList::BEntryList ()`

6.18.3 Member Function Documentation

6.18.3.1 `int BEntryList::isSet (BString name)`

1 if name is in list and value is set

6.18.3.2 `BEntry * BEntryList::find (BString name)`

Returns entry if name is found otherwise NULL.

6.18.3.3 `BString BEntryList::findValue (BString name)`

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

6.18.3.4 `int BEntryList::setValue (BString name, BString value)`

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

6.18.3.5 `int BEntryList::setValueRaw (BString name, BString value)`

Raw setting of value without looking up existing entry.

6.18.3.6 `void BEntryList::deleteEntry (BString name)`

Deletes the entry.

6.18.3.7 `void BEntryList::print ()`

Print list.

6.18.3.8 BString BEntryList::getString ()

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

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

Insert item before item.

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

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

Delete specified item.

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

6.18.3.11 void BEntryList::clear () [virtual]

Clear the list.

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

Reimplemented in [BEntryFile](#).

6.18.3.12 BEntryList & BEntryList::operator= (const BEntryList & *l*)

6.18.4 Member Data Documentation

6.18.4.1 BIter BEntryList::olastPos [private]

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

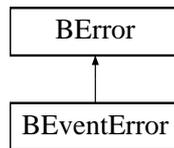
- [BEntry.h](#)
- [BEntry.cpp](#)

6.19 BError Class Reference

Error return class.

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

Inheritance diagram for BError::



Public Types

- enum `Type` { `NONE` = 0, `ERROR` = 1 }

Public Member Functions

- `BError` (int errNo=`NONE`, `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 getErrorNo () const
Get The error number.
- operator int () const
Return error number.

Private Attributes

- [int oerrNo](#)
- [BString oerrStr](#)

6.19.1 Detailed Description

Error return class.

6.19.2 Member Enumeration Documentation

6.19.2.1 enum BError::Type

Enumerator:

NONE

ERROR

6.19.3 Constructor & Destructor Documentation

6.19.3.1 BError::BError (int *errNo* = NONE, BString *errStr* = "")

Create object.

6.19.3.2 BError::BError (BString *errStr*)

Create with error set and error string.

6.19.4 Member Function Documentation

6.19.4.1 BError BError::copy ()

Return an independant copy.

6.19.4.2 BError & BError::set (int *errNo*, BString *errStr* = "")

Set error number and message.

6.19.4.3 BError & BError::clear ()

Clear the error.

6.19.4.4 BError & BError::setError (BString *errStr* = "")

Set error type ERROR with optional message.

6.19.4.5 BString BError::getString () const

Get error message.

6.19.4.6 int BError::getErrorNo () const

Get The error number.

6.19.4.7 BError::operator int () const

Return error number.

6.19.5 Member Data Documentation

6.19.5.1 int BError::oerrNo [private]

6.19.5.2 BString BError::oerrStr [private]

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

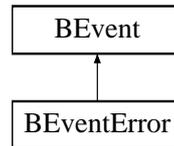
- [BError.h](#)
- [BError.cpp](#)

6.20 BEvent Class Reference

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

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

Inheritance diagram for BEvent::



Public Member Functions

- [BEvent](#) (uint32_t type)
- virtual [~BEvent](#) ()
- 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.

6.20.1 Detailed Description

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

6.20.2 Constructor & Destructor Documentation

6.20.2.1 [BEvent::BEvent](#) (uint32_t type)

6.20.2.2 [BEvent::~~BEvent](#) () [virtual]

6.20.3 Member Function Documentation

6.20.3.1 uint32_t [BEvent::getType](#) ()

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

Reimplemented in [BEventError](#).

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

Reimplemented in [BEventError](#).

6.20.4 Member Data Documentation

6.20.4.1 `uint32_t BEvent::otype` [private]

The event type.

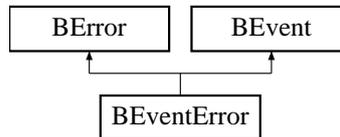
The documentation for this class was generated from the following files:

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

6.21 BEventError Class Reference

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

Inheritance diagram for BEventError::



Public Member Functions

- [BEventError](#) (int errNo=NONE, [BString](#) errStr="")
- [BError](#) [getBinary](#) (void *data, uint32_t &size)
- [BError](#) [setBinary](#) (void *data, uint32_t &size)

6.21.1 Constructor & Destructor Documentation

6.21.1.1 [BEventError::BEventError](#) (int *errNo* = NONE, [BString](#) *errStr* = "")

6.21.2 Member Function Documentation

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

Reimplemented from [BEvent](#).

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

Reimplemented from [BEvent](#).

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

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

6.22 BEventInt 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 <BEvent.h>
```

Public Member Functions

- [BEventInt \(\)](#)
- [~BEventInt \(\)](#)
- 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.

6.22.1 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.

6.22.2 Constructor & Destructor Documentation

6.22.2.1 BEventInt::BEventInt ()

6.22.2.2 BEventInt::~BEventInt ()

6.22.3 Member Function Documentation

6.22.3.1 void BEventInt::clear ()

Clear events pending.

6.22.3.2 BError BEventInt::sendEvent (int *event*)

Send an event.

6.22.3.3 BError BEventInt::getEvent (int & *event*, int *timeOutUs* = -1)

Receive the event.

6.22.3.4 int BEventInt::getFd ()

6.22.4 Member Data Documentation

6.22.4.1 int BEventInt::ofds[2] [private]

File descriptors for pipe.

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

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

6.23 BEventPipe 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 <BEvent.h>
```

Public Member Functions

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

Private Attributes

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

6.23.1 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.

6.23.2 Constructor & Destructor Documentation

6.23.2.1 [BEventPipe::BEventPipe \(\)](#)

6.23.2.2 [BEventPipe::~~BEventPipe \(\)](#)

6.23.3 Member Function Documentation

6.23.3.1 [void BEventPipe::clear \(\)](#)

Clear events pending.

6.23.3.2 BError BEventPipe::sendEvent (BEvent * event)

Send an event.

6.23.3.3 BError BEventPipe::getEvent (BEvent * event, int timeOutUs = -1)

Receive the event.

6.23.3.4 int BEventPipe::getReceiveFd ()

returns the receive file descriptor for the poll system call

6.23.4 Member Data Documentation

6.23.4.1 int BEventPipe::ofds[2] [private]

File descriptors for pipe.

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

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

6.24 BFifo< Type > Class Template Reference

This class implements a thread safe FIFO buffer.

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

Public Types

- enum { `defaultSize` = 1024 }

Public Member Functions

- `BFifo` (uint32_t size=defaultSize)
- `~BFifo` ()
- 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.

Private Member Functions

- [BError mapCircularBuffer](#) (uint32_t size)
- void [unmapCircularBuffer](#) ()

Private Attributes

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

6.24.1 Detailed Description

`template<class Type> class BFifo< Type >`

This class implements a thread safe FIFO buffer.

6.24.2 Member Enumeration Documentation

6.24.2.1 `template<class Type> anonymous enum`

Enumerator:

defaultSize

6.24.3 Constructor & Destructor Documentation

6.24.3.1 `template<class Type> BFifo< Type >::BFifo (uint32_t size = defaultSize)`

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

6.24.4 Member Function Documentation

6.24.4.1 `template<class Type> uint32_t BFifo< Type >::size ()`

Return the buffers actual size.

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

Clear all of the data in the buffer.

6.24.4.3 `template<class Type> uint32_t BFifo< Type >::writeAvailable ()`

Returns the space available to write.

6.24.4.4 `template<class Type> BError BFifo< Type >::writeWaitAvailable (uint32_t numFifoSamples)`

Wait for the given number of samples.

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

Writes the data to the buffer. Blocks until complete.

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

Return a pointer to the current start of the buffer.

6.24.4.7 `template<class Type> void BFifo< Type >::writeDone (uint32_t numFifoSamples)`

Update the write pointer.

6.24.4.8 `template<class Type> uint32_t BFifo< Type >::readAvailable ()`

Returns the number of bytes of data available.

6.24.4.9 `template<class Type> BError BFifo< Type >::readWaitAvailable (uint32_t numFifoSamples)`

Wait for given number of samples.

6.24.4.10 `template<class Type> BError BFifo< Type >::read (Type * data, uint32_t numFifoSamples)`**6.24.4.11** `template<class Type> Type* BFifo< Type >::readData ()`

Pointer to raw data.

6.24.4.12 `template<class Type> BError BFifo< Type >::readDone (uint32_t numFifoSamples)`

Updates read pointer.

6.24.4.13 `]`

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

Direct access to read samples in buffer.

6.24.4.14 `template<class Type> BError BFifo< Type >::mapCircularBuffer (uint32_t size)`
[private]

6.24.4.15 `template<class Type> void BFifo< Type >::unmapCircularBuffer ()` [private]

6.24.5 Member Data Documentation

6.24.5.1 `template<class Type> BMutex BFifo< Type >::olock` [private]

6.24.5.2 `template<class Type> uint32_t BFifo< Type >::ovmSize` [private]

6.24.5.3 `template<class Type> uint32_t BFifo< Type >::osize` [private]

6.24.5.4 `template<class Type> Type* BFifo< Type >::odata` [private]

6.24.5.5 `template<class Type> BFifoPos BFifo< Type >::owritePos` [private]

Current write position.

6.24.5.6 `template<class Type> BCondValue BFifo< Type >::owriteNumFifoSamples`
[private]

The number of samples in the FIFO.

6.24.5.7 `template<class Type> BFifoPos BFifo< Type >::oreadPos` [private]

Current read position.

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

- [BFifo.h](#)

6.25 BFifoPos Class Reference

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

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

Public Member Functions

- [BFifoPos](#) (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 [BFifoPos](#) &pos)
Return the difference between the two pointers.
- [operator int](#) ()
- void [operator+=](#) (uint32_t numFifoSamples)
- int [operator==](#) (const [BFifoPos](#) &pos)
- int [operator!=](#) (const [BFifoPos](#) &pos)

Private Attributes

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

6.25.1 Detailed Description

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

6.25.2 Constructor & Destructor Documentation

6.25.2.1 [BFifoPos::BFifoPos](#) (uint32_t size)

6.25.3 Member Function Documentation

6.25.3.1 [void BFifoPos::setSize](#) (uint32_t size)

6.25.3.2 [void BFifoPos::set](#) (uint32_t pos)

Sets the position.

6.25.3.3 uint32_t BFifoPos::pos ()

The current position.

6.25.3.4 void BFifoPos::increment (uint32_t numFifoSamples)

Increment the pointer by the given value.

6.25.3.5 uint32_t BFifoPos::difference (const BFifoPos & pos)

Return the difference between the two pointers.

6.25.3.6 BFifoPos::operator int ()

6.25.3.7 void BFifoPos::operator+= (uint32_t numFifoSamples)

6.25.3.8 int BFifoPos::operator== (const BFifoPos & pos)

6.25.3.9 int BFifoPos::operator!= (const BFifoPos & pos)

6.25.4 Member Data Documentation

6.25.4.1 uint32_t BFifoPos::osize [private]

6.25.4.2 uint32_t BFifoPos::opos [private]

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

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

6.26 BFile Class Reference

File operations class.

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

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.
- void [close](#) ()
Close file.
- 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)
- 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.
- `BFile & operator= (const BFile &file)`

Private Attributes

- `FILE * ofile`
- `BString ofileName`
- `BString omode`

6.26.1 Detailed Description

File operations class.

6.26.2 Constructor & Destructor Documentation

6.26.2.1 BFile::BFile ()

6.26.2.2 BFile::BFile (const BFile &file)

Create opened specified file.

6.26.2.3 BFile::~~BFile ()

6.26.3 Member Function Documentation

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

Open file.

6.26.3.2 BError BFile::open (FILE *file)

Assign object to opened file handle.

6.26.3.3 BError BFile::open (int *fd*, BString *mode*)

Assign object to opened file descriptor.

6.26.3.4 void BFile::close ()

Close file.

6.26.3.5 int BFile::isEnd ()

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

6.26.3.6 FILE * BFile::getFd ()

File descriptor.

6.26.3.7 BUInt64 BFile::length ()

File size in bytes.

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

Set stream buffering options.

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

Read from file.

6.26.3.10 int BFile::readString (BString & *str*)

Read string. (ref fgets).

6.26.3.11 char * BFile::fgets (char * *buf*, size_t *size*)**6.26.3.12 int BFile::write (const void * *buf*, int *nbytes*)**

Write to file.

6.26.3.13 int BFile::writeString (const BString & *str*)

Write string to file.

6.26.3.14 int BFile::seek (BUInt64 *pos*)

Set seek position.

6.26.3.15 BUInt64 BFile::position ()

The files position.

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

Formatted print into the file.

6.26.3.17 BError BFile::truncate ()

Truncate the file.

6.26.3.18 BError BFile::flush ()

Flush the file.

6.26.3.19 BFile & BFile::operator= (const BFile & *file*)**6.26.4 Member Data Documentation****6.26.4.1 FILE* BFile::ofile** [private]**6.26.4.2 BString BFile::fileName** [private]**6.26.4.3 BString BFile::omode** [private]

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

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

6.27 BIter Class Reference

Iterator for [BList](#).

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

Public Member Functions

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

Private Attributes

- [BNode](#) * oi

6.27.1 Detailed Description

Iterator for [BList](#).

6.27.2 Constructor & Destructor Documentation

6.27.2.1 [BIter::BIter](#) ([BNode](#) * *i* = 0) [[inline](#)]

6.27.3 Member Function Documentation

6.27.3.1 [BIter::operator BNode *](#) () [[inline](#)]

6.27.3.2 [int BIter::operator==](#) (const [BIter](#) & *i*) [[inline](#)]

6.27.4 Member Data Documentation

6.27.4.1 [BNode*](#) [BIter::oi](#) [[private](#)]

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

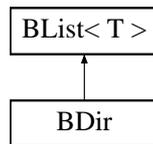
- [BList.h](#)

6.28 BList< T > Class Template Reference

Template based list class.

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

Inheritance diagram for BList< T >::



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](#) ([BIter](#) &i) const
Iterator to start of list.
- [BIter begin](#) () const
Iterator for start of list.
- [BIter end](#) () const
Iterator for end of list.
- [BIter end](#) ([BIter](#) &i) const
Iterator for end of list.
- void [next](#) ([BIter](#) &i) const
Iterator for next item in list.
- void [prev](#) ([BIter](#) &i)
Iterator for previous item in list.
- [BIter goTo](#) (int pos)
Iterator for pos item in list.
- int [position](#) ([BIter](#) i)
Position in list item with iterator i.
- unsigned int [number](#) ()

Number of items in list.

- unsigned int `size ()`
Number of items in list.
- int `isEnd (BIter &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 (BIter i)`
Get item specified by iterator in list.
- const T & `get (BIter 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.
- 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` ()

Classes

- class Node

6.28.1 Detailed Description

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

Template based list class.

6.28.2 Member Typedef Documentation

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

Prototype for sorting function.

6.28.3 Constructor & Destructor Documentation

6.28.3.1 `template<class T> BList< T >::BList () [inline]`

6.28.3.2 `template<class T> BList< T >::BList (const BList< T > &l) [inline]`

6.28.3.3 `template<class T> BList< T >::~~BList () [inline, virtual]`

6.28.4 Member Function Documentation

6.28.4.1 `template<class T> void BList< T >::start (BIter &i) const [inline]`

Iterator to start of list.

6.28.4.2 `template<class T> BIter BList< T >::begin () const [inline]`

Iterator for start of list.

6.28.4.3 `template<class T> BIter BList< T >::end () const [inline]`

Iterator for end of list.

6.28.4.4 `template<class T> BIter BList< T >::end (BIter &i) const [inline]`

Iterator for end of list.

6.28.4.5 `template<class T> void BList< T >::next (BIter &i) const [inline]`

Iterator for next item in list.

6.28.4.6 `template<class T> void BList< T >::prev (BIter &i) [inline]`

Iterator for previous item in list.

6.28.4.7 `template<class T> BIter BList< T >::goTo (int pos) [inline]`

Iterator for pos item in list.

6.28.4.8 `template<class T> int BList< T >::position (BIter i) [inline]`

Position in list item with iterator i.

6.28.4.9 `template<class T> unsigned int BList< T >::number ()` [inline]

Number of items in list.

6.28.4.10 `template<class T> unsigned int BList< T >::size ()` [inline]

Number of items in list.

6.28.4.11 `template<class T> int BList< T >::isEnd (BIter & i) const` [inline]

True if iterator refers to last item.

6.28.4.12 `template<class T> T & BList< T >::front ()` [inline]

Get first item in list.

6.28.4.13 `template<class T> T & BList< T >::rear ()` [inline]

Get last item in list.

6.28.4.14 `template<class T> T & BList< T >::get (BIter i)` [inline]

Get item specified by iterator in list.

6.28.4.15 `template<class T> const T & BList< T >::get (BIter i) const` [inline]

Get item specified by iterator in list.

6.28.4.16 `template<class T> void BList< T >::append (const T & item)` [inline]

Append item to list.

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

Insert item before item.

Reimplemented in [BEntryList](#).

6.28.4.18 `template<class T> void BList< T >::insertAfter (BIter & i, const T & item)` [inline]

Insert item after item.

6.28.4.19 `template<class T> void BList< T >::clear () [inline, virtual]`

Clear the list.

Reimplemented in [BDir](#), [BEntryList](#), and [BEntryFile](#).

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

Delete specified item.

Reimplemented in [BEntryList](#).

6.28.4.21 `template<class T> void BList< T >::deleteLast () [inline]`

Delete last item.

6.28.4.22 `template<class T> void BList< T >::deleteFirst () [inline]`

Delete first item.

6.28.4.23 `template<class T> void BList< T >::push (const T & i) [inline]`

Push item onto list.

6.28.4.24 `template<class T> T BList< T >::pop () [inline]`

Pop item from list deleting item.

6.28.4.25 `template<class T> void BList< T >::queueAdd (const T & i) [inline]`

Add item to end of list.

6.28.4.26 `template<class T> T BList< T >::queueGet () [inline]`

Get item from front of list deleting item.

6.28.4.27 `template<class T> void BList< T >::append (const BList< T > & l) [inline]`

Append list to list.

6.28.4.28 `template<class T> void BList< T >::swap (BIter i1, BIter i2) [inline]`

Swap two items in list.

6.28.4.29 `template<class T> void BList< T >::sort () [inline]`

Sort list based on get(i) values.

6.28.4.30 `template<class T> void BList< T >::sort (SortFunc func)` [inline]

Sort list based on Sort func.

6.28.4.31 `template<class T> BList< T > & BList< T >::operator= (const BList< T > & l)`
[inline]

6.28.4.32]

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

6.28.4.33]

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

6.28.4.34]

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

6.28.4.35]

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

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

6.28.4.36 `template<class T> BList< T > BList< T >::operator+ (const BList< T > & l) const`
[inline]

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

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

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

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

6.28.5 Member Data Documentation

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

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

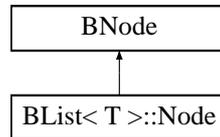
The documentation for this class was generated from the following files:

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

6.29 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](#)

```
template<class T> class BList< T >::Node
```

6.29.1 Constructor & Destructor Documentation

6.29.1.1 `template<class T> BList< T >::Node::Node (const T & i) [inline]`

6.29.2 Member Data Documentation

6.29.2.1 `template<class T> T BList< T >::Node::item`

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

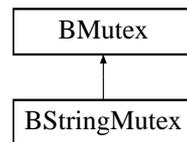
- [BList.h](#)

6.30 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`

6.30.1 Detailed Description

Mutex class.

6.30.2 Member Enumeration Documentation

6.30.2.1 enum BMutex::Type

Enumerator:

Normal

Recursive

6.30.3 Constructor & Destructor Documentation

6.30.3.1 BMutex::BMutex (Type *type* = Normal)

6.30.3.2 BMutex::BMutex (const BMutex & *mutex*)

6.30.3.3 BMutex::~~BMutex ()

6.30.4 Member Function Documentation

6.30.4.1 int BMutex::lock ()

Set lock, wait as necessary.

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

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

6.30.4.3 int BMutex::unlock ()

Unlock the lock.

6.30.4.4 int BMutex::tryLock ()

Test the lock.

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

6.30.5 Member Data Documentation

6.30.5.1 pthread_mutex_t BMutex::omutex [private]

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

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

6.31 BMutexLock Class Reference

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

Public Member Functions

- [BMutexLock \(BMutex &lock\)](#)
- [~BMutexLock \(\)](#)

Private Attributes

- [BMutex & olock](#)

6.31.1 Constructor & Destructor Documentation

6.31.1.1 [BMutexLock::BMutexLock \(BMutex & lock\)](#) [inline]

6.31.1.2 [BMutexLock::~~BMutexLock \(\)](#) [inline]

6.31.2 Member Data Documentation

6.31.2.1 [BMutex& BMutexLock::olock](#) [private]

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

- [BMutex.h](#)

6.32 BMySQL Class Reference

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

Public Member Functions

- [BMySQL \(\)](#)
- [~BMySQL \(\)](#)
- [BError open \(BString hostName, BString dataBase, BString userName, BString password\)](#)
- [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 query \(BString cmd, BList< BDictString > &result\)](#)
- [MYSQL & db \(\)](#)
- [void setDebug \(int debug\)](#)

Private Attributes

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

6.32.1 Constructor & Destructor Documentation

6.32.1.1 `BMySQL::BMySQL ()`

6.32.1.2 `BMySQL::~~BMySQL ()`

6.32.2 Member Function Documentation

6.32.2.1 `BError BMySQL::open (BString hostName, BString dataBase, BString userName, BString password)`

6.32.2.2 `BError BMySQL::get (BString table, BString where, BDictString & fields)`

6.32.2.3 `BError BMySQL::insert (BString table, BDictString fields, BUInt32 * id = 0)`

6.32.2.4 `BError BMySQL::update (BString table, BUInt32 id, BDictString fields)`

6.32.2.5 `BError BMySQL::query (BString cmd, BList< BDictString > & result)`

6.32.2.6 `MYSQL & BMySQL::db ()`

6.32.2.7 `void BMySQL::setDebug (int debug)`

6.32.3 Member Data Documentation

6.32.3.1 `MYSQL BMySQL::odb` [private]

6.32.3.2 `int BMySQL::oopened` [private]

6.32.3.3 `int BMySQL::odebug` [private]

6.32.3.4 `BMutex BMySQL::olock` [private]

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

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

6.33 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](#)

```
template<class T> class BNameValue< T >
```

6.33.1 Constructor & Destructor Documentation

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

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

6.33.2 Member Function Documentation

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

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

6.33.3 Member Data Documentation

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

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

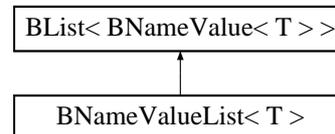
The documentation for this class was generated from the following file:

- [BNameValue.h](#)

6.34 BNameValueList< T > Class Template Reference

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

Inheritance diagram for BNameValueList< T >::



Public Member Functions

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

```
template<class T> class BNameValueList< T >
```

6.34.1 Member Function Documentation

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

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

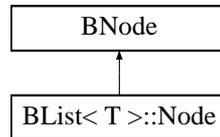
The documentation for this class was generated from the following file:

- [BNameValue.h](#)

6.35 BNode Class Reference

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

Inheritance diagram for BNode::



Public Member Functions

- [BNode \(\)](#)

Public Attributes

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

6.35.1 Constructor & Destructor Documentation

6.35.1.1 [BNode::BNode \(\)](#) [inline]

6.35.2 Member Data Documentation

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

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

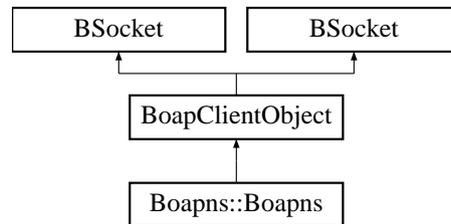
The documentation for this class was generated from the following file:

- [BList.h](#)

6.36 BoapClientObject Class Reference

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

Inheritance diagram for BoapClientObject::



Public Member Functions

- [BoapClientObject](#) (BString name="")
- [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.

- **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

6.36.1 Constructor & Destructor Documentation

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

6.36.1.2 **BoapClientObject::BoapClientObject** (**BString** name)

6.36.2 Member Function Documentation

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

Connects to the named service.

6.36.2.2 **BError BoapClientObject::disconnectService** ()

Disconnects from the named service.

6.36.2.3 **BString BoapClientObject::getServiceName** ()

Get the name of the service.

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

Pings the connection and finds the remotes version number.

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

Sets the connection priority.

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

Sets the maximum packet length.

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

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

6.36.2.8 BError BoapClientObject::pingLocked (BUInt32 & *apiVersion*) [protected]**6.36.2.9 BError BoapClientObject::checkApiVersion ()** [protected]**6.36.2.10 BError BoapClientObject::performCall (BoapPacket & *tx*, BoapPacket & *rx*)**
[protected]

Performs a RPC call to the named service.

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

Performs a send to the named service.

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

Performs a receive.

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

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

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

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

6.36.3 Member Data Documentation

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

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

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

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

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

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

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

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

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

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

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

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

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

6.37 Boapns::BoapEntry Class Reference

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

Public Member Functions

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

Public Attributes

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

6.37.1 Constructor & Destructor Documentation

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

6.37.2 Member Data Documentation

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

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

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

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

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

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

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

6.38 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

6.38.1 Constructor & Destructor Documentation

6.38.1.1 [BoapFuncEntry::BoapFuncEntry](#) (int *cmd*, [BoapFunc](#) *func*)

6.38.1.2 [BoapFuncEntry::BoapFuncEntry](#) (int *cmd*, [BoapFunc](#) *func*)

6.38.2 Member Data Documentation

6.38.2.1 [BUInt32](#) [BoapFuncEntry::ocmd](#)

6.38.2.2 [BoapFunc](#) [BoapFuncEntry::ofunc](#)

6.38.2.3 [UInt32](#) [BoapFuncEntry::ocmd](#)

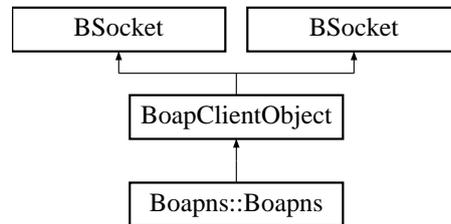
The documentation for this class was generated from the following files:

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

6.39 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)`

6.39.1 Constructor & Destructor Documentation

6.39.1.1 `Boapns::Boapns::Boapns (BString name = "")`

6.39.2 Member Function Documentation

6.39.2.1 `BError Boapns::Boapns::getVersion (BString & version)`

6.39.2.2 `BError Boapns::Boapns::getEntryList (BList< BoapEntry > & entryList)`

6.39.2.3 `BError Boapns::Boapns::getEntry (BString name, BoapEntry & entry)`

6.39.2.4 `BError Boapns::Boapns::addEntry (BoapEntry entry)`

6.39.2.5 `BError Boapns::Boapns::delEntry (BString name)`

6.39.2.6 `BError Boapns::Boapns::getNewName (BString & name)`

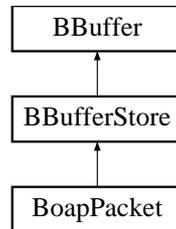
The documentation for this class was generated from the following file:

- [BoapnsC.h](#)

6.40 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 \(\)](#)
 - The 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`

6.40.1 Constructor & Destructor Documentation

6.40.1.1 `BoapPacket::BoapPacket ()`

6.40.1.2 `BoapPacket::~~BoapPacket ()`

6.40.1.3 `BoapPacket::BoapPacket ()`

6.40.1.4 `BoapPacket::~~BoapPacket ()`

6.40.2 Member Function Documentation

6.40.2.1 `BUInt32 BoapPacket::getCmd ()`

6.40.2.2 `int BoapPacket::peekHead (BoapPacketHead & head)`

6.40.2.3 `int BoapPacket::pushHead (BoapPacketHead & head)`

6.40.2.4 `int BoapPacket::popHead (BoapPacketHead & head)`

6.40.2.5 `void BoapPacket::updateHead ()`

6.40.2.6 `int BoapPacket::resize (int size)`

6.40.2.7 `BError BoapPacket::setData (void * data, int nbytes)`

6.40.2.8 `int BoapPacket::nbytes ()`

6.40.2.9 `char * BoapPacket::data ()`

The data.

Reimplemented from [BBuffer](#).

6.40.2.10 `int BoapPacket::pushHead (BoapPacketHead & head)`

6.40.2.11 `int BoapPacket::push (Int8 v)`

Reimplemented from [BBufferStore](#).

6.40.2.12 `int BoapPacket::push (UInt8 v)`

Reimplemented from [BBufferStore](#).

6.40.2.13 `int BoapPacket::push (Int16 v)`

Reimplemented from [BBufferStore](#).

6.40.2.14 `int BoapPacket::push (UInt16 v)`

Reimplemented from [BBufferStore](#).

6.40.2.15 `int BoapPacket::push (Int32 v)`

Reimplemented from [BBufferStore](#).

6.40.2.16 `int BoapPacket::push (UInt32 v)`

Reimplemented from [BBufferStore](#).

6.40.2.17 `int BoapPacket::push (BString & v)`

6.40.2.18 `int BoapPacket::push (Double v)`

Reimplemented from [BBufferStore](#).

6.40.2.19 `int BoapPacket::push (BError & v)`

6.40.2.20 `int BoapPacket::push (UInt32 nBytes, const void * data)`

6.40.2.21 `int BoapPacket::popHead (BoapPacketHead & head)`

6.40.2.22 `int BoapPacket::pop (Int8 & v)`

Reimplemented from [BBufferStore](#).

6.40.2.23 `int BoapPacket::pop (UInt8 & v)`

Reimplemented from [BBufferStore](#).

6.40.2.24 int BoapPacket::pop (Int16 & v)

Reimplemented from [BBufferStore](#).

6.40.2.25 int BoapPacket::pop (UInt16 & v)

Reimplemented from [BBufferStore](#).

6.40.2.26 int BoapPacket::pop (Int32 & v)

Reimplemented from [BBufferStore](#).

6.40.2.27 int BoapPacket::pop (UInt32 & v)

Reimplemented from [BBufferStore](#).

6.40.2.28 int BoapPacket::pop (BString & v)

Reimplemented from [BBufferStore](#).

6.40.2.29 int BoapPacket::pop (Double & v)

Reimplemented from [BBufferStore](#).

6.40.2.30 int BoapPacket::pop (BError & v)

Reimplemented from [BBufferStore](#).

6.40.2.31 int BoapPacket::pop (UInt32 nBytes, void * data)**6.40.2.32 void BoapPacket::updateLen () [private]****6.40.3 Member Data Documentation****6.40.3.1 int BoapPacket::osize [private]**

Reimplemented from [BBuffer](#).

6.40.3.2 int BoapPacket::onbytes [private]**6.40.3.3 char* BoapPacket::odata [private]**

Reimplemented from [BBuffer](#).

6.40.3.4 int BoapPacket::opos [private]

Reimplemented from [BBufferStore](#).

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

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

6.41 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 \[12\]](#)

6.41.1 Member Data Documentation

6.41.1.1 [BUInt32 BoapPacketHead::type](#)

6.41.1.2 [BUInt32 BoapPacketHead::length](#)

6.41.1.3 [BUInt32 BoapPacketHead::service](#)

6.41.1.4 [BUInt32 BoapPacketHead::cmd](#)

6.41.1.5 [UInt32 BoapPacketHead::length](#)

6.41.1.6 [BoapType BoapPacketHead::type](#)

6.41.1.7 [BoapService BoapPacketHead::service](#)

6.41.1.8 [UInt32 BoapPacketHead::cmd](#)

6.41.1.9 [UInt32 BoapPacketHead::reserved\[12\]](#)

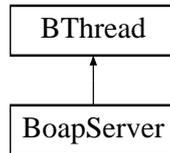
The documentation for this struct was generated from the following files:

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

6.42 BoapServer Class Reference

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

Inheritance diagram for BoapServer::



Public Types

- enum { `NOTHREADS` = 0, `THREADED` = 1 }

Public Member Functions

- `BoapServer ()`
- `~BoapServer ()`
- `BError init (BString boapNsHost="", int port=0, int threaded=0, int isBoapns=0)`
- `BError run (int inThread=0)`
- `BError processEvent (BoapPacket &rx)`
- `BError addObject (BoapServiceObject *object)`
- `BError process (BoapServerConnection *conn, BoapPacket &rx, BoapPacket &tx)`
- `BError sendEvent (BoapPacket &tx)`
- `BSocket & getSocket ()`
- `BSocket & getEventSocket ()`
- `BError processEvent (int fd)`
- `BString getHostName ()`
- `void clientGone (BoapServerConnection *client)`
- `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](#)
- [BEventInt](#) [oclientGoneEvent](#)
- [BList](#)< [BoapServiceEntry](#) > [oservices](#)
- [BPoll](#) [opoll](#)
- [BSocket](#) [onet](#)
- [BSocket](#) [onetEvent](#)
- [BSocketAddressINET](#) [onetEventAddress](#)
- [BString](#) [ohostName](#)
- int [oboapNs](#)
- [BoapPacket](#) [orx](#)
- [BoapPacket](#) [otx](#)

6.42.1 Member Enumeration Documentation

6.42.1.1 anonymous enum

Enumerator:

NOTHREADS

THREADED

6.42.2 Constructor & Destructor Documentation

6.42.2.1 **BoapServer::BoapServer ()**

6.42.2.2 **BoapServer::~~BoapServer ()**

6.42.2.3 **BoapServer::BoapServer ()**

6.42.3 Member Function Documentation

6.42.3.1 **BError BoapServer::init (BString *boapNsHost* = "", int *port* = 0, int *threaded* = 0, int *isBoapns* = 0)**

6.42.3.2 **BError BoapServer::run (int *inThread* = 0)**

6.42.3.3 **BError BoapServer::processEvent (BoapPacket & *rx*)**

6.42.3.4 **BError BoapServer::addObject (BoapServiceObject * *object*)**

6.42.3.5 **BError BoapServer::process (BoapServerConnection * *conn*, BoapPacket & *rx*, BoapPacket & *tx*)**

6.42.3.6 **BError BoapServer::sendEvent (BoapPacket & *tx*)**

6.42.3.7 **BSocket & BoapServer::getSocket ()**

6.42.3.8 **BSocket & BoapServer::getEventSocket ()**

6.42.3.9 **BError BoapServer::processEvent (int *fd*)**

6.42.3.10 **BString BoapServer::getHostName ()**

6.42.3.11 **void BoapServer::clientGone (BoapServerConnection * *client*)**

6.42.3.12 **int BoapServer::getConnectionsNumber ()**

6.42.3.13 **BoapServerConnection * BoapServer::newConnection (int *fd*, BSocketAddressINET *address*) [virtual]**

6.42.3.14 **void * BoapServer::function () [private, virtual]**

Reimplemented from [BThread](#).

- 6.42.3.15 BError BoapServer::init (int *boapNs* = 0)
- 6.42.3.16 BError BoapServer::run ()
- 6.42.3.17 BError BoapServer::processEvent (BoapPacket & *rx*)
- 6.42.3.18 BError BoapServer::addObject (BoapServiceObject * *object*)
- 6.42.3.19 BError BoapServer::process (int *fd*)
- 6.42.3.20 BError BoapServer::sendEvent (BoapPacket & *tx*)
- 6.42.3.21 BSocket& BoapServer::getSocket ()
- 6.42.3.22 BSocket& BoapServer::getEventSocket ()
- 6.42.3.23 BError BoapServer::processEvent (int *fd*)
- 6.42.3.24 BString BoapServer::getHostName ()

6.42.4 Member Data Documentation

- 6.42.4.1 int BoapServer::othreaded [private]
- 6.42.4.2 int BoapServer::oisBoapns [private]
- 6.42.4.3 Boapns::Boapns* BoapServer::oboapns [private]
- 6.42.4.4 BList<BoapServerConnection*> BoapServer::oclients [private]
- 6.42.4.5 BEventInt BoapServer::oclientGoneEvent [private]
- 6.42.4.6 BList< BoapServiceEntry > BoapServer::oservices [private]
- 6.42.4.7 BPoll BoapServer::opoll [private]
- 6.42.4.8 BSocket BoapServer::onet [private]
- 6.42.4.9 BSocket BoapServer::onetEvent [private]
- 6.42.4.10 BSocketAddressINET BoapServer::onetEventAddress [private]
- 6.42.4.11 BString BoapServer::ohostName [private]
- 6.42.4.12 BUInt64 BoapServer::onumOperations
- 6.42.4.13 int BoapServer::oboapNs [private]
- 6.42.4.14 BoapPacket BoapServer::orx [private]
- 6.42.4.15 BoapPacket BoapServer::otx [private]

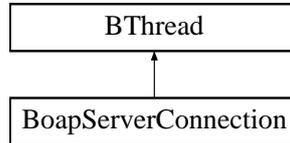
The documentation for this class was generated from the following files:

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

6.43 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

6.43.1 Constructor & Destructor Documentation

6.43.1.1 [BoapServerConnection::BoapServerConnection](#) ([BoapServer](#) & boapServer, int fd)

6.43.1.2 [BoapServerConnection::~~BoapServerConnection](#) () [virtual]

6.43.2 Member Function Documentation

6.43.2.1 [BError](#) [BoapServerConnection::init](#) () [virtual]

Initialise connection.

6.43.2.2 `BError BoapServerConnection::process ()` [virtual]

6.43.2.3 `BSocket & BoapServerConnection::getSocket ()` [virtual]

6.43.2.4 `void BoapServerConnection::setMaxLength (BUInt32 maxLength)` [virtual]

6.43.2.5 `BError BoapServerConnection::getHead (BoapPacketHead & head)` [virtual]

6.43.2.6 `BError BoapServerConnection::validate ()` [virtual]

Validate the connection.

6.43.2.7 `void * BoapServerConnection::function ()` [private, virtual]

Reimplemented from [BThread](#).

6.43.3 Member Data Documentation

6.43.3.1 `BoapServer& BoapServerConnection::oboapServer` [private]

6.43.3.2 `BSocket BoapServerConnection::osocket` [private]

6.43.3.3 `BoapPacket BoapServerConnection::orx` [private]

6.43.3.4 `BoapPacket BoapServerConnection::otx` [private]

6.43.3.5 `BUInt32 BoapServerConnection::omaxLength` [private]

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

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

6.44 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

6.44.1 Constructor & Destructor Documentation

6.44.1.1 [BoapServiceEntry::BoapServiceEntry](#) ([BoapService](#) *service* = 0, [BoapServiceObject](#) * *object* = 0) [[inline](#)]

6.44.1.2 [BoapServiceEntry::BoapServiceEntry](#) ([BoapService](#) *service* = 0, [BoapServiceObject](#) * *object* = 0) [[inline](#)]

6.44.2 Member Data Documentation

6.44.2.1 [BoapService](#) [BoapServiceEntry::oservice](#)

6.44.2.2 [BoapServiceObject](#) * [BoapServiceEntry::oobject](#)

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

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

6.45 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

6.45.1 Constructor & Destructor Documentation

6.45.1.1 **BoapServiceObject::BoapServiceObject** (BoapServer & *server*, BString *name* = "")

6.45.1.2 **BoapServiceObject::~~BoapServiceObject** () [virtual]

6.45.1.3 **BoapServiceObject::BoapServiceObject** (BoapServer & *server*, BString *name*)

6.45.1.4 **virtual BoapServiceObject::~~BoapServiceObject** () [virtual]

6.45.2 Member Function Documentation

6.45.2.1 **BError BoapServiceObject::setName** (BString *name*)

6.45.2.2 **BError BoapServiceObject::sendEvent** (BString *signalName*, BInt32 *arg*)

6.45.2.3 **BError BoapServiceObject::processEvent** (BString *objectName*, BString *name*, BInt32 *arg*) [virtual]

6.45.2.4 **BString BoapServiceObject::name** ()

6.45.2.5 **BError BoapServiceObject::doPing** (BoapServerConnection * *conn*, BoapPacket & *rx*, BoapPacket & *tx*)

6.45.2.6 **BError BoapServiceObject::doConnectionPriority** (BoapServerConnection * *conn*, BoapPacket & *rx*, BoapPacket & *tx*)

6.45.2.7 **BError BoapServiceObject::process** (BoapServerConnection * *conn*, BoapPacket & *rx*, BoapPacket & *tx*)

6.45.2.8 **BError BoapServiceObject::processEvent** (BoapPacket & *rx*) [virtual]

6.45.2.9 **BError BoapServiceObject::sendEvent** (BoapPacket & *tx*) [protected]

6.45.2.10 **BError BoapServiceObject::sendEvent** (BString *signalName*, BInt32 *arg*)

6.45.2.11 **virtual BError BoapServiceObject::processEvent** (BString *objectName*, BString *name*, BInt32 *arg*) [virtual]

6.45.2.12 **BString BoapServiceObject::name** ()

6.45.2.13 **BError BoapServiceObject::process** (BoapPacket & *rx*, BoapPacket & *tx*)

6.45.2.14 **virtual BError BoapServiceObject::processEvent** (BoapPacket & *rx*) [virtual]

6.45.2.15 **BError BoapServiceObject::sendEvent** (BoapPacket & *tx*) [protected]

6.45.3 Member Data Documentation

6.45.3.1 **BoapServer & BoapServiceObject::oserver** [protected]

6.45.3.2 **BString BoapServiceObject::oname** [protected]

6.45.3.3 **BUInt32 BoapServiceObject::oapiVersion** [protected]

Generated on Wed Jun 3 13:59:47 2009 for LibBeamApi by Doxygen

6.45.3.4 **BList< BoapFuncEntry > BoapServiceObject::ofuncList** [protected]

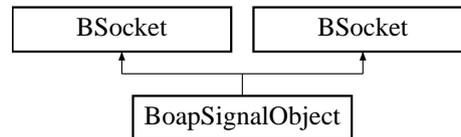
The documentation for this class was generated from the following files:

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

6.46 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](#)

6.46.1 Constructor & Destructor Documentation

6.46.1.1 [BoapSignalObject::BoapSignalObject \(\)](#)

6.46.1.2 [BoapSignalObject::BoapSignalObject \(\)](#)

6.46.2 Member Function Documentation

6.46.2.1 [BError BoapSignalObject::performSend \(BoapPacket & tx\)](#) [protected]

6.46.2.2 [BError BoapSignalObject::performSend \(BoapPacket & tx\)](#) [protected]

6.46.3 Member Data Documentation

6.46.3.1 [BoapPacket BoapSignalObject::otx](#) [protected]

6.46.3.2 [BoapPacket BoapSignalObject::orx](#) [protected]

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

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

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

6.47 BObj Class Reference

```
#include <BObj.h>
```

Public Member Functions

- [BObj \(\)](#)
- virtual [~BObj \(\)](#)
- virtual [BString getType \(\)](#)
- virtual [BError getMembers \(BDictString &members\)](#)
- virtual [BError setMembers \(BDictString &members\)](#)
- virtual [BString getDebugString \(\)](#)

Returns contents as a debug string.

6.47.1 Constructor & Destructor Documentation

6.47.1.1 [BObj::BObj \(\)](#)

6.47.1.2 [BObj::~~BObj \(\)](#) [virtual]

6.47.2 Member Function Documentation

6.47.2.1 [BString BObj::getType \(\)](#) [virtual]

6.47.2.2 [BError BObj::getMembers \(BDictString & members\)](#) [virtual]

6.47.2.3 [BError BObj::setMembers \(BDictString & members\)](#) [virtual]

6.47.2.4 [BString BObj::getDebugString \(\)](#) [virtual]

Returns contents as a debug string.

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

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

6.48 BObject Class Reference

```
#include <BObject.h>
```

Public Member Functions

- [BObject](#) ()
- virtual [~BObject](#) ()
- virtual [BError](#) [getBinary](#) (BDataBuf &buf)
- virtual [BError](#) [setBinary](#) (BDataBuf &buf)
- virtual [BString](#) [getString](#) ()
- virtual [BError](#) [setString](#) ([BString](#) str)
- virtual [BMemberList](#) [getMemberList](#) ()
- virtual [BError](#) [addMember](#) ([BString](#) name, [BObject](#) *object)
- virtual void [printIt](#) ()
- virtual BType & [getType](#) ()

Static Public Member Functions

- static [BObject](#) * [createObj](#) ()

Static Public Attributes

- static BType [otype](#) = btypesList.appendType(BType("BObject", BTypeDomainBase, BTypeObject, createObj))

6.48.1 Constructor & Destructor Documentation

6.48.1.1 BObject::BObject ()

6.48.1.2 BObject::~~BObject () [virtual]

6.48.2 Member Function Documentation

6.48.2.1 BError BObject::getBinary (BDataBuf & *buf*) [virtual]

6.48.2.2 BError BObject::setBinary (BDataBuf & *buf*) [virtual]

6.48.2.3 BString BObject::getString () [virtual]

6.48.2.4 BError BObject::setString (BString *str*) [virtual]

6.48.2.5 BMemberList BObject::getMemberList () [virtual]

6.48.2.6 BError BObject::addMember (BString *name*, BObject * *object*) [virtual]

6.48.2.7 void BObject::printIt () [virtual]

6.48.2.8 BType & BObject::getType () [virtual]

6.48.2.9 BObject * BObject::createObj () [static]

6.48.3 Member Data Documentation

6.48.3.1 BType BObject::otype = btypesList.appendType(BType("BObject", BTypeDomainBase, BTypeObject, createObj)) [static]

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

- [BObject.h](#)
- [BObject.cc](#)

6.49 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.
- 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.

6.49.1 Detailed Description

This class provides an interface for polling a number of file descriptors. It uses round robin polling.

6.49.2 Member Typedef Documentation

6.49.2.1 `typedef struct pollfd BPoll::PollFd` [read]

6.49.3 Constructor & Destructor Documentation

6.49.3.1 `BPoll::BPoll ()`

6.49.3.2 `BPoll::~~BPoll ()`

6.49.4 Member Function Documentation

6.49.4.1 `void BPoll::append (int fd, int events = POLLIN|POLLERR|POLLHUP|POLLNVAL)`

Append a file descriptor to polling list.

6.49.4.2 `void BPoll::delFd (int fd)`

Remove a file descriptor from polling list.

6.49.4.3 `BError BPoll::doPoll (int &fd, int timeoutUs = -1)`

Perform polling operation.

6.49.4.4 `int BPoll::getPollFdsNum ()`

6.49.4.5 `BPoll::PollFd * BPoll::getPollFds ()`

6.49.4.6 `void BPoll::clear ()`

6.49.4.7 `int BPoll::nextFd (int i)` [private]

6.49.5 Member Data Documentation

6.49.5.1 `int BPoll::ofdsNum` [private]

The number of FD's in list.

6.49.5.2 `PollFd* BPoll::ofds` [private]

The list of poll fd's.

6.49.5.3 `int BPoll::ofdsNext` [private]

The next list entry for round robin polling.

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

- [BPoll.h](#)

- [BPoll.cpp](#)

6.50 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.

6.50.1 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.

6.50.2 Constructor & Destructor Documentation

6.50.2.1 `BRefData::BRefData ()`

6.50.2.2 `BRefData::BRefData (int len)`

6.50.2.3 `BRefData::BRefData (const BRefData & refData)`

6.50.2.4 `BRefData::~~BRefData ()`

6.50.3 Member Function Documentation

6.50.3.1 `BRefData * BRefData::copy ()`

Create a copy of this reference for writing, if necessary.

6.50.3.2 `BRefData * BRefData::addRef ()`

Increment the reference counter.

6.50.3.3 `int BRefData::deleteRef ()`

Decrement the reference counter.

6.50.3.4 `char* BRefData::data () [inline]`

Return the raw data pointer.

6.50.3.5 `int BRefData::len () [inline]`

Return the length in bytes.

6.50.3.6 `BRefData & BRefData::operator= (const BRefData & refData)`

6.50.3.7 `void BRefData::setLen (int len)`

Set the length in bytes. Note should only be used if `orefCount = 1`.

6.50.4 Member Data Documentation

6.50.4.1 `BAtomicCount BRefData::orefCount [private]`

The reference count, how many users.

6.50.4.2 `int BRefData::olen [private]`

The actual length of data in `oData`.

6.50.4.3 void* BRefData::odata [private]

Pointer to the data.

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

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

6.51 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](#)

6.51.1 Detailed Description

Realtime clock.

6.51.2 Constructor & Destructor Documentation

6.51.2.1 BRtc::BRtc ()

6.51.2.2 BRtc::~~BRtc ()

6.51.3 Member Function Documentation

6.51.3.1 BError BRtc::init (int rate)

Setup interrupt rate.

6.51.3.2 void BRtc::wait (int delayUs)

Wait specified uS.

6.51.4 Member Data Documentation

6.51.4.1 int BRtc::ofd [private]

6.51.4.2 int BRtc::orate [private]

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

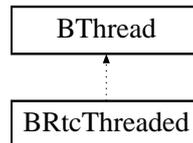
- [BRtc.h](#)
- [BRtc.cpp](#)

6.52 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 orts](#)
- [int orate](#)
- [BCond ocond](#)

6.52.1 Detailed Description

Threaded real time clock.

6.52.2 Constructor & Destructor Documentation

6.52.2.1 BRtcThreaded::BRtcThreaded ()

6.52.2.2 BRtcThreaded::~~BRtcThreaded ()

6.52.3 Member Function Documentation

6.52.3.1 BError BRtcThreaded::init (int rate)

Setup interrupt rate.

6.52.3.2 void BRtcThreaded::wait (int *delayUs*)

Wait specified uS.

6.52.3.3 void * BRtcThreaded::function () [private, virtual]

Reimplemented from [BThread](#).

6.52.4 Member Data Documentation

6.52.4.1 BRtc BRtcThreaded::ortc [private]

6.52.4.2 int BRtcThreaded::orate [private]

6.52.4.3 BCond BRtcThreaded::ocond [private]

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

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

6.53 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`

6.53.1 Detailed Description

thread read-write locks

6.53.2 Constructor & Destructor Documentation

6.53.2.1 BRWLock::BRWLock ()

6.53.2.2 BRWLock::BRWLock (const BRWLock & rwlock)

6.53.2.3 BRWLock::~~BRWLock ()

6.53.3 Member Function Documentation

6.53.3.1 int BRWLock::rdLock ()

Set lock, wait if necessary.

6.53.3.2 int BRWLock::tryRdLock ()

Test the lock.

6.53.3.3 int BRWLock::wrLock ()

Set lock, wait if necessary.

6.53.3.4 int BRWLock::tryWrLock ()

Test the lock.

6.53.3.5 int BRWLock::unlock ()

Unlock the lock.

6.53.3.6 BRWLock & BRWLock::operator= (const BRWLock & *rwlock*)

6.53.4 Member Data Documentation

6.53.4.1 pthread_rwlock_t BRWLock::oLock [private]

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

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

6.54 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](#)

6.54.1 Detailed Description

Sempahore class.

6.54.2 Constructor & Destructor Documentation

6.54.2.1 [BSema::BSema](#) (int *value* = 0)

6.54.2.2 [BSema::BSema](#) (const [BSema](#) & *sema*)

6.54.2.3 [BSema::~~BSema](#) ()

6.54.3 Member Function Documentation

6.54.3.1 int [BSema::post](#) ()

Post condition.

6.54.3.2 int BSema::wait ()

Wait for contition.

6.54.3.3 int BSema::timedWait (int *timeUs*)

Wait for condition with timeout.

6.54.3.4 int BSema::tryWait ()

Test for the condition.

6.54.3.5 int BSema::getValue () const**6.54.3.6 BSema & BSema::operator= (const BSema & *sema*)****6.54.4 Member Data Documentation****6.54.4.1 sem_t BSema::osema [private]**

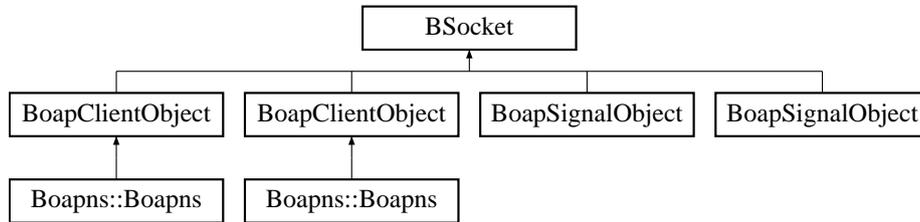
The documentation for this class was generated from the following files:

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

6.55 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](#) ()
- [BError](#) init (NType type)
- [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`

6.55.1 Member Enumeration Documentation

6.55.1.1 enum BSocket::NType

Enumerator:

STREAM

DGRAM

6.55.1.2 enum BSocket::Priority

Enumerator:

PriorityLow

PriorityNormal

PriorityHigh

6.55.2 Constructor & Destructor Documentation

6.55.2.1 BSocket::BSocket ()

6.55.2.2 BSocket::BSocket (int *fd*)

6.55.2.3 BSocket::BSocket (NType *type*)

6.55.2.4 BSocket::~~BSocket ()

6.55.3 Member Function Documentation

6.55.3.1 BError BSocket::init (NType *type*)

6.55.3.2 int BSocket::getFd ()

6.55.3.3 BError BSocket::bind (const BSocketAddress & *add*)

6.55.3.4 BError BSocket::connect (const BSocketAddress & *add*)

6.55.3.5 BError BSocket::shutdown (int *how*)

6.55.3.6 BError BSocket::close ()

6.55.3.7 BError BSocket::listen (int *backlog* = 5)

6.55.3.8 BError BSocket::accept (int & *fd*)

6.55.3.9 BError BSocket::accept (int & *fd*, BSocketAddress & *address*)

6.55.3.10 BError BSocket::send (const void * *buf*, BSize *nbytes*, BSize & *nbytesSent*, int *flags* = 0)

6.55.3.11 BError BSocket::sendTo (const BSocketAddress & *address*, const void * *buf*, BSize *nbytes*, BSize & *nbytesSent*, int *flags* = 0)

6.55.3.12 BError BSocket::recv (void * *buf*, BSize *maxbytes*, BSize & *nbytesRecv*, int *flags* = 0)

6.55.3.13 BError BSocket::recvFrom (BSocketAddress & *address*, void * *buf*, BSize *maxbytes*, BSize & *nbytesRecv*, int *flags* = 0)

6.55.3.14 BError BSocket::recvWithTimeout (void * *buf*, BSize *maxbytes*, BSize & *nbytesRecv*, int *timeout*, int *flags* = 0)

6.55.3.15 BError BSocket::recvFromWithTimeout (BSocketAddress & *address*, void * *buf*, BSize *maxbytes*, BSize & *nbytesRecv*, int *timeout*, int *flags* = 0)

6.55.3.16 BError BSocket::setSockOpt (int *level*, int *optname*, void * *optval*, unsigned int *optlen*)

6.55.3.17 BError BSocket::getSockOpt (int *level*, int *optname*, void * *optval*, unsigned int * *optlen*)

6.55.3.18 BError BSocket::setReuseAddress (int *on*)

6.55.3.19 BError BSocket::setBroadCast (int *on*)

Generated on Wed Jun 3 13:59:47 2009 for LibBeamApi by Doxygen

6.55.3.20 BError BSocket::setPriority (Priority *priority*)

6.55.3.21 BError BSocket::getMTU (uint32_t & *mtu*)

6.55.3.22 BError BSocket::getAddress (BSocketAddress & *address*)

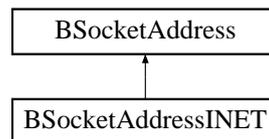
- [BSocket.h](#)
- [BSocket.cpp](#)

6.56 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](#)

6.56.1 Detailed Description

Socket Address.

6.56.2 Member Typedef Documentation

6.56.2.1 `typedef struct sockaddr BSocketAddress::SockAddr` [read]

6.56.3 Constructor & Destructor Documentation

6.56.3.1 `BSocketAddress::BSocketAddress ()`

6.56.3.2 `BSocketAddress::BSocketAddress (const BSocketAddress & add)`

6.56.3.3 `BSocketAddress::BSocketAddress (SockAddr * address, int len)`

6.56.3.4 `BSocketAddress::~~BSocketAddress ()`

6.56.4 Member Function Documentation

6.56.4.1 `BError BSocketAddress::set (SockAddr * address, int len)`

6.56.4.2 `const BSocketAddress::SockAddr * BSocketAddress::raw () const`

6.56.4.3 `int BSocketAddress::len () const`

6.56.4.4 `BSocketAddress & BSocketAddress::operator= (const BSocketAddress & add)`

6.56.4.5 `BSocketAddress::operator const SockAddr * () const` [inline]

6.56.4.6 `int BSocketAddress::operator== (const BSocketAddress & add) const`

6.56.4.7 `int BSocketAddress::operator!= (const BSocketAddress & add) const`

6.56.5 Member Data Documentation

6.56.5.1 `int BSocketAddress::olen` [private]

6.56.5.2 `SockAddr* BSocketAddress::oaddress` [private]

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

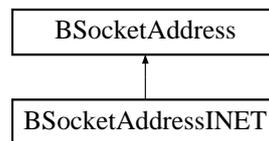
- [BSocket.h](#)
- [BSocket.cpp](#)

6.57 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.

6.57.1 Detailed Description

IP aware socket address.

6.57.2 Member Typedef Documentation

6.57.2.1 `typedef struct sockaddr_in BSocketAddressINET::SockAddrIP` [read]

6.57.3 Member Function Documentation

6.57.3.1 `BError BSocketAddressINET::set (BString hostName, uint32_t port)`

6.57.3.2 `BError BSocketAddressINET::set (uint32_t address, uint32_t port)`

6.57.3.3 `BError BSocketAddressINET::set (BString hostName, BString service, BString type)`

6.57.3.4 `void BSocketAddressINET::setPort (uint32_t port)`

6.57.3.5 `uint32_t BSocketAddressINET::address ()`

Returns socket ip address.

6.57.3.6 `uint32_t BSocketAddressINET::port ()`

Returns socket port.

6.57.3.7 `BString BSocketAddressINET::getString ()`

Return string version of address <ip>:<port>.

6.57.3.8 `BString BSocketAddressINET::getHostName ()` [static]

Get this hosts network name.

6.57.3.9 `BList< uint32_t > BSocketAddressINET::getIpAddresses ()` [static]

Get a list of all the IP addresses of this host.

6.57.3.10 `BList< BString > BSocketAddressINET::getIpAddressList ()` [static]

Get a list of all the IP addresses of this host under hostname.

6.57.3.11 `BList< BString > BSocketAddressINET::getIpAddressListAll ()` [static]

Get a list of all the IP addresses of this host looking at physical interfaces.

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

- [BSocket.h](#)

- [BSocket.cpp](#)

6.58 BString Class Reference

```
#include <BString.h>
```

Public Member Functions

- [BString](#) ()
- [BString](#) (const [BString](#) &string)
- [BString](#) (const char *str)
- [BString](#) (char ch)
- [BString](#) (int v)
- [BString](#) (unsigned int v)
- [BString](#) (long v)
- [BString](#) (unsigned long long)
- [BString](#) (double v)
- [~BString](#) ()
- [BString copy](#) ()
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.

- [BString append](#) (const [BString](#) &str) const
Append a string.

- **BString** & **truncate** (int len)
Truncate to length len.
- **BString** & **pad** (int len)
Pad to length len.
- **BString** & **toUpper** ()
Convert to uppercase.
- **BString** & **toLower** ()
Convert to lowercase.
- void **removeNL** ()
Remove if present NL from last char.
- **BString** **justify** (int leftMargin, int width)
Justify the string to the given width.
- **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.
- void **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.
- **BList**< **BString** > **getTokenList** (**BString** separators)
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.
- [BString dirname \(\)](#)
- [BString basename \(\)](#)
- [BString extension \(\)](#)
- [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+ \(int i\) const](#)
- [BString operator+ \(unsigned int i\) const](#)
- [BString operator+ \(unsigned long long i\) 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 \(int value\)](#)
Converts int to string.
- static [BString convert \(unsigned int value\)](#)
Converts uint to string.
- static [BString convert \(long value\)](#)
Converts long to string.
- static [BString convert \(double value\)](#)

Converts double to string.

- static [BString convert](#) (unsigned long long value)
Converts u long long to string.
- static [BString convertHex](#) (int value)
Converts int to string as hex value.
- static [BString convertHex](#) (unsigned int 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

6.58.1 Constructor & Destructor Documentation

6.58.1.1 BString::BString ()

6.58.1.2 BString::BString (const BString & string)

6.58.1.3 BString::BString (const char * str)

6.58.1.4 BString::BString (char ch)

6.58.1.5 BString::BString (int v)

6.58.1.6 BString::BString (unsigned int v)

6.58.1.7 BString::BString (long v)

6.58.1.8 BString::BString (unsigned long long value)

6.58.1.9 BString::BString (double v)

6.58.1.10 BString::~~BString ()

6.58.2 Member Function Documentation

6.58.2.1 BString BString::convert (char ch) [static]

Converts char to string.

6.58.2.2 BString BString::convert (int *value*) [static]

Converts int to string.

6.58.2.3 BString BString::convert (unsigned int *value*) [static]

Converts uint to string.

6.58.2.4 BString BString::convert (long *value*) [static]

Converts long to string.

6.58.2.5 BString BString::convert (double *value*) [static]

Converts double to string.

6.58.2.6 BString BString::convert (unsigned long long *value*) [static]

Converts u long long to string.

6.58.2.7 BString BString::convertHex (int *value*) [static]

Converts int to string as hex value.

6.58.2.8 BString BString::convertHex (unsigned int *value*) [static]

Converts uint to string as hex value.

6.58.2.9 BString BString::copy ()

Return an independant copy.

6.58.2.10 int BString::len () const

Length of string.

6.58.2.11 const char * BString::retStr () const

Ptr to char* representation.

6.58.2.12 char * BString::retStrDup () const

Ptr to newly malloc'd char*.

6.58.2.13 int BString::retInt () const

Return string as a int.

6.58.2.14 unsigned int BString::retUInt () const

Return string as a int.

6.58.2.15 double BString::retDouble () const

Return string as a double.

6.58.2.16 int BString::compare (const BString & *string*) const

Compare strings.

6.58.2.17 int BString::compareWild (const BString & *string*) const

Compare string to string with wildcards.

6.58.2.18 int BString::compareWildExpression (const BString & *string*) const

Compare string to space delimited patterns.

6.58.2.19 BString BString::append (const BString & *str*) const

Append a string.

6.58.2.20 BString & BString::truncate (int *len*)

Truncate to length len.

6.58.2.21 BString & BString::pad (int *len*)

Pad to length len.

6.58.2.22 BString & BString::toUpper ()

Convert to uppercase.

6.58.2.23 BString & BString::toLower ()

Convert to lowercase.

6.58.2.24 void BString::removeNL ()

Remove if present NL from last char.

6.58.2.25 BString BString::justify (int *leftMargin*, int *width*)

Justify the string to the given width.

6.58.2.26 BString BString::subString (int *start*, int *len*) const

Returns substring.

6.58.2.27 int BString::del (int *start*, int *len*)

Delete substring.

6.58.2.28 int BString::insert (int *start*, BString *str*)

Insert substring.

6.58.2.29 void BString::printf (const char * *fmt*, ...)

Formatted print into the string.

6.58.2.30 int BString::find (char *ch*) const

Find *ch* in string searching forwards.

6.58.2.31 int BString::find (BString *str*) const

Find string in string searching forwards.

6.58.2.32 int BString::findReverse (char *ch*) const

Find *ch* in string searching backwards.

6.58.2.33 BList< BString > BString::getTokenList (BString *separators*)

Break string into tokens.

6.58.2.34 BString BString::removeSeparators (BString *separators*)

Remove any char from separators from string.

6.58.2.35 BString BString::pullToken (BString *terminators*)

Pull token from start of string.

6.58.2.36 BString BString::pullSeparators (BString *separators*)

Pull separators from start of string.

6.58.2.37 BString BString::pullWord ()

Pull a word out of the head of the string.

6.58.2.38 BString BString::pullLine ()

Pull a line out of the head of the string.

6.58.2.39 BString BString::dirname ()**6.58.2.40 BString BString::basename ()****6.58.2.41 BString BString::extension ()****6.58.2.42 BString & BString::operator= (const BString & *string*)****6.58.2.43]**

char & BString::operator[] (int *pos*)

- 6.58.2.44 `int BString::operator==(const BString & s) const` [inline]
- 6.58.2.45 `int BString::operator==(const char * s) const` [inline]
- 6.58.2.46 `int BString::operator>(const BString & s) const` [inline]
- 6.58.2.47 `int BString::operator>(const char * s) const` [inline]
- 6.58.2.48 `int BString::operator<(const BString & s) const` [inline]
- 6.58.2.49 `int BString::operator<(const char * s) const` [inline]
- 6.58.2.50 `int BString::operator>=(const BString & s) const` [inline]
- 6.58.2.51 `int BString::operator<=(const BString & s) const` [inline]
- 6.58.2.52 `int BString::operator!=(const BString & s) const` [inline]
- 6.58.2.53 `int BString::operator!=(const char * s) const` [inline]
- 6.58.2.54 `BString BString::operator+(const BString & s) const` [inline]
- 6.58.2.55 `BString BString::operator+(const char * s) const` [inline]
- 6.58.2.56 `BString BString::operator+=(const BString & s)` [inline]
- 6.58.2.57 `BString BString::operator+=(const char * s)` [inline]
- 6.58.2.58 `BString BString::operator+(char ch) const` [inline]
- 6.58.2.59 `BString BString::operator+(int i) const` [inline]
- 6.58.2.60 `BString BString::operator+(unsigned int i) const` [inline]
- 6.58.2.61 `BString BString::operator+(unsigned long long i) const` [inline]
- 6.58.2.62 `BString::operator const char * () const` [inline]
- 6.58.2.63 `BString BString::field(int field) const`
- 6.58.2.64 `char ** BString::fields()`
- 6.58.2.65 `void BString::Init(const char * str)` [private]
- 6.58.2.66 `int BString::inString(int pos) const` [private]
- 6.58.2.67 `int BString::isSpace(char ch) const` [private]

6.58.3 Member Data Documentation

- 6.58.3.1 `BRefData* BString::ostr` [protected]

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

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

6.59 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](#)

6.59.1 Constructor & Destructor Documentation

6.59.1.1 [BStringLocked::BStringLocked \(\)](#) [inline]

6.59.1.2 [BStringLocked::BStringLocked \(const BStringLocked & s\)](#) [inline]

6.59.1.3 [BStringLocked::BStringLocked \(const BString & s\)](#) [inline]

6.59.2 Member Function Documentation

6.59.2.1 [int BStringLocked::len \(\) const](#) [inline]

Length of string.

6.59.2.2 [BStringLocked::operator BString \(\) const](#) [inline]

6.59.2.3 [BStringLocked BStringLocked::operator+ \(const BStringLocked & s\) const](#) [inline]

6.59.2.4 [BStringLocked& BStringLocked::operator= \(const BStringLocked & s\)](#) [inline]

6.59.3 Member Data Documentation

6.59.3.1 [BStringMutex BStringLocked::olock](#) [mutable, private]

6.59.3.2 [BString BStringLocked::ostr](#) [private]

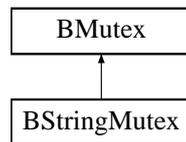
The documentation for this class was generated from the following file:

- [BStringLocked.h](#)

6.60 BStringMutex Class Reference

```
#include <BStringLocked.h>
```

Inheritance diagram for BStringMutex::



Public Member Functions

- [BStringMutex \(\)](#)

6.60.1 Constructor & Destructor Documentation

6.60.1.1 BStringMutex::BStringMutex () [inline]

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

- [BStringLocked.h](#)

6.61 BTable Class Reference

```
#include <BTable.h>
```

Public Member Functions

- [BTable \(\)](#)
- [~BTable \(\)](#)
- 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](#)

6.61.1 Constructor & Destructor Documentation

6.61.1.1 [BTable::BTable \(\)](#)

6.61.1.2 [BTable::~~BTable \(\)](#)

6.61.2 Member Function Documentation

6.61.2.1 void [BTable::setTitle](#) ([BArray](#)< [BString](#) > *title*)

6.61.2.2 void [BTable::addRow](#) ([BArray](#)< [BString](#) > *data*)

6.61.2.3 void [BTable::print](#) ()

6.61.2.4 void [BTable::calculateWidths](#) () [private]

6.61.2.5 void [BTable::printLine](#) ([BArray](#)< [BString](#) > *line*, int *comment* = 0) [private]

6.61.3 Member Data Documentation

6.61.3.1 [BArray](#)<[BString](#)> [BTable::otitle](#) [private]

6.61.3.2 [BList](#)<[BArray](#)<[BString](#)> > [BTable::odata](#) [private]

6.61.3.3 [BArray](#)<int> [BTable::ocolumnWidths](#) [private]

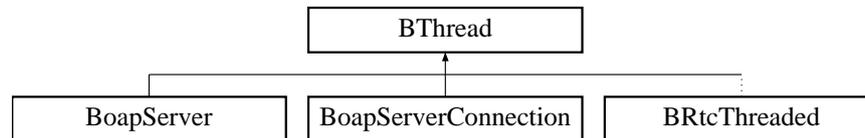
The documentation for this class was generated from the following files:

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

6.62 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 * oresult](#)

6.62.1 Constructor & Destructor Documentation

6.62.1.1 `BThread::BThread ()`

6.62.1.2 `BThread::~~BThread ()` [virtual]

6.62.2 Member Function Documentation

6.62.2.1 `int BThread::setInitPriority (int policy, int priority)`

6.62.2.2 `int BThread::setInitStackSize (size_t stackSize)`

6.62.2.3 `int BThread::start ()`

6.62.2.4 `void * BThread::result ()`

6.62.2.5 `int BThread::running ()`

6.62.2.6 `int BThread::setPriority (int policy, int priority)`

6.62.2.7 `int BThread::cancel ()`

6.62.2.8 `void * BThread::waitForCompletion ()`

6.62.2.9 `pthread_t BThread::getThread ()`

6.62.2.10 `void * BThread::function ()` [virtual]

Reimplemented in [BoapServerConnection](#), [BoapServer](#), and [BRtcThreaded](#).

6.62.2.11 `void * BThread::startFunc (void * arg)` [static, private]

6.62.3 Member Data Documentation

6.62.3.1 `pthread_t BThread::othread` [private]

6.62.3.2 `size_t BThread::ostackSize` [private]

6.62.3.3 `int BThread::opolicy` [private]

6.62.3.4 `int BThread::opriority` [private]

6.62.3.5 `int BThread::orunning` [private]

6.62.3.6 `void* BThread::oreult` [private]

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

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

6.63 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](#)

6.63.1 Detailed Description

Stopwatch style timer.

6.63.2 Constructor & Destructor Documentation

6.63.2.1 BTimer::BTimer ()

6.63.2.2 BTimer::~~BTimer ()

6.63.3 Member Function Documentation

6.63.3.1 void BTimer::start ()

Start timer.

6.63.3.2 void BTimer::stop ()

Stop timer.

6.63.3.3 void BTimer::clear ()

Clear timer.

6.63.3.4 double BTimer::getElapsedTime ()

Returns the elapsed time from the last start.

6.63.3.5 void BTimer::add (BTimer & *timer*)

Add two timers.

6.63.3.6 double BTimer::average ()

Average time is duration between [start\(\)](#) and [stop\(\)](#) / number of stops.

6.63.3.7 double BTimer::peak ()

Peak time.

6.63.3.8 `double BTimer::getTime ()` [static, private]

6.63.4 Member Data Documentation

6.63.4.1 `BMutex BTimer::oLock` [private]

6.63.4.2 `unsigned int BTimer::oNum` [private]

6.63.4.3 `double BTimer::oStartTime` [private]

6.63.4.4 `double BTimer::oEndTime` [private]

6.63.4.5 `double BTimer::oAverage` [private]

6.63.4.6 `double BTimer::oPeak` [private]

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

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

6.64 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](#) (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](#) ()
- int [yday](#) ()
- int [month](#) ()
- int [day](#) ()
- int [hour](#) ()
- int [minute](#) ()
- int [second](#) ()
- int [microSecond](#) ()
- void [getDate](#) (int &year, int &mon, int &day)
- [BString](#) [getString](#) (BString separator="T")
Get the time as an ISO date/time string.
- [BError](#) [setString](#) (BString dateTime)
Set the time from an ISO date/time.
- [BString](#) [getStringNoMs](#) (BString separator="T")

Get the time as an ISO date/time string without microseconds.

- [BString getStringFormatted](#) ([BString](#) format)
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](#) (int 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](#) ()
Get number of seconds within the year.
- uint64_t [getYearMicroSeconds](#) ()
Get number of micro seconds within the year.
- int [isSet](#) ()
- int [compare](#) (const [BTimeStamp](#) &timeStamp)
Compare two dates.
- [operator BString](#) ()
- [BTimeStamp](#) & [operator=](#) (const [BTimeStampMs](#) &timeStamp)
- int [operator==](#) (const [BTimeStamp](#) &timeStamp)
- int [operator!=](#) (const [BTimeStamp](#) &timeStamp)
- int [operator>](#) (const [BTimeStamp](#) &timeStamp)
- int [operator>=](#) (const [BTimeStamp](#) &timeStamp)
- int [operator<](#) (const [BTimeStamp](#) &timeStamp)
- int [operator<=](#) (const [BTimeStamp](#) &timeStamp)

Static Public Member Functions

- static int [isLeap](#) (int year)
- static [BUInt64](#) [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).

6.64.1 Constructor & Destructor Documentation

6.64.1.1 BTimeStamp::BTimeStamp ()

6.64.1.2 BTimeStamp::BTimeStamp (int *year*, int *month* = 1, int *day* = 1, int *hour* = 0, int *minute* = 0, int *second* = 0, int *microsecond* = 0)

6.64.1.3 BTimeStamp::BTimeStamp (BString *str*)

6.64.1.4 BTimeStamp::~~BTimeStamp ()

6.64.2 Member Function Documentation

6.64.2.1 void BTimeStamp::clear ()

Clear the date/time.

6.64.2.2 void BTimeStamp::setFirst ()

Set the first date available.

6.64.2.3 void BTimeStamp::setLast ()

Set the last date available.

6.64.2.4 void BTimeStamp::set (time_t *time*, int *microSeconds*)

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

6.64.2.5 void BTimeStamp::set (int *year* = 0, int *month* = 1, int *day* = 1, int *hour* = 0, int *minute* = 0, int *second* = 0, int *microsecond* = 0)

6.64.2.6 void BTimeStamp::set (const BTimeStampMs & *timeStamp*)

Set the timeStamp to given MS time stamp.

6.64.2.7 void BTimeStamp::setYDay (int *year* = 0, int *yday* = 0, int *hour* = 0, int *minute* = 0, int *second* = 0, int *microsecond* = 0)

6.64.2.8 void BTimeStamp::setTime (int *hour* = 0, int *minute* = 0, int *second* = 0, int *microsecond* = 0)

6.64.2.9 void BTimeStamp::setNow ()

Set the timeStamp to now.

6.64.2.10 int BTimeStamp::year ()

6.64.2.11 int BTimeStamp::yday ()

6.64.2.12 int BTimeStamp::month ()

6.64.2.13 int BTimeStamp::day ()

6.64.2.14 int BTimeStamp::hour ()

6.64.2.15 int BTimeStamp::minute ()

6.64.2.16 int BTimeStamp::second ()

6.64.2.17 int BTimeStamp::microSecond ()

6.64.2.18 void BTimeStamp::getDate (int & *year*, int & *mon*, int & *day*)

6.64.2.19 BString BTimeStamp::getString (BString *separator* = "T")

Get the time as an ISO date/time string.

6.64.2.20 BError BTimeStamp::setString (BString *dateTime*)

Set the time from an ISO date/time.

6.64.2.21 BString BTimeStamp::getStringNoMs (BString *separator* = "T")

Get the time as an ISO date/time string without microseconds.

6.64.2.22 BString BTimeStamp::getStringFormatted (BString *format*)

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

6.64.2.23 void BTimeStamp::addMilliseconds (int *milliseconds*)

Add the given number of milli seconds. This should be less than a year.

6.64.2.24 void BTimeStamp::addMicroSeconds (int *microSeconds*)

Add the given number of micro seconds. This should be less than a year.

6.64.2.25 void BTimeStamp::addSeconds (int *seconds*)

Add the given number of seconds. This should be less than a year.

6.64.2.26 uint32_t BTimeStamp::getYearSeconds ()

Get number of seconds within the year.

6.64.2.27 uint64_t BTimeStamp::getYearMicroSeconds ()

Get number of micro seconds within the year.

6.64.2.28 int BTimeStamp::isSet () [inline]**6.64.2.29 int BTimeStamp::compare (const BTimeStamp & *timeStamp*)**

Compare two dates.

6.64.2.30 BTimeStamp::operator BString () [inline]**6.64.2.31 BTimeStamp& BTimeStamp::operator= (const BTimeStampMs & *timeStamp*) [inline]****6.64.2.32 int BTimeStamp::operator== (const BTimeStamp & *timeStamp*) [inline]****6.64.2.33 int BTimeStamp::operator!= (const BTimeStamp & *timeStamp*) [inline]****6.64.2.34 int BTimeStamp::operator> (const BTimeStamp & *timeStamp*) [inline]****6.64.2.35 int BTimeStamp::operator>= (const BTimeStamp & *timeStamp*) [inline]****6.64.2.36 int BTimeStamp::operator< (const BTimeStamp & *timeStamp*) [inline]****6.64.2.37 int BTimeStamp::operator<= (const BTimeStamp & *timeStamp*) [inline]****6.64.2.38 int BTimeStamp::isLeap (int *year*) [static]****6.64.2.39 BUInt64 BTimeStamp::difference (BTimeStamp *t2*, BTimeStamp *t1*) [static]****6.64.3 Member Data Documentation****6.64.3.1 uint16_t BTimeStamp::oyear**

Year (0 .. 65535).

6.64.3.2 uint16_t BTimeStamp::oyday

Day in year (0 .. 365).

6.64.3.3 uint8_t BTimeStamp::ohour

Hour (0 .. 23).

6.64.3.4 uint8_t BTimeStamp::ominate

Minute (0 .. 59).

6.64.3.5 uint8_t BTimeStamp::osecond

Second (0 .. 59).

6.64.3.6 uint8_t BTimeStamp::ospare

Padding.

6.64.3.7 uint32_t BTimeStamp::omicroSecond

MicroSecond (0 .. 999999).

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

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

6.65 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.

6.65.1 Constructor & Destructor Documentation

6.65.1.1 BTimeStampMs::BTimeStampMs (BString *str* = " ")

6.65.1.2 BTimeStampMs::~~BTimeStampMs ()

6.65.2 Member Function Documentation

6.65.2.1 void BTimeStampMs::clear ()

Clear the date/time.

6.65.2.2 void BTimeStampMs::setNow ()

Set the timeStamp to now.

6.65.2.3 BTimeStampMs & BTimeStampMs::addMilliseconds (int *milliseconds*)

Add the given number of milli seconds. This should be less than a year.

6.65.2.4 BTimeStampMs & BTimeStampMs::subMilliseconds (int *milliseconds*)

Add the given number of milli seconds. This should be less than a year.

6.65.2.5 BTimeStampMs & BTimeStampMs::addSeconds (int *seconds*)

Add the given number of seconds. This should be less than a year.

6.65.2.6 BTimeStampMs & BTimeStampMs::subSeconds (int *seconds*)

Subtract the given number of seconds. This should be less than a year.

6.65.2.7 uint32_t BTimeStampMs::getYearSeconds ()

Get number of seconds within the year.

6.65.2.8 uint64_t BTimeStampMs::getYearMilliseconds ()

Get number of seconds within the year.

6.65.2.9 BString BTimeStampMs::getString (BString *separator* = "T")

Get the time as an ISO date/time string.

6.65.2.10 BString BTimeStampMs::getStringNoMs (BString *separator* = "T")

Get the time as an ISO date/time string with no ms.

6.65.2.11 BError BTimeStampMs::setString (BString *dateTime*)

Set the time from an ISO date/time.

6.65.2.12 BString BTimeStampMs::getDurationString (BString *separator* = "T")

Get the time as an ISO date/time string but with month's and days starting from 0.

6.65.2.13 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.

6.65.2.14 BError BTimeStampMs::setDurationString (BString *dateTime*)

Set the time from an ISO date/time string but with month's and days starting from 0.

6.65.2.15 BString BTimeStampMs::getStringRaw ()**6.65.2.16 void BTimeStampMs::getDate (int & *year*, int & *mon*, int & *day*)**

Get the year, month and day.

6.65.2.17 int BTimeStampMs::compare (const BTimeStampMs & *timeStamp*)

Compare two dates.

6.65.2.18 int BTimeStampMs::operator> (const BTimeStampMs & *timeStamp*) [inline]**6.65.2.19 int BTimeStampMs::operator>= (const BTimeStampMs & *timeStamp*) [inline]****6.65.2.20 int BTimeStampMs::operator< (const BTimeStampMs & *timeStamp*) [inline]****6.65.2.21 int BTimeStampMs::operator<= (const BTimeStampMs & *timeStamp*) [inline]****6.65.2.22 int BTimeStampMs::isLeap (int *year*) [static]****6.65.2.23 BUInt64 BTimeStampMs::difference (BTimeStampMs *t2*, BTimeStampMs *t1*) [static]****6.65.3 Member Data Documentation****6.65.3.1 uint16_t BTimeStampMs::year**

Year (2000 .. 3000).

6.65.3.2 uint16_t BTimeStampMs::yday

Day in year (0 .. 365).

6.65.3.3 uint16_t BTimeStampMs::hour

Hour (0 .. 23).

6.65.3.4 uint16_t BTimeStampMs::minute

Minute (0 .. 59).

6.65.3.5 uint16_t BTimeStampMs::second

Second (0 .. 59).

6.65.3.6 uint16_t BTimeStampMs::milliSecond

MilliSecond (0 .. 999).

6.65.3.7 int32_t BTimeStampMs::sampleNumber

The sample number this time refers to.

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

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

6.66 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](#)

6.66.1 Detailed Description

Basic access to a Url.

6.66.2 Constructor & Destructor Documentation

6.66.2.1 BUrl::BUrl ()

6.66.2.2 BUrl::~BUrl ()

6.66.3 Member Function Documentation

6.66.3.1 BError BUrl::readString (BString url, BString & str)

Reads URL.

6.66.3.2 `size_t BUrl::writeData (void * data, size_t size, size_t elSize, void * stream)` [static, private]

6.66.4 Member Data Documentation

6.66.4.1 `int BUrl::oinit` [static, private]

6.66.4.2 `BString BUrl::ores` [private]

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

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

6.67 vector Class Reference

Inherited by [BArray< BString >](#), and [BArray< int >](#).

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

- [BArray.h](#)

Chapter 7

File Documentation

7.1 BArray.h File Reference

```
#include <BTypes.h>  
#include <vector>
```

Classes

- class [BArray< T >](#)

Defines

- #define [BArray_H 1](#)

7.1.1 Define Documentation

7.1.1.1 #define BArray_H 1

7.2 BAtomicCount.h File Reference

```
#include <bits/atomicity.h>
```

Classes

- class [BAtomicCount](#)
BAtomicCount class.

Defines

- #define [BAtomicCount_H](#) 1

7.2.1 Define Documentation

7.2.1.1 #define BAtomicCount_H 1

7.3 BBuffer.cpp File Reference

```
#include <stdlib.h>
#include <memory.h>
#include <BBuffer.h>
#include <byteswap.h>
```

Variables

- const int `roundSize` = 256

7.3.1 Variable Documentation

7.3.1.1 const int roundSize = 256

7.4 BBuffer.h File Reference

```
#include <BTypes.h>
#include <BString.h>
#include <BError.h>
#include <BTimeStamp.h>
```

Classes

- class [BBuffer](#)
- class [BBufferStore](#)

Defines

- #define [BBUFFER_H](#) 1

7.4.1 Define Documentation

7.4.1.1 #define BBUFFER_H 1

7.5 BCond.cpp File Reference

```
#include <BCond.h>
#include <sys/time.h>
#include <stdio.h>
```

7.6 BCond.h File Reference

```
#include <pthread.h>
```

Classes

- class [BCond](#)

Defines

- #define [BCOND_H](#) 1

7.6.1 Define Documentation

7.6.1.1 #define BCOND_H 1

7.7 BCondInt.cpp File Reference

```
#include <BCondInt.h>
#include <sys/time.h>
#include <stdio.h>
#include <errno.h>
```

7.8 BCondInt.h File Reference

```
#include <BTypes.h>
#include <pthread.h>
```

Classes

- class [BCondValue](#)
Thread conditional value.
- class [BCondInt](#)
Thread conditional integer.
- class [BCondBool](#)
Thread conditional boolean.
- class [BCondWrap](#)

Defines

- #define [BCONDINT_H](#) 1

7.8.1 Define Documentation

7.8.1.1 #define BCONDINT_H 1

7.9 BConfig.cpp File Reference

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

7.10 BConfig.h File Reference

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

Classes

- class [BConfig](#)
This class implements the configuration file access.

7.11 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 <BDebug.h>
#include <errno.h>
#include <linux/unistd.h>
```

Defines

- #define [BTRACE_SIZE](#) 100

Functions

- void [hd8](#) (void *data, int n)
- void [hda8](#) (void *data, int n)
- void [hd32](#) (void *data, int n)
- void [hda32](#) (void *data, int n)
- double [getTime](#) ()
- void [setDebug](#) (int d)
- void [tprintf](#) (int log, const char *fmt,...)
- pid_t [gettid](#) ()

Variables

- int [bdebug](#)
- const unsigned int [STRBUF_SIZE](#) = (64 * 1024)

7.11.1 Define Documentation

7.11.1.1 `#define BTRACE_SIZE 100`

7.11.2 Function Documentation

7.11.2.1 `pid_t gettid ()`

7.11.2.2 `double getTime ()`

7.11.2.3 `void hd32 (void * data, int n)`

7.11.2.4 `void hd8 (void * data, int n)`

7.11.2.5 `void hda32 (void * data, int n)`

7.11.2.6 `void hda8 (void * data, int n)`

7.11.2.7 `void setDebug (int d)`

7.11.2.8 `void tprintf (int log, const char * fmt, ...)`

7.11.3 Variable Documentation

7.11.3.1 `int bdebug`

7.11.3.2 `const unsigned int STRBUF_SIZE = (64 * 1024)`

7.12 BDebug.h File Reference

```
#include <stdio.h>
#include <syslog.h>
#include <time.h>
```

Classes

- class [BDebugBacktrace](#)

Defines

- #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, int n)
- void [hda8](#) (void *data, int n)
- void [hd32](#) (void *data, int n)
- void [hds32](#) (void *data, int n)
- double [getTime](#) ()
- void [setDebug](#) (int debug)
- void [tprintf](#) (int log, const char *fmt,...)
- pid_t [gettid](#) ()

Variables

- int [bdebug](#)

7.12.1 Define Documentation

7.12.1.1 #define [BDebug_STD](#) 0x000001

7.12.1.2 #define [dprintf](#)(level, fmt, a...)

General debug functions.

7.12.1.3 `#define eprintf(fmt, a...) syslog(LOG_ERR, fmt, ##a)`

7.12.1.4 `#define nprintf(fmt, a...) syslog(LOG_NOTICE, fmt, ##a)`

Warnings and errors logging.

7.12.1.5 `#define wprintf(fmt, a...) syslog(LOG_WARNING, fmt, ##a)`

7.12.2 Function Documentation

7.12.2.1 `pid_t gettid ()`

7.12.2.2 `double getTime ()`

7.12.2.3 `void hd32 (void * data, int n)`

7.12.2.4 `void hd8 (void * data, int n)`

7.12.2.5 `void hda8 (void * data, int n)`

7.12.2.6 `void hds32 (void * data, int n)`

7.12.2.7 `void setDebug (int debug)`

7.12.2.8 `void tprintf (int log, const char *fmt, ...)`

7.12.3 Variable Documentation

7.12.3.1 `int bdebug`

7.13 BDict.cpp File Reference

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

Functions

- void [toBString](#) ([BDictString](#) &*v*, [BString](#) &*s*)
- void [fromBString](#) ([BString](#) &*str*, [BDictString](#) &*v*)

7.13.1 Function Documentation

7.13.1.1 void [fromBString](#) ([BString](#) & *str*, [BDictString](#) & *v*)

7.13.1.2 void [toBString](#) ([BDictString](#) & *v*, [BString](#) & *s*)

7.14 BDict.h File Reference

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

Classes

- class [BDictItem< Type >](#)
Template based Dictionary class.
- class [BDict< Type >](#)

Defines

- #define [BDict_H 1](#)

Typedefs

- typedef [BDict< BString > BDictString](#)

Functions

- void [toBString](#) ([BDictString](#) &v, [BString](#) &s)
- void [fromBString](#) ([BString](#) &s, [BDictString](#) &v)

7.14.1 Define Documentation

7.14.1.1 #define [BDict_H 1](#)

7.14.2 Typedef Documentation

7.14.2.1 typedef [BDict<BString> BDictString](#)

7.14.3 Function Documentation

7.14.3.1 void [fromBString](#) ([BString](#) & s, [BDictString](#) & v)

7.14.3.2 void [toBString](#) ([BDictString](#) & v, [BString](#) & s)

7.15 BDictMap.h File Reference

```
#include <BString.h>
#include <map>
```

Classes

- class [BDictMap< Value >](#)

Defines

- #define [BDictMap_H 1](#)

Typedefs

- typedef [BDictMap< BString >](#) [BDictMapString](#)

7.15.1 Define Documentation

7.15.1.1 #define [BDictMap_H 1](#)

7.15.2 Typedef Documentation

7.15.2.1 typedef [BDictMap<BString>](#) [BDictMapString](#)

7.16 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](#)

7.16.1 Function Documentation

7.16.1.1 `static int wild (const dirent * e)` [static]

7.16.2 Variable Documentation

7.16.2.1 `BString wildString` [static]

7.17 BDir.h File Reference

```
#include <BList.h>
#include <BString.h>
#include <BError.h>
#include <sys/stat.h>
```

Classes

- class [BDir](#)
File system directory class.

Defines

- #define [BDIR_H](#) 1

7.17.1 Define Documentation

7.17.1.1 #define BDIR_H 1

7.18 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>
```

7.19 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.

7.20 BError.cpp File Reference

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

7.21 BError.h File Reference

```
#include <BString.h>
```

Classes

- class [BError](#)
Error return class.

Defines

- #define [BERROR_H 1](#)

7.21.1 Define Documentation

7.21.1.1 #define BERROR_H 1

7.22 BEvent.cpp File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <BEvent.h>
#include <BPoll.h>
```

7.23 BEvent.h File Reference

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

Classes

- class [BEvent](#)

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

- class [BEventError](#)
- class [BEventPipe](#)

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 [BEventInt](#)

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.

Defines

- #define [BEvent_H](#) 1

Enumerations

- enum [BEventType](#) { [BEventTypeNone](#), [BEventTypeInt](#), [BEventTypeError](#) }

7.23.1 Define Documentation

7.23.1.1 #define BEvent_H 1

7.23.2 Enumeration Type Documentation

7.23.2.1 enum BEventType

Enumerator:

BEventTypeNone

BEventTypeInt

BEventTypeError

7.24 BFifo.cpp File Reference

```
#include <BFifo.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/mman.h>
```

Defines

- #define [DEBUG](#) 0
- #define [dprintf](#)(fmt, a...)

7.24.1 Define Documentation

7.24.1.1 #define [DEBUG](#) 0

7.24.1.2 #define [dprintf](#)(fmt, a...)

7.25 BFifo.h File Reference

```
#include <stdint.h>
#include <BError.h>
#include <BCondInt.h>
#include <BMutex.h>
#include <BFifo.inc>
```

Classes

- class [BFifoPos](#)
This class implements a pointer into the Fifo's circular buffer.
- class [BFifo< Type >](#)
This class implements a thread safe FIFO buffer.

Defines

- `#define BFIFO_H 1`

7.25.1 Define Documentation

7.25.1.1 `#define BFIFO_H 1`

7.26 BFifo.inc File Reference

7.27 BFile.cpp File Reference

```
#include <stdarg.h>
#include <BFile.h>
#include <sys/stat.h>
#include <string.h>
#include <errno.h>
```

Defines

- #define [STRBUF](#) 10240

7.27.1 Define Documentation

7.27.1.1 #define STRBUF 10240

7.28 BFile.h File Reference

```
#include <stdio.h>
#include <BTypes.h>
#include <BString.h>
#include <BError.h>
```

Classes

- class [BFile](#)
File operations class.

Defines

- #define [BFILE_H](#) 1

7.28.1 Define Documentation

7.28.1.1 #define BFILE_H 1

7.29 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](#)

Defines

- #define [BLIST_H](#) 1
- #define [BListLoop](#)(list, i) for([BIter](#) i = list.begin(); !list.isEnd(i); list.next(i))

7.29.1 Define Documentation

7.29.1.1 #define [BLIST_H](#) 1

7.29.1.2 #define [BListLoop](#)(list, i) for([BIter](#) i = list.begin(); !list.isEnd(i); list.next(i))

7.30 BList_func.h File Reference

```
#include <stdlib.h>  
#include <stdio.h>  
#include <memory.h>
```

7.31 BMutex.cpp File Reference

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

Defines

- #define [MDEBUG](#) 0

7.31.1 Define Documentation

7.31.1.1 #define MDEBUG 0

7.32 BMutex.h File Reference

```
#include <pthread.h>
```

Classes

- class [BMutex](#)
Mutex class.
- class [BMutexLock](#)

Defines

- #define [BMUTEX_H](#) 1

7.32.1 Define Documentation

7.32.1.1 #define BMUTEX_H 1

7.33 BMySQL.cpp File Reference

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

7.34 BMySQL.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BDict.h>
#include <BMutex.h>
#include <mysql/mysql.h>
```

Classes

- class [BMySQL](#)

Defines

- #define [BMySQL_H](#) 1

7.34.1 Define Documentation

7.34.1.1 #define BMySQL_H 1

7.35 BNameValue.h File Reference

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

Classes

- class [BNameValue< T >](#)
- class [BNameValueList< T >](#)

Defines

- `#define BNAMEVALUE_H 1`

7.35.1 Define Documentation

7.35.1.1 `#define BNAMEVALUE_H 1`

7.36 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>
```

Defines

- #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.

7.36.1 Define Documentation

7.36.1.1 #define [APIVERSION_TEST](#) 1

7.36.1.2 #define [DEBUG](#) 0

7.36.1.3 #define [dprintf](#)(fmt, a...)

7.36.1.4 #define [IS_BIG_ENDIAN](#) 1

7.36.2 Variable Documentation

7.36.2.1 const int [boapPort](#) = 12000

The default BOAP connection port.

7.37 Boap.h File Reference

```
#include <stdint.h>
#include <BTypes.h>
#include <BPoll.h>
#include <BSocket.h>
#include <BThread.h>
#include <BError.h>
#include <BEvent.h>
#include <BMutex.h>
#include <BTimeStamp.h>
#include <BBuffer.h>
```

Namespaces

- namespace [Boapns](#)

Classes

- struct [BoapPacketHead](#)
- class [BoapPacket](#)
- class [BoapClientObject](#)
- class [BoapSignalObject](#)
- class [BoapServiceEntry](#)
- class [BoapServerConnection](#)
- class [BoapServer](#)
- class [BoapFuncEntry](#)
- class [BoapServiceObject](#)

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

7.37.1 Typedef Documentation

7.37.1.1 `typedef BError(BoapServiceObject::* BoapFunc)(BoapServerConnection *conn, BoapPacket &rx, BoapPacket &tx)`

7.37.1.2 `typedef BUInt32 BoapService`

7.37.2 Enumeration Type Documentation

7.37.2.1 `enum BoapPriority`

Enumerator:

BoapPriorityLow

BoapPriorityNormal

BoapPriorityHigh

7.37.2.2 `enum BoapType`

Enumerator:

BoapTypeRpc

BoapTypeRpcReply

BoapTypeSignal

BoapTypeRpcError

BoapTypeRpc

BoapTypeSignal

7.37.3 Variable Documentation

7.37.3.1 `const BUInt32 BoapMagic = 0x424F4100`

7.38 BoapnsC.cc File Reference

```
#include <BoapnsC.h>
```

Namespaces

- namespace [Boapns](#)

Functions

- [Boapns::Boapns](#) (BString name)
- [BError Boapns::getVersion](#) (BString &version)
- [BError Boapns::getEntryList](#) (BList< BoapEntry > &entryList)
- [BError Boapns::getEntry](#) (BString name, BoapEntry &entry)
- [BError Boapns::addEntry](#) (BoapEntry entry)
- [BError Boapns::delEntry](#) (BString name)
- [BError Boapns::getNewName](#) (BString &name)

7.39 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>
```

Namespaces

- namespace [Boapns](#)

Classes

- class [Boapns::Boapns](#)

Defines

- #define [BOAPNSC_H 1](#)

Variables

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

7.39.1 Define Documentation

7.39.1.1 #define BOAPNSC_H 1

7.40 BoapnsD.cc File Reference

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

Namespaces

- namespace [Boapns](#)

7.41 BoapnsD.h File Reference

```
#include <Boap.h>
#include <BObj.h>
#include <BList.h>
#include <BArray.h>
```

Namespaces

- namespace [Boapns](#)

Classes

- class [Boapns::BoapEntry](#)

Defines

- #define [BOAPNSD_H](#) 1

7.41.1 Define Documentation

7.41.1.1 #define BOAPNSD_H 1

7.42 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>
```

Defines

- #define `DEBUG` 0
- #define `dprintf`(fmt, a...)

Variables

- const int `roundSize` = 256

7.42.1 Define Documentation

7.42.1.1 #define `DEBUG` 0

7.42.1.2 #define `dprintf`(fmt, a...)

7.42.2 Variable Documentation

7.42.2.1 const int `roundSize` = 256

7.43 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](#) }

7.43.1 Typedef Documentation

7.43.1.1 typedef BError(BoapServiceObject::* BoapFunc)(BoapPacket &rx, BoapPacket &tx)

7.43.1.2 typedef uint32_t BoapService

7.43.1.3 typedef double Double

7.43.1.4 typedef int16_t Int16

7.43.1.5 typedef int32_t Int32

7.43.1.6 typedef int8_t Int8

7.43.1.7 typedef uint16_t UInt16

7.43.1.8 typedef uint32_t UInt32

7.43.1.9 typedef uint8_t UInt8

7.43.2 Enumeration Type Documentation

7.43.2.1 enum BoapType

Enumerator:

BoapTypeRpc

BoapTypeRpcReply

BoapTypeSignal

BoapTypeRpcError

BoapTypeRpc

BoapTypeSignal

7.44 BObj.cpp File Reference

```
#include <BObj.h>
```

7.45 BObj.h File Reference

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

Classes

- class [BObj](#)

Defines

- #define [BObj_H 1](#)

7.45.1 Define Documentation

7.45.1.1 #define BObj_H 1

7.46 BObject.cc File Reference

```
#include <stdio.h>
#include <ctype.h>
#include <memory.h>
#include <string.h>
#include <BObject.h>
#include <iostream>
```

Defines

- #define [DEBUG](#) 0

7.46.1 Define Documentation

7.46.1.1 #define [DEBUG](#) 0

7.47 BObject.h File Reference

```
#include <BType.h>
#include <BDataBuf.h>
#include <BString.h>
#include <BNameValue.h>
#include <BList.h>
#include <BError.h>
```

Classes

- class [BObject](#)

Defines

- #define [BOBJECT_H](#) 1

Typedefs

- typedef [BNameValue](#)< [BObject](#) * > [BMember](#)
- typedef [BNameValueList](#)< [BObject](#) * > [BMemberList](#)

7.47.1 Define Documentation

7.47.1.1 #define [BOBJECT_H](#) 1

7.47.2 Typedef Documentation

7.47.2.1 typedef [BNameValue](#)<[BObject](#)*> [BMember](#)

7.47.2.2 typedef [BNameValueList](#)<[BObject](#)*> [BMemberList](#)

7.48 BPoll.cpp File Reference

```
#include <stdlib.h>  
#include <unistd.h>  
#include <errno.h>  
#include <BPoll.h>
```

7.49 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.

Defines

- #define [BPOLL_H](#) 1

7.49.1 Define Documentation

7.49.1.1 #define BPOLL_H 1

7.50 BRefData.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <BRefData.h>
```

Defines

- #define [DEBUG](#) 0
- #define [CHUNK](#) 16

7.50.1 Define Documentation

7.50.1.1 #define [CHUNK](#) 16

7.50.1.2 #define [DEBUG](#) 0

7.51 BRefData.h File Reference

```
#include <BAAtomicCount.h>
```

Classes

- class [BRefData](#)

Defines

- #define [BREFDATA_H](#) 1

7.51.1 Define Documentation

7.51.1.1 #define BREFDATA_H 1

7.52 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>
```

7.53 BRtc.h File Reference

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

Classes

- class [BRtc](#)
Realtime clock.
- class [BRtcThreaded](#)
Threaded real time clock.

7.54 BRWLock.cpp File Reference

```
#include <BRWLock.h>
```

7.55 BRWLock.h File Reference

```
#include <pthread.h>
```

Classes

- class [BRWLock](#)
thread read-write locks

Defines

- #define [BRWLOCK_H](#) 1

7.55.1 Define Documentation

7.55.1.1 #define BRWLOCK_H 1

7.56 BSema.cpp File Reference

```
#include <BSema.h>
#include <errno.h>
#include <sys/time.h>
```

7.57 BSema.h File Reference

```
#include <semaphore.h>
```

Classes

- class [BSema](#)
Sempahore class.

Defines

- #define [BSEMA_H](#) 1

7.57.1 Define Documentation

7.57.1.1 #define BSEMA_H 1

7.58 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"
```

Defines

- #define [IP_MTU](#) 14

7.58.1 Define Documentation

7.58.1.1 #define IP_MTU 14

7.59 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](#)

Defines

- #define [BSOCKET_H 1](#)

7.59.1 Define Documentation

7.59.1.1 #define BSOCKET_H 1

7.60 BString.cpp File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdarg.h>
#include <ctype.h>
#include "BString.h"
```

Defines

- #define `DEBUG` 0
- #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)
- `BStringList` `charToList` (char **str)
- `BString` `barrayToString` (const `BStringArray` &list)
- `BStringArray` `bstringToArray` (`BString` str)
- `BStringArray` `charToArray` (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)

7.60.1 Define Documentation

7.60.1.1 `#define DEBUG 0`

7.60.1.2 `#define MINUS '-'`

7.60.1.3 `#define STRIP 0x7f`

7.60.2 Function Documentation

7.60.2.1 `BString barrayToString (const BStringArray & list)`

7.60.2.2 `BString blistToString (const BStringList & list)`

7.60.2.3 `int bstringListinList (BStringList & list, BString s)`

7.60.2.4 `BStringArray bstringToArray (BString str)`

7.60.2.5 `BStringList bstringToList (BString str)`

7.60.2.6 `BStringArray charToArray (char ** str)`

7.60.2.7 `BStringList charToList (char ** str)`

7.60.2.8 `void fromBString (BString & s, BFloat64 & v)`

7.60.2.9 `void fromBString (BString & s, BUInt64 & v)`

7.60.2.10 `void fromBString (BString & s, BUInt32 & v)`

7.60.2.11 `void fromBString (BString & s, BInt32 & v)`

7.60.2.12 `void fromBString (BString & s, BStringList & v)`

7.60.2.13 `void fromBString (BString & s, BString & v)`

7.60.2.14 `static int gmatch (const char * s, const char * p) [static]`

7.60.2.15 `std::ostream& operator<<< (std::ostream & o, BString & s)`

7.60.2.16 `std::istream& operator>>> (std::istream & i, BString & s)`

7.60.2.17 `void toBString (BFloat64 & v, BString & s)`

7.60.2.18 `void toBString (BUInt64 & v, BString & s)`

7.60.2.19 `void toBString (BUInt32 & v, BString & s)`

7.60.2.20 `void toBString (BInt32 & v, BString & s)`

7.60.2.21 `void toBString (BStringList & v, BString & s)`

7.60.2.22 `void toBString (BString & v, BString & s)`

7.61 BString.h File Reference

```
#include <BTypes.h>
#include <BRefData.h>
#include <BList.h>
#include <BArray.h>
#include <iostream>
```

Classes

- class [BString](#)

Defines

- #define [BSTRING_H](#) 1

Functions

- `std::ostream & operator<<< (std::ostream &o, BString &s)`
- `std::istream & operator>>> (std::istream &i, BString &s)`
- `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)`

7.61.1 Define Documentation

7.61.1.1 `#define BSTRING_H 1`

7.61.2 Function Documentation

7.61.2.1 `void fromBString (BString & s, BFloat64 & v)`

7.61.2.2 `void fromBString (BString & s, BUInt64 & v)`

7.61.2.3 `void fromBString (BString & s, BUInt32 & v)`

7.61.2.4 `void fromBString (BString & s, BInt32 & v)`

7.61.2.5 `void fromBString (BString & s, BStringList & v)`

7.61.2.6 `void fromBString (BString & s, BString & v)`

7.61.2.7 `std::ostream& operator<< (std::ostream & o, BString & s)`

7.61.2.8 `std::istream& operator>> (std::istream & i, BString & s)`

7.61.2.9 `void toBString (BFloat64 & v, BString & s)`

7.61.2.10 `void toBString (BUInt64 & v, BString & s)`

7.61.2.11 `void toBString (BUInt32 & v, BString & s)`

7.61.2.12 `void toBString (BInt32 & v, BString & s)`

7.61.2.13 `void toBString (BStringList & v, BString & s)`

7.61.2.14 `void toBString (BString & v, BString & s)`

7.62 BStringLocked.h File Reference

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

Classes

- class [BStringMutex](#)
- class [BStringLocked](#)

Defines

- #define [BStringLocked_H 1](#)

7.62.1 Define Documentation

7.62.1.1 #define BStringLocked_H 1

7.63 BTable.cpp File Reference

```
#include <BTable.h>
```

7.64 BTable.h File Reference

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

Classes

- class [BTable](#)

7.65 BThread.cpp File Reference

```
#include <BThread.h>
#include <unistd.h>
#include <errno.h>
#include <sys/types.h>
```

7.66 BThread.h File Reference

```
#include <pthread.h>
```

Classes

- class [BThread](#)

Defines

- #define [BTHREAD_H 1](#)

7.66.1 Define Documentation

7.66.1.1 #define BTHREAD_H 1

7.67 BTimer.cpp File Reference

```
#include <BTimer.h>  
#include <sys/time.h>
```

7.68 BTimer.h File Reference

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

Classes

- class [BTimer](#)
Stopwatch style timer.

7.69 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]

7.69.1 Function Documentation

7.69.1.1 void [fromBString](#) ([BString](#) & *s*, [BTimeStamp](#) & *v*)

7.69.1.2 void [toBString](#) ([BTimeStamp](#) & *v*, [BString](#) & *s*)

7.69.2 Variable Documentation

7.69.2.1 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 }
}
```

7.70 BTimeStamp.h File Reference

```
#include <stdint.h>
```

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

Classes

- class [BTimeStamp](#)

Defines

- #define [BTimeStamp_H 1](#)

Functions

- void [toBString](#) ([BTimeStamp](#) &*v*, [BString](#) &*s*)
- void [fromBString](#) ([BString](#) &*s*, [BTimeStamp](#) &*v*)

7.70.1 Define Documentation

7.70.1.1 #define [BTimeStamp_H 1](#)

7.70.2 Function Documentation

7.70.2.1 void [fromBString](#) ([BString](#) & *s*, [BTimeStamp](#) & *v*)

7.70.2.2 void [toBString](#) ([BTimeStamp](#) & *v*, [BString](#) & *s*)

7.71 BTimeStampMs.cpp File Reference

```
#include <BTimeStampMs.h>
#include <sys/time.h>
```

Variables

- static int [mon_yday](#) [2][13]

7.71.1 Variable Documentation

7.71.1.1 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 }
}
```

7.72 BTimeStampMs.h File Reference

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

Classes

- class [BTimeStampMs](#)

Defines

- #define [BTimeStampMs_H 1](#)

7.72.1 Define Documentation

7.72.1.1 #define BTimeStampMs_H 1

7.73 BTypes.h File Reference

```
#include <stdint.h>
#include <sys/types.h>
#include <vector>
```

Defines

- #define [BTYPES_H](#) 1

Typedefs

- 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 float [BFloat](#)
- typedef double [BDouble](#)
- typedef size_t [BSize](#)
- typedef uint32_t [BUInt](#)
- typedef std::vector< float > [BArrayFloat](#)
- typedef std::vector< double > [BArrayDouble](#)
- typedef [BInt32](#) [BInt](#)

Functions

- void [byteSwap8](#) (void *d, void *s)
- void [byteSwap16](#) (void *d, void *s)
- void [byteSwap32](#) (void *d, void *s)
- void [byteSwap64](#) (void *d, void *s)

7.73.1 Define Documentation

7.73.1.1 `#define BTYPES_H 1`

7.73.2 Typedef Documentation

7.73.2.1 `typedef std::vector<double> BArrayDouble`

7.73.2.2 `typedef std::vector<float> BArrayFloat`

7.73.2.3 `typedef double BDouble`

7.73.2.4 `typedef float BFloat`

7.73.2.5 `typedef float BFloat32`

7.73.2.6 `typedef double BFloat64`

7.73.2.7 `typedef BInt32 BInt`

7.73.2.8 `typedef int16_t BInt16`

7.73.2.9 `typedef int32_t BInt32`

7.73.2.10 `typedef int64_t BInt64`

7.73.2.11 `typedef int8_t BInt8`

7.73.2.12 `typedef size_t BSize`

7.73.2.13 `typedef BUInt32 BUInt`

7.73.2.14 `typedef uint16_t BUInt16`

7.73.2.15 `typedef uint32_t BUInt32`

7.73.2.16 `typedef uint64_t BUInt64`

7.73.2.17 `typedef uint8_t BUInt8`

7.73.3 Function Documentation

7.73.3.1 `void byteSwap16 (void * d, void * s)` [inline]

7.73.3.2 `void byteSwap32 (void * d, void * s)` [inline]

7.73.3.3 `void byteSwap64 (void * d, void * s)` [inline]

7.73.3.4 `void byteSwap8 (void * d, void * s)` [inline]

7.74 BUrl.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
#include <BUrl.h>
#include <curl/curl.h>
```

7.75 BUrl.h File Reference

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

Classes

- class [BUrl](#)
Basic access to a Url.

Defines

- #define [BURL_H](#) 1

7.75.1 Define Documentation

7.75.1.1 #define BURL_H 1

Index

- ~BBuffer
 - BBuffer, 15
- ~BBufferStore
 - BBufferStore, 18
- ~BCond
 - BCond, 21
- ~BCondBool
 - BCondBool, 22
- ~BCondInt
 - BCondInt, 25
- ~BCondValue
 - BCondValue, 28
- ~BCondWrap
 - BCondWrap, 31
- ~BDebugBacktrace
 - BDebugBacktrace, 35
- ~BDir
 - BDir, 42
- ~BEntryFile
 - BEntryFile, 48
- ~BEvent
 - BEvent, 55
- ~BEventInt
 - BEventInt, 58
- ~BEventPipe
 - BEventPipe, 60
- ~BFifo
 - BFifo, 63
- ~BFile
 - BFile, 69
- ~BList
 - BList, 76
- ~BMutex
 - BMutex, 82
- ~BMutexLock
 - BMutexLock, 83
- ~BMySQL
 - BMySQL, 85
- ~BObj
 - BObj, 117
- ~BObject
 - BObject, 119
- ~BPoll
 - BPoll, 121
- ~BRWLock
 - BRWLock, 130
- ~BRefData
 - BRefData, 124
- ~BRtc
 - BRtc, 126
- ~BRtcThreaded
 - BRtcThreaded, 128
- ~BSema
 - BSema, 132
- ~BSocket
 - BSocket, 137
- ~BSocketAddress
 - BSocketAddress, 140
- ~BString
 - BString, 147
- ~BTable
 - BTable, 158
- ~BThread
 - BThread, 161
- ~BTimeStamp
 - BTimeStamp, 167
- ~BTimeStampMs
 - BTimeStampMs, 173
- ~BTimer
 - BTimer, 163
- ~BUrl
 - BUrl, 176
- ~BoapPacket
 - BoapPacket, 97
- ~BoapServer
 - BoapServer, 104
- ~BoapServerConnection
 - BoapServerConnection, 108
- ~BoapServiceObject
 - BoapServiceObject, 113
- accept
 - BSocket, 137
- add
 - BAtomicCount, 13
 - BTimer, 163
- addEntry
 - Boapns, 10
 - Boapns::Boapns, 95
- addMember

- BObject, 119
- addMicroSeconds
 - BTimeStamp, 168
- addMilliSeconds
 - BTimeStamp, 168
 - BTimeStampMs, 173
- addObject
 - BoapServer, 104, 106
- addRef
 - BRefData, 124
- address
 - BSocketAddressINET, 142
- addressList
 - Boapns::BoapEntry, 93
- addRow
 - BTable, 158
- addSeconds
 - BTimeStamp, 169
 - BTimeStampMs, 173
- apiVersion
 - Boapns, 10
- APIVERSION_TEST
 - Boap.cpp, 216
- append
 - BArray, 11
 - BList, 77, 78
 - BPoll, 121
 - BString, 149
- average
 - BTimer, 163
- BArray, 11
 - append, 11
 - BArray, 11
- BArray.h, 179
 - BArray_H, 179
- BArray_H
 - BArray.h, 179
- BArrayDouble
 - BTypes.h, 259
- BArrayFloat
 - BTypes.h, 259
- barrayToString
 - BString.cpp, 244
- basename
 - BString, 151
- BAtomicCount, 13
 - add, 13
 - BAtomicCount, 13
 - getValue, 13
 - operator long, 13
 - operator++, 13
 - operator--, 13
 - ovalue, 13
- BAtomicCount.h, 180
 - BAtomicCount_H, 180
- BAtomicCount_H
 - BAtomicCount.h, 180
- BBuffer, 15
 - ~BBuffer, 15
 - BBuffer, 15
 - data, 16
 - odata, 16
 - odataSize, 16
 - osize, 16
 - resize, 16
 - setData, 16
 - setSize, 16
 - size, 16
 - writeData, 16
- BBuffer.cpp, 181
 - roundSize, 181
- BBuffer.h, 182
 - BBUFFER_H, 182
- BBUFFER_H
 - BBuffer.h, 182
- BBufferStore, 17
 - ~BBufferStore, 18
 - BBufferStore, 18
 - copyWithSwap, 20
 - getHexString, 18
 - getPos, 18
 - opos, 20
 - pop, 19, 20
 - push, 18, 19
 - setHexString, 18
 - setPos, 18
- BCond, 21
 - ~BCond, 21
 - BCond, 21
 - ocond, 21
 - omutex, 21
 - signal, 21
 - timedWait, 21
 - wait, 21
- BCond.cpp, 183
- BCond.h, 184
 - BCOND_H, 184
- BCOND_H
 - BCond.h, 184
- BCondBool, 22
 - ~BCondBool, 22
 - BCondBool, 22
 - clear, 22
 - ocond, 23
 - omutex, 23
 - operator int, 23
 - ovalue, 23

- set, 22
- timedWait, 23
- value, 23
- wait, 23
- BCondInt, 24
 - ~BCondInt, 25
 - BCondInt, 25
 - decrement, 25
 - increment, 25
 - ocond, 26
 - omutex, 26
 - operator++, 26
 - operator-, 26
 - ovalue, 26
 - setValue, 25
 - timedWait, 25
 - tryNotZeroDecrement, 25
 - value, 25
 - wait, 25
 - waitIncrement, 25
 - waitNotZero, 25
 - waitNotZeroDecrement, 25
- BCondInt.cpp, 185
- BCondInt.h, 186
 - BCONDINT_H, 186
- BCONDINT_H
 - BCondInt.h, 186
- BCondValue, 27
 - ~BCondValue, 28
 - BCondValue, 28
 - decrement, 28
 - increment, 28
 - ocond, 29
 - omutex, 29
 - operator++, 28
 - operator+=", 28
 - operator-, 29
 - operator-=, 28
 - ovalue, 29
 - setValue, 28
 - value, 28
 - waitLessThan, 28
 - waitLessThanOrEqual, 28
 - waitMoreThanOrEqual, 28
- BCondWrap, 30
 - ~BCondWrap, 31
 - BCondWrap, 31
 - decrement, 31
 - diff, 32
 - increment, 31
 - ocond, 32
 - omutex, 32
 - operator++, 32
 - operator+=", 31
 - operator-, 32
 - operator=, 31
 - ovalue, 32
 - setValue, 31
 - value, 31
 - waitLessThan, 31
 - waitLessThanOrEqual, 31
 - waitMoreThanOrEqual, 31
- BConfig, 33
 - findValue, 33
 - ofile, 33
 - oclock, 33
 - open, 33
 - read, 33
 - write, 33
- BConfig.cpp, 187
- BConfig.h, 188
- bdebug
 - BDebug.cpp, 190
 - BDebug.h, 192
- BDebug.cpp, 189
 - bdebug, 190
 - BTRACE_SIZE, 190
 - gettid, 190
 - getTime, 190
 - hd32, 190
 - hd8, 190
 - hda32, 190
 - hda8, 190
 - setDebug, 190
 - STRBUF_SIZE, 190
 - tprintf, 190
- BDebug.h, 191
 - bdebug, 192
 - BDebug_STD, 191
 - dprintf, 191
 - eprintf, 191
 - gettid, 192
 - getTime, 192
 - hd32, 192
 - hd8, 192
 - hda8, 192
 - hds32, 192
 - nprintf, 192
 - setDebug, 192
 - tprintf, 192
 - wprintf, 192
- BDebug_STD
 - BDebug.h, 191
- BDebugBacktrace, 35
 - ~BDebugBacktrace, 35
 - BDebugBacktrace, 35
 - dumpBacktrace, 35
 - dumpBacktraceFile, 35

- dumpBacktraceStdout, 35
- dumpBacktraceSyslog, 35
- BDict, 36
 - del, 36
 - find, 36
 - hasKey, 36
 - iterator, 36
 - key, 36
 - operator+, 37
 - operator=, 37
- BDict.cpp, 193
 - fromBString, 193
 - toBString, 193
- BDict.h, 194
 - BDict_H, 194
 - BDictString, 194
 - fromBString, 194
 - toBString, 194
- BDict_H
 - BDict.h, 194
- BDictItem, 38
 - BDictItem, 38
 - key, 38
 - value, 38
- BDictMap, 39
 - clear, 40
 - del, 40
 - hasKey, 40
 - isEnd, 40
 - iterator, 40
 - key, 40
 - next, 40
 - size, 40
 - start, 40
- BDictMap.h, 195
 - BDictMap_H, 195
 - BDictMapString, 195
- BDictMap_H
 - BDictMap.h, 195
- BDictMapString
 - BDictMap.h, 195
- BDictString
 - BDict.h, 194
- BDir, 41
 - ~BDir, 42
 - BDir, 42
 - clear, 42
 - entryName, 42
 - entryStat, 43
 - entryStat64, 43
 - error, 42
 - odirname, 43
 - oerror, 43
 - open, 42
 - osort, 43
 - owild, 43
 - read, 42
 - setSort, 42
 - setWild, 42
- BDir.cpp, 196
 - wild, 196
 - wildString, 196
- BDir.h, 197
 - BDir_H, 197
- BDir_H
 - BDir.h, 197
- BDouble
 - BTypes.h, 259
- begin
 - BList, 76
- BEntry, 44
 - BEntry, 45
 - getName, 45
 - getValue, 45
 - line, 45
 - oname, 46
 - ovalue, 46
 - print, 45
 - setLine, 45
 - setName, 45
 - setValue, 45
- BEntry.cpp, 198
- BEntry.h, 199
- BEntryFile, 47
 - ~BEntryFile, 48
 - BEntryFile, 48
 - clear, 48
 - ocomments, 48
 - ofilename, 48
 - open, 48
 - read, 48
 - write, 48
 - writeList, 48
- BEntryList, 49
 - BEntryList, 50
 - clear, 51
 - del, 51
 - deleteEntry, 50
 - find, 50
 - findValue, 50
 - getString, 50
 - insert, 51
 - isSet, 50
 - olastPos, 51
 - operator=, 51
 - print, 50
 - setValue, 50
 - setValueRaw, 50

- BError, 52
 - BError, 53
 - clear, 53
 - copy, 53
 - ERROR, 53
 - getErrorNo, 54
 - getString, 53
 - NONE, 53
 - oerrNo, 54
 - oerrStr, 54
 - operator int, 54
 - set, 53
 - setError, 53
 - Type, 53
- BError.cpp, 200
- BError.h, 201
 - BERROR_H, 201
- BERROR_H
 - BError.h, 201
- BEvent, 55
 - ~BEvent, 55
 - BEvent, 55
 - getBinary, 55
 - getType, 55
 - otype, 56
 - setBinary, 55
- BEvent.cpp, 202
- BEvent.h, 203
 - BEvent_H, 203
 - BEventType, 203
 - BEventTypeError, 203
 - BEventTypeInt, 203
 - BEventTypeNone, 203
- BEvent_H
 - BEvent.h, 203
- BEventError, 57
 - BEventError, 57
 - getBinary, 57
 - setBinary, 57
- BEventInt, 58
 - ~BEventInt, 58
 - BEventInt, 58
 - clear, 58
 - getEvent, 58
 - getFd, 59
 - ofds, 59
 - sendEvent, 58
- BEventPipe, 60
 - ~BEventPipe, 60
 - BEventPipe, 60
 - clear, 60
 - getEvent, 61
 - getReceiveFd, 61
 - ofds, 61
 - sendEvent, 60
- BEventType
 - BEvent.h, 203
- BEventTypeError
 - BEvent.h, 203
- BEventTypeInt
 - BEvent.h, 203
- BEventTypeNone
 - BEvent.h, 203
- BFifo, 62
 - ~BFifo, 63
 - BFifo, 63
 - clear, 63
 - defaultSize, 63
 - mapCircularBuffer, 64
 - odata, 65
 - oclock, 65
 - oreadPos, 65
 - osize, 65
 - ovmSize, 65
 - owriteNumFifoSamples, 65
 - owritePos, 65
 - read, 64
 - readAvailable, 64
 - readData, 64
 - readDone, 64
 - readWaitAvailable, 64
 - size, 63
 - unmapCircularBuffer, 65
 - write, 64
 - writeAvailable, 63
 - writeData, 64
 - writeDone, 64
 - writeWaitAvailable, 64
- BFifo.cpp, 204
 - DEBUG, 204
 - dprintf, 204
- BFifo.h, 205
 - BFIFO_H, 205
- BFifo.inc, 206
- BFIFO_H
 - BFifo.h, 205
- BFifoPos, 66
 - BFifoPos, 66
 - difference, 67
 - increment, 67
 - operator int, 67
 - operator!=, 67
 - operator+==, 67
 - operator==, 67
 - opos, 67
 - osize, 67
 - pos, 66
 - set, 66

- setSize, 66
- BFile, 68
 - ~BFile, 69
 - BFile, 69
 - close, 70
 - fgets, 70
 - flush, 71
 - getFd, 70
 - isEnd, 70
 - length, 70
 - ofile, 71
 - ofileName, 71
 - omode, 71
 - open, 69
 - operator=, 71
 - position, 70
 - printf, 71
 - read, 70
 - readString, 70
 - seek, 70
 - setVBuf, 70
 - truncate, 71
 - write, 70
 - writeString, 70
- BFile.cpp, 207
- STRBUF, 207
- BFile.h, 208
 - BFILE_H, 208
- BFILE_H
 - BFile.h, 208
- BFloat
 - BTypes.h, 259
- BFloat32
 - BTypes.h, 259
- BFloat64
 - BTypes.h, 259
- bind
 - BSocket, 137
- BInt
 - BTypes.h, 259
- BInt16
 - BTypes.h, 259
- BInt32
 - BTypes.h, 259
- BInt64
 - BTypes.h, 259
- BInt8
 - BTypes.h, 259
- BIter, 72
 - BIter, 72
 - oi, 72
 - operator BNode *, 72
 - operator==, 72
- BList, 73
 - ~BList, 76
 - append, 77, 78
 - begin, 76
 - BList, 76
 - clear, 77
 - del, 78
 - deleteFirst, 78
 - deleteLast, 78
 - end, 76
 - front, 77
 - get, 77
 - goTo, 76
 - insert, 77
 - insertAfter, 77
 - isEnd, 77
 - next, 76
 - nodeCreate, 79
 - nodeGet, 79
 - number, 76
 - olength, 79
 - onodes, 79
 - operator+, 79
 - operator=, 79
 - pop, 78
 - position, 76
 - prev, 76
 - push, 78
 - queueAdd, 78
 - queueGet, 78
 - rear, 77
 - size, 77
 - sort, 78
 - SortFunc, 76
 - start, 76
 - swap, 78
- BList.h, 209
 - BLIST_H, 209
 - BListLoop, 209
- BList::Node, 80
 - item, 80
 - Node, 80
- BList_func.h, 210
- BLIST_H
 - BList.h, 209
- BListLoop
 - BList.h, 209
- blistToString
 - BString.cpp, 244
- BMember
 - BObject.h, 229
- BMemberList
 - BObject.h, 229
- BMutex, 81
 - ~BMutex, 82

- BMutex, 82
- lock, 82
- Normal, 82
- omutex, 82
- operator=, 82
- Recursive, 82
- timedLock, 82
- tryLock, 82
- Type, 82
- unlock, 82
- BMutex.cpp, 211
- MDEBUG, 211
- BMutex.h, 212
- BMUTEX_H, 212
- BMUTEX_H
- BMutex.h, 212
- BMutexLock, 83
- ~BMutexLock, 83
- BMutexLock, 83
- lock, 83
- BMysql, 84
- ~Bmysql, 85
- Bmysql, 85
- db, 85
- get, 85
- insert, 85
- odb, 85
- odebug, 85
- lock, 85
- opened, 85
- open, 85
- query, 85
- setDebug, 85
- update, 85
- BMysql.cpp, 213
- BMysql.h, 214
- Bmysql_H, 214
- BMysql_H
- BMysql.h, 214
- BNameValue, 86
- BNameValue, 86
- getName, 86
- getValue, 86
- oname, 86
- ovalue, 86
- BNameValue.h, 215
- BNAMEVALUE_H, 215
- BNAMEVALUE_H
- BNameValue.h, 215
- BNameValueList, 87
- find, 87
- findPos, 87
- BNode, 88
- BNode, 88
- next, 88
- prev, 88
- Boap.cpp, 216
- APIVERSION_TEST, 216
- boapPort, 216
- DEBUG, 216
- dprintf, 216
- IS_BIG_ENDIAN, 216
- Boap.h, 217
- BoapFunc, 218
- BoapMagic, 218
- BoapPriority, 218
- BoapPriorityHigh, 218
- BoapPriorityLow, 218
- BoapPriorityNormal, 218
- BoapService, 218
- BoapType, 218
- BoapTypeRpc, 218
- BoapTypeRpcError, 218
- BoapTypeRpcReply, 218
- BoapTypeSignal, 218
- BoapClientObject, 89
- BoapClientObject, 90
- checkApiVersion, 91
- connectService, 90, 91
- disconnectService, 90
- getServiceName, 90
- oapiVersion, 92
- oconnected, 92
- lock, 92
- omaxLength, 92
- oname, 92
- opriority, 92
- oreconnect, 92
- orx, 92
- oservice, 92
- otimeout, 92
- otx, 92
- performCall, 91, 92
- performRecv, 91, 92
- performSend, 91, 92
- ping, 90
- pingLocked, 91
- setConnectionPriority, 90
- setMaxLength, 91
- setTimeout, 91
- BoapEntry
- Boapns::BoapEntry, 93
- BoapFunc
- Boap.h, 218
- BoapSimple.h, 225
- BoapFuncEntry, 94
- BoapFuncEntry, 94
- ocmd, 94

- ofunc, 94
- BoapMagic
 - Boap.h, 218
- Boapns, 9
 - addEntry, 10
 - apiVersion, 10
 - Boapns, 10
 - Boapns::Boapns, 95
 - delEntry, 10
 - getEntry, 10
 - getEntryList, 10
 - getNewName, 10
 - getVersion, 10
- Boapns::BoapEntry, 93
 - addressList, 93
 - BoapEntry, 93
 - hostName, 93
 - name, 93
 - port, 93
 - service, 93
- Boapns::Boapns, 95
 - addEntry, 95
 - Boapns, 95
 - delEntry, 95
 - getEntry, 95
 - getEntryList, 95
 - getNewName, 95
 - getVersion, 95
- BoapnsC.cc, 219
- BoapnsC.h, 220
 - BOAPNSC_H, 220
- BOAPNSC_H
 - BoapnsC.h, 220
- BoapnsD.cc, 221
- BoapnsD.h, 222
 - BOAPNSD_H, 222
- BOAPNSD_H
 - BoapnsD.h, 222
- BoapPacket, 96
 - ~BoapPacket, 97
 - BoapPacket, 97
 - data, 97
 - getCmd, 97
 - nbytes, 97
 - odata, 99
 - onbytes, 99
 - opos, 99
 - osize, 99
 - peekHead, 97
 - pop, 98, 99
 - popHead, 97, 98
 - push, 98
 - pushHead, 97
 - resize, 97
 - setData, 97
 - updateHead, 97
 - updateLen, 99
- BoapPacketHead, 101
 - cmd, 101
 - length, 101
 - reserved, 101
 - service, 101
 - type, 101
- boapPort
 - Boap.cpp, 216
- BoapPriority
 - Boap.h, 218
- BoapPriorityHigh
 - Boap.h, 218
- BoapPriorityLow
 - Boap.h, 218
- BoapPriorityNormal
 - Boap.h, 218
- BoapServer, 102
 - ~BoapServer, 104
 - addObject, 104, 106
 - BoapServer, 104
 - clientGone, 104
 - function, 104
 - getConnectionsNumber, 104
 - getEventSocket, 104, 106
 - getHostName, 104, 106
 - getSocket, 104, 106
 - init, 104
 - newConnection, 104
 - NOTTHREADS, 103
 - oboapNs, 106
 - oboapns, 106
 - oclientGoneEvent, 106
 - oclients, 106
 - ohostName, 106
 - oisBoapns, 106
 - onet, 106
 - onetEvent, 106
 - onetEventAddress, 106
 - onumOperations, 106
 - opoll, 106
 - orx, 106
 - oservices, 106
 - othreaded, 106
 - otx, 106
 - process, 104, 106
 - processEvent, 104, 106
 - run, 104, 106
 - sendEvent, 104, 106
 - THREADED, 103
- BoapServerConnection, 108
 - ~BoapServerConnection, 108

- BoapServerConnection, 108
 - function, 109
 - getHead, 109
 - getSocket, 109
 - init, 108
 - oboapServer, 109
 - omaxLength, 109
 - orx, 109
 - osocket, 109
 - otx, 109
 - process, 108
 - setMaxLength, 109
 - validate, 109
- BoapService
 - Boap.h, 218
 - BoapSimple.h, 225
- BoapServiceEntry, 110
 - BoapServiceEntry, 110
 - oobject, 110
 - oservice, 110
- BoapServiceObject, 111
 - ~BoapServiceObject, 113
 - BoapServiceObject, 113
 - doConnectionPriority, 113
 - doPing, 113
 - name, 113
 - oapiVersion, 113
 - ofuncList, 113
 - oname, 113
 - oserver, 113
 - process, 113
 - processEvent, 113
 - sendEvent, 113
 - setName, 113
- BoapSignalObject, 115
 - BoapSignalObject, 115
 - orx, 115
 - otx, 115
 - performSend, 115
- BoapSimple.cc, 223
 - DEBUG, 223
 - dprintf, 223
 - roundSize, 223
- BoapSimple.h, 224
 - BoapFunc, 225
 - BoapService, 225
 - BoapType, 225
 - BoapTypeRpc, 225
 - BoapTypeRpcError, 225
 - BoapTypeRpcReply, 225
 - BoapTypeSignal, 225
 - Double, 225
 - Int16, 225
 - Int32, 225
 - Int8, 225
 - UInt16, 225
 - UInt32, 225
 - UInt8, 225
- BoapType
 - Boap.h, 218
 - BoapSimple.h, 225
- BoapTypeRpc
 - Boap.h, 218
 - BoapSimple.h, 225
- BoapTypeRpcError
 - Boap.h, 218
 - BoapSimple.h, 225
- BoapTypeRpcReply
 - Boap.h, 218
 - BoapSimple.h, 225
- BoapTypeSignal
 - Boap.h, 218
 - BoapSimple.h, 225
- BObj, 117
 - ~BObj, 117
 - BObj, 117
 - getDebugString, 117
 - getMembers, 117
 - getType, 117
 - setMembers, 117
- BObj.cpp, 226
- BObj.h, 227
 - BObj_H, 227
- BObj_H
 - BObj.h, 227
- BObject, 118
 - ~BObject, 119
 - addMember, 119
 - BObject, 119
 - createObj, 119
 - getBinary, 119
 - getMemberList, 119
 - getString, 119
 - getType, 119
 - otype, 119
 - printIt, 119
 - setBinary, 119
 - setString, 119
- BObject.cc, 228
 - DEBUG, 228
- BObject.h, 229
 - BMember, 229
 - BMemberList, 229
 - BOBJECT_H, 229
- BOBJECT_H
 - BObject.h, 229
- BPoll, 120
 - ~BPoll, 121

- append, 121
- BPoll, 121
- clear, 121
- delFd, 121
- doPoll, 121
- getPollFds, 121
- getPollFdsNum, 121
- nextFd, 121
- ofds, 121
- ofdsNext, 121
- ofdsNum, 121
- PollFd, 121
- BPoll.cpp, 230
- BPoll.h, 231
 - B POLL_H, 231
- B POLL_H
 - BPoll.h, 231
- BRefData, 123
 - ~BRefData, 124
 - addRef, 124
 - BRefData, 124
 - copy, 124
 - data, 124
 - deleteRef, 124
 - len, 124
 - odata, 124
 - olen, 124
 - operator=, 124
 - orefCount, 124
 - setLen, 124
- BRefData.cpp, 232
 - CHUNK, 232
 - DEBUG, 232
- BRefData.h, 233
 - BREFDATA_H, 233
- BREFDATA_H
 - BRefData.h, 233
- BRtc, 126
 - ~BRtc, 126
 - BRtc, 126
 - init, 126
 - ofd, 126
 - orate, 126
 - wait, 126
- BRtc.cpp, 234
- BRtc.h, 235
- BRtcThreaded, 128
 - ~BRtcThreaded, 128
 - BRtcThreaded, 128
 - function, 129
 - init, 128
 - ocond, 129
 - orate, 129
 - ortc, 129
 - wait, 128
- BRWLock, 130
 - ~BRWLock, 130
 - BRWLock, 130
 - olock, 131
 - operator=, 131
 - rdLock, 130
 - tryRdLock, 130
 - tryWrLock, 131
 - unlock, 131
 - wrLock, 131
- BRWLock.cpp, 236
- BRWLock.h, 237
 - BRWLOCK_H, 237
- BRWLOCK_H
 - BRWLock.h, 237
- BSema, 132
 - ~BSema, 132
 - BSema, 132
 - getValue, 133
 - operator=, 133
 - osema, 133
 - post, 132
 - timedWait, 133
 - tryWait, 133
 - wait, 132
- BSema.cpp, 238
- BSema.h, 239
 - BSEMA_H, 239
- BSEMA_H
 - BSema.h, 239
- BSize
 - BTypes.h, 259
- BSocket, 134
 - ~BSocket, 137
 - accept, 137
 - bind, 137
 - BSocket, 137
 - close, 137
 - connect, 137
 - DGRAM, 135
 - getAddress, 137
 - getFd, 137
 - getMTU, 137
 - getSockOpt, 137
 - init, 137
 - listen, 137
 - NType, 135
 - osocket, 137
 - Priority, 135
 - PriorityHigh, 135
 - PriorityLow, 135
 - PriorityNormal, 135
 - recv, 137

- recvFrom, 137
- recvFromWithTimeout, 137
- recvWithTimeout, 137
- send, 137
- sendTo, 137
- setBroadCast, 137
- setPriority, 137
- setReuseAddress, 137
- setSockOpt, 137
- shutdown, 137
- STREAM, 135
- BSocket.cpp, 240
 - IP_MTU, 240
- BSocket.h, 241
 - BSOCKET_H, 241
- BSOCKET_H
 - BSocket.h, 241
- BSocketAddress, 139
 - ~BSocketAddress, 140
 - BSocketAddress, 140
 - len, 140
 - oaddress, 140
 - olen, 140
 - operator const SockAddr *, 140
 - operator!=, 140
 - operator=, 140
 - operator==, 140
 - raw, 140
 - set, 140
 - SockAddr, 140
- BSocketAddressINET, 141
 - address, 142
 - getHostName, 142
 - getIpAddresses, 142
 - getIpAddressList, 142
 - getIpAddressListAll, 142
 - getString, 142
 - port, 142
 - set, 142
 - setPort, 142
 - SockAddrIP, 142
- BString, 144
 - ~BString, 147
 - append, 149
 - basename, 151
 - BString, 147
 - compare, 149
 - compareWild, 149
 - compareWildExpression, 149
 - convert, 147, 148
 - convertHex, 148
 - copy, 148
 - del, 150
 - dirname, 151
 - extension, 151
 - field, 153
 - fields, 153
 - find, 150
 - findReverse, 150
 - getTokenList, 150
 - Init, 153
 - insert, 150
 - inString, 153
 - isSpace, 153
 - justify, 150
 - len, 148
 - operator const char *, 153
 - operator!=, 153
 - operator<, 153
 - operator<=, 153
 - operator>, 153
 - operator>=, 153
 - operator+, 153
 - operator+=, 153
 - operator=, 151
 - operator==, 151, 153
 - ostr, 153
 - pad, 149
 - printf, 150
 - pullLine, 151
 - pullSeparators, 151
 - pullToken, 150
 - pullWord, 151
 - removeNL, 149
 - removeSeparators, 150
 - retDouble, 149
 - retInt, 148
 - retStr, 148
 - retStrDup, 148
 - retUInt, 149
 - subString, 150
 - toLower, 149
 - toUpper, 149
 - truncate, 149
- BString.cpp, 242
 - barrayToString, 244
 - blistToString, 244
 - bstringListinList, 244
 - bstringToArray, 244
 - bstringToList, 244
 - charToArray, 244
 - charToList, 244
 - DEBUG, 244
 - fromBString, 244
 - gmatch, 244
 - MINUS, 244
 - operator<<, 244
 - operator>>, 244

- STRIP, 244
- toBString, 244
- BString.h, 245
- BSTRING_H, 246
- fromBString, 246
- operator<<, 246
- operator>>, 246
- toBString, 246
- BSTRING_H
- BString.h, 246
- bstringListinList
- BString.cpp, 244
- BStringLocked, 155
- BStringLocked, 155
- len, 155
- lock, 155
- operator BString, 155
- operator+, 155
- operator=, 155
- ostr, 155
- BStringLocked.h, 247
- BStringLocked_H, 247
- BStringLocked_H
- BStringLocked.h, 247
- BStringMutex, 157
- BStringMutex, 157
- bstringToArray
- BString.cpp, 244
- bstringToList
- BString.cpp, 244
- BTable, 158
- ~BTable, 158
- addRow, 158
- BTable, 158
- calculateWidths, 158
- ocolumnWidths, 158
- odata, 158
- otitle, 158
- print, 158
- printLine, 158
- setTitle, 158
- BTable.cpp, 248
- BTable.h, 249
- BThread, 160
- ~BThread, 161
- BThread, 161
- cancel, 161
- function, 161
- getThread, 161
- opolicy, 161
- opriority, 161
- oreult, 161
- orunning, 161
- ostackSize, 161
- othread, 161
- result, 161
- running, 161
- setInitPriority, 161
- setInitStackSize, 161
- setPriority, 161
- start, 161
- startFunc, 161
- waitForCompletion, 161
- BThread.cpp, 250
- BThread.h, 251
- BTHREAD_H, 251
- BTHREAD_H
- BThread.h, 251
- BTimer, 162
- ~BTimer, 163
- add, 163
- average, 163
- BTimer, 163
- clear, 163
- getElapsedTime, 163
- getTime, 163
- oaverage, 164
- oendTime, 164
- lock, 164
- onum, 164
- opeak, 164
- ostartTime, 164
- peak, 163
- start, 163
- stop, 163
- BTimer.cpp, 252
- BTimer.h, 253
- BTimeStamp, 165
- ~BTimeStamp, 167
- addMicroSeconds, 168
- addMilliSeconds, 168
- addSeconds, 169
- BTimeStamp, 167
- clear, 167
- compare, 169
- day, 168
- difference, 169
- getDate, 168
- getString, 168
- getStringFormatted, 168
- getStringNoMs, 168
- getYearMicroSeconds, 169
- getYearSeconds, 169
- hour, 168
- isLeap, 169
- isSet, 169
- microSecond, 168
- minute, 168

- month, 168
- ohour, 170
- omicroSecond, 170
- omminute, 170
- operator BString, 169
- operator!=, 169
- operator<, 169
- operator<=, 169
- operator>, 169
- operator>=, 169
- operator=, 169
- operator==, 169
- osecond, 170
- ospare, 170
- oyday, 169
- oyear, 169
- second, 168
- set, 167
- setFirst, 167
- setLast, 167
- setNow, 168
- setString, 168
- setTime, 168
- setYDay, 167
- yday, 168
- year, 168
- BTimeStamp.cpp, 254
 - fromBString, 254
 - mon_yday, 254
 - toBString, 254
- BTimeStamp.h, 255
 - BTimeStamp_H, 255
 - fromBString, 255
 - toBString, 255
- BTimeStamp_H
 - BTimeStamp.h, 255
- BTimeStampMs, 171
 - ~BTimeStampMs, 173
 - addMilliseconds, 173
 - addSeconds, 173
 - BTimeStampMs, 173
 - clear, 173
 - compare, 174
 - difference, 174
 - getDate, 174
 - getDurationString, 174
 - getDurationStringNoMs, 174
 - getString, 173
 - getStringNoMs, 173
 - getStringRaw, 174
 - getYearMilliseconds, 173
 - getYearSeconds, 173
 - hour, 174
 - isLeap, 174
 - milliSecond, 175
 - minute, 175
 - operator<, 174
 - operator<=, 174
 - operator>, 174
 - operator>=, 174
 - sampleNumber, 175
 - second, 175
 - setDurationString, 174
 - setNow, 173
 - setString, 173
 - subMilliseconds, 173
 - subSeconds, 173
 - yday, 174
 - year, 174
- BTimeStampMs.cpp, 256
 - mon_yday, 256
- BTimeStampMs.h, 257
 - BTimeStampMs_H, 257
- BTimeStampMs_H
 - BTimeStampMs.h, 257
- BTRACE_SIZE
 - BDebug.cpp, 190
- BTypes.h, 258
 - BArrayDouble, 259
 - BArrayFloat, 259
 - BDouble, 259
 - BFloat, 259
 - BFloat32, 259
 - BFloat64, 259
 - BInt, 259
 - BInt16, 259
 - BInt32, 259
 - BInt64, 259
 - BInt8, 259
 - BSize, 259
 - BTYPES_H, 259
 - BUInt, 259
 - BUInt16, 259
 - BUInt32, 259
 - BUInt64, 259
 - BUInt8, 259
 - byteSwap16, 259
 - byteSwap32, 259
 - byteSwap64, 259
 - byteSwap8, 259
- BTYPES_H
 - BTypes.h, 259
- BUInt
 - BTypes.h, 259
- BUInt16
 - BTypes.h, 259
- BUInt32
 - BTypes.h, 259
- BUInt64
 - BTypes.h, 259
- BUInt8
 - BTypes.h, 259

- BUInt64
 - BTypes.h, 259
- BUInt8
 - BTypes.h, 259
- BUrl, 176
 - ~BUrl, 176
 - BUrl, 176
 - oinit, 177
 - ores, 177
 - readString, 176
 - writeData, 176
- BUrl.cpp, 260
- BUrl.h, 261
 - BURL_H, 261
- BURL_H
 - BUrl.h, 261
- byteSwap16
 - BTypes.h, 259
- byteSwap32
 - BTypes.h, 259
- byteSwap64
 - BTypes.h, 259
- byteSwap8
 - BTypes.h, 259
- calculateWidths
 - BTable, 158
- cancel
 - BThread, 161
- charToArray
 - BString.cpp, 244
- charToList
 - BString.cpp, 244
- checkApiVersion
 - BoapClientObject, 91
- CHUNK
 - BRefData.cpp, 232
- clear
 - BCondBool, 22
 - BDictMap, 40
 - BDir, 42
 - BEntryFile, 48
 - BEntryList, 51
 - BError, 53
 - BEventInt, 58
 - BEventPipe, 60
 - BFifo, 63
 - BList, 77
 - BPoll, 121
 - BTimer, 163
 - BTimeStamp, 167
 - BTimeStampMs, 173
- clientGone
 - BoapServer, 104
- close
 - BFile, 70
 - BSocket, 137
- cmd
 - BoapPacketHead, 101
- compare
 - BString, 149
 - BTimeStamp, 169
 - BTimeStampMs, 174
- compareWild
 - BString, 149
- compareWildExpression
 - BString, 149
- connect
 - BSocket, 137
- connectService
 - BoapClientObject, 90, 91
- convert
 - BString, 147, 148
- convertHex
 - BString, 148
- copy
 - BError, 53
 - BRefData, 124
 - BString, 148
- copyWithSwap
 - BBufferStore, 20
- createObj
 - BObject, 119
- data
 - BBuffer, 16
 - BoapPacket, 97
 - BRefData, 124
- day
 - BTimeStamp, 168
- db
 - BMysql, 85
- DEBUG
 - BFifo.cpp, 204
 - Boap.cpp, 216
 - BoapSimple.cc, 223
 - BObject.cc, 228
 - BRefData.cpp, 232
 - BString.cpp, 244
- decrement
 - BCondInt, 25
 - BCondValue, 28
 - BCondWrap, 31
- defaultSize
 - BFifo, 63
- del
 - BDict, 36
 - BDictMap, 40

- BEntryList, 51
- BList, 78
- BString, 150
- delEntry
 - Boapns, 10
 - Boapns::Boapns, 95
- deleteEntry
 - BEntryList, 50
- deleteFirst
 - BList, 78
- deleteLast
 - BList, 78
- deleteRef
 - BRefData, 124
- delFd
 - BPoll, 121
- DGRAM
 - BSocket, 135
- diff
 - BCondWrap, 32
- difference
 - BFifoPos, 67
 - BTimeStamp, 169
 - BTimeStampMs, 174
- dirname
 - BString, 151
- disconnectService
 - BoapClientObject, 90
- doConnectionPriority
 - BoapServiceObject, 113
- doPing
 - BoapServiceObject, 113
- doPoll
 - BPoll, 121
- Double
 - BoapSimple.h, 225
- dprintf
 - BDebug.h, 191
 - BFifo.cpp, 204
 - Boap.cpp, 216
 - BoapSimple.cc, 223
- dumpBacktrace
 - BDebugBacktrace, 35
- dumpBacktraceFile
 - BDebugBacktrace, 35
- dumpBacktraceStdout
 - BDebugBacktrace, 35
- dumpBacktraceSyslog
 - BDebugBacktrace, 35
- end
 - BList, 76
- entryName
 - BDir, 42
- entryStat
 - BDir, 43
- entryStat64
 - BDir, 43
- eprintf
 - BDebug.h, 191
- ERROR
 - BError, 53
- error
 - BDir, 42
- extension
 - BString, 151
- fgets
 - BFile, 70
- field
 - BString, 153
- fields
 - BString, 153
- find
 - BDict, 36
 - BEntryList, 50
 - BNameValueList, 87
 - BString, 150
- findPos
 - BNameValueList, 87
- findReverse
 - BString, 150
- findValue
 - BConfig, 33
 - BEntryList, 50
- flush
 - BFile, 71
- fromBString
 - BDict.cpp, 193
 - BDict.h, 194
 - BString.cpp, 244
 - BString.h, 246
 - BTimeStamp.cpp, 254
 - BTimeStamp.h, 255
- front
 - BList, 77
- function
 - BoapServer, 104
 - BoapServerConnection, 109
 - BRtcThreaded, 129
 - BThread, 161
- get
 - BList, 77
 - BMysql, 85
- getAddress
 - BSocket, 137
- getBinary

- BEvent, 55
- BEventError, 57
- BObject, 119
- getCmd
 - BoapPacket, 97
- getConnectionsNumber
 - BoapServer, 104
- getDate
 - BTimeStamp, 168
 - BTimeStampMs, 174
- getDebugString
 - BObj, 117
- getDurationString
 - BTimeStampMs, 174
- getDurationStringNoMs
 - BTimeStampMs, 174
- getElapsedTime
 - BTimer, 163
- getEntry
 - Boapns, 10
 - Boapns::Boapns, 95
- getEntryList
 - Boapns, 10
 - Boapns::Boapns, 95
- getErrorNo
 - BError, 54
- getEvent
 - BEventInt, 58
 - BEventPipe, 61
- getEventSocket
 - BoapServer, 104, 106
- getFd
 - BEventInt, 59
 - BFile, 70
 - BSocket, 137
- getHead
 - BoapServerConnection, 109
- getHexString
 - BBufferStore, 18
- getHostName
 - BoapServer, 104, 106
 - BSocketAddressINET, 142
- getIpAddresses
 - BSocketAddressINET, 142
- getIpAddressList
 - BSocketAddressINET, 142
- getIpAddressListAll
 - BSocketAddressINET, 142
- getMemberList
 - BObject, 119
- getMembers
 - BObj, 117
- getMTU
 - BSocket, 137
- getName
 - BEntry, 45
 - BNameValue, 86
- getNewName
 - Boapns, 10
 - Boapns::Boapns, 95
- getPollFds
 - BPoll, 121
- getPollFdsNum
 - BPoll, 121
- getPos
 - BBufferStore, 18
- getReceiveFd
 - BEventPipe, 61
- getServiceName
 - BoapClientObject, 90
- getSocket
 - BoapServer, 104, 106
 - BoapServerConnection, 109
- getSockOpt
 - BSocket, 137
- getString
 - BEntryList, 50
 - BError, 53
 - BObject, 119
 - BSocketAddressINET, 142
 - BTimeStamp, 168
 - BTimeStampMs, 173
- getStringFormatted
 - BTimeStamp, 168
- getStringNoMs
 - BTimeStamp, 168
 - BTimeStampMs, 173
- getStringRaw
 - BTimeStampMs, 174
- getThread
 - BThread, 161
- gettid
 - BDebug.cpp, 190
 - BDebug.h, 192
- getTime
 - BDebug.cpp, 190
 - BDebug.h, 192
 - BTimer, 163
- getTokenList
 - BString, 150
- getType
 - BEvent, 55
 - BObj, 117
 - BObject, 119
- getValue
 - BAtomicCount, 13
 - BEntry, 45
 - BNameValue, 86

- BSema, 133
- getVersion
 - Boapns, 10
 - Boapns::Boapns, 95
- getYearMicroSeconds
 - BTimeStamp, 169
- getYearMilliSeconds
 - BTimeStampMs, 173
- getYearSeconds
 - BTimeStamp, 169
 - BTimeStampMs, 173
- gmatch
 - BString.cpp, 244
- goTo
 - BList, 76
- hasKey
 - BDict, 36
 - BDictMap, 40
- hd32
 - BDebug.cpp, 190
 - BDebug.h, 192
- hd8
 - BDebug.cpp, 190
 - BDebug.h, 192
- hda32
 - BDebug.cpp, 190
- hda8
 - BDebug.cpp, 190
 - BDebug.h, 192
- hds32
 - BDebug.h, 192
- hostName
 - Boapns::BoapEntry, 93
- hour
 - BTimeStamp, 168
 - BTimeStampMs, 174
- increment
 - BCondInt, 25
 - BCondValue, 28
 - BCondWrap, 31
 - BFifoPos, 67
- Init
 - BString, 153
- init
 - BoapServer, 104
 - BoapServerConnection, 108
 - BRtc, 126
 - BRtcThreaded, 128
 - BSocket, 137
- insert
 - BEntryList, 51
 - BList, 77
 - BMySQL, 85
 - BString, 150
- insertAfter
 - BList, 77
- inString
 - BString, 153
- Int16
 - BoapSimple.h, 225
- Int32
 - BoapSimple.h, 225
- Int8
 - BoapSimple.h, 225
- IP_MTU
 - BSocket.cpp, 240
- IS_BIG_ENDIAN
 - Boap.cpp, 216
- isEnd
 - BDictMap, 40
 - BFile, 70
 - BList, 77
- isLeap
 - BTimeStamp, 169
 - BTimeStampMs, 174
- isSet
 - BEntryList, 50
 - BTimeStamp, 169
- isSpace
 - BString, 153
- item
 - BList::Node, 80
- iterator
 - BDict, 36
 - BDictMap, 40
- justify
 - BString, 150
- key
 - BDict, 36
 - BDictItem, 38
 - BDictMap, 40
- len
 - BRefData, 124
 - BSocketAddress, 140
 - BString, 148
 - BStringLocked, 155
- length
 - BFile, 70
 - BoapPacketHead, 101
- line
 - BEntry, 45
- listen
 - BSocket, 137

- lock
 - BMutex, 82
- mapCircularBuffer
 - BFifo, 64
- MDEBUG
 - BMutex.cpp, 211
- microSecond
 - BTimeStamp, 168
- milliSecond
 - BTimeStampMs, 175
- MINUS
 - BString.cpp, 244
- minute
 - BTimeStamp, 168
 - BTimeStampMs, 175
- mon_yday
 - BTimeStamp.cpp, 254
 - BTimeStampMs.cpp, 256
- month
 - BTimeStamp, 168
- name
 - Boapns::BoapEntry, 93
 - BoapServiceObject, 113
- nbytes
 - BoapPacket, 97
- newConnection
 - BoapServer, 104
- next
 - BDictMap, 40
 - BList, 76
 - BNode, 88
- nextFd
 - BPoll, 121
- Node
 - BList::Node, 80
- nodeCreate
 - BList, 79
- nodeGet
 - BList, 79
- NONE
 - BError, 53
- Normal
 - BMutex, 82
- NOTHREADS
 - BoapServer, 103
- nprintf
 - BDebug.h, 192
- NType
 - BSocket, 135
- number
 - BList, 76
- oaddress
 - BSocketAddress, 140
- oapiVersion
 - BoapClientObject, 92
 - BoapServiceObject, 113
- oaverage
 - BTimer, 164
- oboapNs
 - BoapServer, 106
- oboapns
 - BoapServer, 106
- oboapServer
 - BoapServerConnection, 109
- oclientGoneEvent
 - BoapServer, 106
- oclients
 - BoapServer, 106
- ocmd
 - BoapFuncEntry, 94
- ocolumnWidths
 - BTable, 158
- ocomments
 - BEntryFile, 48
- ocond
 - BCond, 21
 - BCondBool, 23
 - BCondInt, 26
 - BCondValue, 29
 - BCondWrap, 32
 - BRtcThreaded, 129
- oconnected
 - BoapClientObject, 92
- odata
 - BBuffer, 16
 - BFifo, 65
 - BoapPacket, 99
 - BRefData, 124
 - BTable, 158
- odataSize
 - BBuffer, 16
- odb
 - BMysql, 85
- odebug
 - BMysql, 85
- odirname
 - BDir, 43
- oendTime
 - BTimer, 164
- oerrNo
 - BError, 54
- oerror
 - BDir, 43
- oerrStr
 - BError, 54

- ofd
 - BRtc, 126
- ofds
 - BEventInt, 59
 - BEventPipe, 61
 - BPoll, 121
- ofdsNext
 - BPoll, 121
- ofdsNum
 - BPoll, 121
- ofile
 - BConfig, 33
 - BFile, 71
- ofilename
 - BFile, 71
- ofilename
 - BEntryFile, 48
- ofunc
 - BoapFuncEntry, 94
- ofuncList
 - BoapServiceObject, 113
- ohostName
 - BoapServer, 106
- ohour
 - BTimeStamp, 170
- oi
 - BIter, 72
- oinit
 - BUrl, 177
- oisBoapns
 - BoapServer, 106
- olastPos
 - BEntryList, 51
- olen
 - BRefData, 124
 - BSocketAddress, 140
- olength
 - BList, 79
- oclock
 - BConfig, 33
 - BFifo, 65
 - BMutexLock, 83
 - BMysql, 85
 - BoapClientObject, 92
 - BRWLock, 131
 - BStringLocked, 155
 - BTimer, 164
- omaxLength
 - BoapClientObject, 92
 - BoapServerConnection, 109
- omicroSecond
 - BTimeStamp, 170
- ominute
 - BTimeStamp, 170
- omode
 - BFile, 71
- omutex
 - BCond, 21
 - BCondBool, 23
 - BCondInt, 26
 - BCondValue, 29
 - BCondWrap, 32
 - BMutex, 82
- oname
 - BEntry, 46
 - BNameValue, 86
 - BoapClientObject, 92
 - BoapServiceObject, 113
- onbytes
 - BoapPacket, 99
- onet
 - BoapServer, 106
- onetEvent
 - BoapServer, 106
- onetEventAddress
 - BoapServer, 106
- onodes
 - BList, 79
- onum
 - BTimer, 164
- onumOperations
 - BoapServer, 106
- object
 - BoapServiceEntry, 110
- opened
 - BMysql, 85
- opeak
 - BTimer, 164
- open
 - BConfig, 33
 - BDir, 42
 - BEntryFile, 48
 - BFile, 69
 - BMysql, 85
- operator BNode *
 - BIter, 72
- operator BString
 - BStringLocked, 155
 - BTimeStamp, 169
- operator const char *
 - BString, 153
- operator const SockAddr *
 - BSocketAddress, 140
- operator int
 - BCondBool, 23
 - BError, 54
 - BFifoPos, 67
- operator long

- BAtomicCount, 13
- operator!=
 - BFifoPos, 67
 - BSocketAddress, 140
 - BString, 153
 - BTimeStamp, 169
- operator<
 - BString, 153
 - BTimeStamp, 169
 - BTimeStampMs, 174
- operator<<
 - BString.cpp, 244
 - BString.h, 246
- operator<=
 - BString, 153
 - BTimeStamp, 169
 - BTimeStampMs, 174
- operator>
 - BString, 153
 - BTimeStamp, 169
 - BTimeStampMs, 174
- operator>>
 - BString.cpp, 244
 - BString.h, 246
- operator>=
 - BString, 153
 - BTimeStamp, 169
 - BTimeStampMs, 174
- operator+
 - BDict, 37
 - BList, 79
 - BString, 153
 - BStringLocked, 155
- operator++
 - BAtomicCount, 13
 - BCondInt, 26
 - BCondValue, 28
 - BCondWrap, 32
- operator+=
 - BCondValue, 28
 - BCondWrap, 31
 - BFifoPos, 67
 - BString, 153
- operator-
 - BAtomicCount, 13
 - BCondInt, 26
 - BCondValue, 29
 - BCondWrap, 32
- operator-=
 - BCondValue, 28
 - BCondWrap, 31
- operator=
 - BDict, 37
 - BEntryList, 51
 - BFile, 71
 - BList, 79
 - BMutex, 82
 - BRefData, 124
 - BRWLock, 131
 - BSema, 133
 - BSocketAddress, 140
 - BString, 151
 - BStringLocked, 155
 - BTimeStamp, 169
- operator==
 - BFifoPos, 67
 - BIter, 72
 - BSocketAddress, 140
 - BString, 151, 153
 - BTimeStamp, 169
- opolicy
 - BThread, 161
- opoll
 - BoapServer, 106
- opos
 - BBufferStore, 20
 - BFifoPos, 67
 - BoapPacket, 99
- opriority
 - BoapClientObject, 92
 - BThread, 161
- orate
 - BRtc, 126
 - BRtcThreaded, 129
- oreadPos
 - BFifo, 65
- oreconnect
 - BoapClientObject, 92
- orefCount
 - BRefData, 124
- ores
 - BUrl, 177
- oresult
 - BThread, 161
- ortc
 - BRtcThreaded, 129
- orunning
 - BThread, 161
- orx
 - BoapClientObject, 92
 - BoapServer, 106
 - BoapServerConnection, 109
 - BoapSignalObject, 115
- osecond
 - BTimeStamp, 170
- osema
 - BSema, 133
- oserver

- BoapServiceObject, 113
- oservice
 - BoapClientObject, 92
 - BoapServiceEntry, 110
- oservices
 - BoapServer, 106
- osize
 - BBuffer, 16
 - BFifo, 65
 - BFifoPos, 67
 - BoapPacket, 99
- osocket
 - BoapServerConnection, 109
 - BSocket, 137
- osort
 - BDir, 43
- ospare
 - BTimeStamp, 170
- ostackSize
 - BThread, 161
- ostartTime
 - BTimer, 164
- ostr
 - BString, 153
 - BStringLocked, 155
- othread
 - BThread, 161
- othreaded
 - BoapServer, 106
- otimeout
 - BoapClientObject, 92
- otitle
 - BTable, 158
- otx
 - BoapClientObject, 92
 - BoapServer, 106
 - BoapServerConnection, 109
 - BoapSignalObject, 115
- otype
 - BEvent, 56
 - BObject, 119
- ovalue
 - BAtomicCount, 13
 - BCondBool, 23
 - BCondInt, 26
 - BCondValue, 29
 - BCondWrap, 32
 - BEntry, 46
 - BNameValue, 86
- ovmSize
 - BFifo, 65
- owild
 - BDir, 43
- owriteNumFifoSamples
 - BFifo, 65
- owritePos
 - BFifo, 65
- oyday
 - BTimeStamp, 169
- oyear
 - BTimeStamp, 169
- pad
 - BString, 149
- peak
 - BTimer, 163
- peekHead
 - BoapPacket, 97
- performCall
 - BoapClientObject, 91, 92
- performRecv
 - BoapClientObject, 91, 92
- performSend
 - BoapClientObject, 91, 92
 - BoapSignalObject, 115
- ping
 - BoapClientObject, 90
- pingLocked
 - BoapClientObject, 91
- PollFd
 - BPoll, 121
- pop
 - BBufferStore, 19, 20
 - BList, 78
 - BoapPacket, 98, 99
- popHead
 - BoapPacket, 97, 98
- port
 - Boapns::BoapEntry, 93
 - BSocketAddressINET, 142
- pos
 - BFifoPos, 66
- position
 - BFile, 70
 - BList, 76
- post
 - BSema, 132
- prev
 - BList, 76
 - BNode, 88
- print
 - BEntry, 45
 - BEntryList, 50
 - BTable, 158
- printf
 - BFile, 71
 - BString, 150
- printIt

- BObject, 119
- printLine
 - BTable, 158
- Priority
 - BSocket, 135
- PriorityHigh
 - BSocket, 135
- PriorityLow
 - BSocket, 135
- PriorityNormal
 - BSocket, 135
- process
 - BoapServer, 104, 106
 - BoapServerConnection, 108
 - BoapServiceObject, 113
- processEvent
 - BoapServer, 104, 106
 - BoapServiceObject, 113
- pullLine
 - BString, 151
- pullSeparators
 - BString, 151
- pullToken
 - BString, 150
- pullWord
 - BString, 151
- push
 - BBufferStore, 18, 19
 - BList, 78
 - BoapPacket, 98
- pushHead
 - BoapPacket, 97
- query
 - BMySQL, 85
- queueAdd
 - BList, 78
- queueGet
 - BList, 78
- raw
 - BSocketAddress, 140
- rdLock
 - BRWLock, 130
- read
 - BConfig, 33
 - BDir, 42
 - BEntryFile, 48
 - BFifo, 64
 - BFile, 70
- readAvailable
 - BFifo, 64
- readData
 - BFifo, 64
- readDone
 - BFifo, 64
- readString
 - BFile, 70
 - BUrl, 176
- readWaitAvailable
 - BFifo, 64
- rear
 - BList, 77
- Recursive
 - BMutex, 82
- recv
 - BSocket, 137
- recvFrom
 - BSocket, 137
- recvFromWithTimeout
 - BSocket, 137
- recvWithTimeout
 - BSocket, 137
- removeNL
 - BString, 149
- removeSeparators
 - BString, 150
- reserved
 - BoapPacketHead, 101
- resize
 - BBuffer, 16
 - BoapPacket, 97
- result
 - BThread, 161
- retDouble
 - BString, 149
- retInt
 - BString, 148
- retStr
 - BString, 148
- retStrDup
 - BString, 148
- retUInt
 - BString, 149
- roundSize
 - BBuffer.cpp, 181
 - BoapSimple.cc, 223
- run
 - BoapServer, 104, 106
- running
 - BThread, 161
- sampleNumber
 - BTimeStampMs, 175
- second
 - BTimeStamp, 168
 - BTimeStampMs, 175
- seek

- BFile, 70
- send
 - BSocket, 137
- sendEvent
 - BEventInt, 58
 - BEventPipe, 60
 - BoapServer, 104, 106
 - BoapServiceObject, 113
- sendTo
 - BSocket, 137
- service
 - Boapns::BoapEntry, 93
 - BoapPacketHead, 101
- set
 - BCondBool, 22
 - BError, 53
 - BFifoPos, 66
 - BSocketAddress, 140
 - BSocketAddressINET, 142
 - BTimeStamp, 167
- setBinary
 - BEvent, 55
 - BEventError, 57
 - BObject, 119
- setBroadCast
 - BSocket, 137
- setConnectionPriority
 - BoapClientObject, 90
- setData
 - BBuffer, 16
 - BoapPacket, 97
- setDebug
 - BDebug.cpp, 190
 - BDebug.h, 192
 - BMysql, 85
- setDurationString
 - BTimeStampMs, 174
- setError
 - BError, 53
- setFirst
 - BTimeStamp, 167
- setHexString
 - BBufferStore, 18
- setInitPriority
 - BThread, 161
- setInitStackSize
 - BThread, 161
- setLast
 - BTimeStamp, 167
- setLen
 - BRefData, 124
- setLine
 - BEntry, 45
- setMaxLength
 - BoapClientObject, 91
 - BoapServerConnection, 109
- setMembers
 - BObj, 117
- setName
 - BEntry, 45
 - BoapServiceObject, 113
- setNow
 - BTimeStamp, 168
 - BTimeStampMs, 173
- setPort
 - BSocketAddressINET, 142
- setPos
 - BBufferStore, 18
- setPriority
 - BSocket, 137
 - BThread, 161
- setReuseAddress
 - BSocket, 137
- setSize
 - BBuffer, 16
 - BFifoPos, 66
- setSockOpt
 - BSocket, 137
- setSort
 - BDir, 42
- setString
 - BObject, 119
 - BTimeStamp, 168
 - BTimeStampMs, 173
- setTime
 - BTimeStamp, 168
- setTimeout
 - BoapClientObject, 91
- setTitle
 - BTable, 158
- setValue
 - BCondInt, 25
 - BCondValue, 28
 - BCondWrap, 31
 - BEntry, 45
 - BEntryList, 50
- setValueRaw
 - BEntryList, 50
- setVBuf
 - BFile, 70
- setWild
 - BDir, 42
- setYDay
 - BTimeStamp, 167
- shutdown
 - BSocket, 137
- signal
 - BCond, 21

- size
 - BBuffer, 16
 - BDictMap, 40
 - BFifo, 63
 - BList, 77
- SocketAddr
 - BSocketAddress, 140
- SocketAddrIP
 - BSocketAddressINET, 142
- sort
 - BList, 78
- SortFunc
 - BList, 76
- start
 - BDictMap, 40
 - BList, 76
 - BThread, 161
 - BTimer, 163
- startFunc
 - BThread, 161
- std::vector, 178
- stop
 - BTimer, 163
- STRBUF
 - BFile.cpp, 207
- STRBUF_SIZE
 - BDebug.cpp, 190
- STREAM
 - BSocket, 135
- STRIP
 - BString.cpp, 244
- subMilliseconds
 - BTimeStampMs, 173
- subSeconds
 - BTimeStampMs, 173
- subString
 - BString, 150
- swap
 - BList, 78
- THREADED
 - BoapServer, 103
- timedLock
 - BMutex, 82
- timedWait
 - BCond, 21
 - BCondBool, 23
 - BCondInt, 25
 - BSema, 133
- toBString
 - BDict.cpp, 193
 - BDict.h, 194
 - BString.cpp, 244
 - BString.h, 246
 - BTimeStamp.cpp, 254
 - BTimeStamp.h, 255
- toLower
 - BString, 149
- toUpper
 - BString, 149
- tprintf
 - BDebug.cpp, 190
 - BDebug.h, 192
- truncate
 - BFile, 71
 - BString, 149
- tryLock
 - BMutex, 82
- tryNotZeroDecrement
 - BCondInt, 25
- tryRdLock
 - BRWLock, 130
- tryWait
 - BSema, 133
- tryWrLock
 - BRWLock, 131
- Type
 - BError, 53
 - BMutex, 82
- type
 - BoapPacketHead, 101
- UInt16
 - BoapSimple.h, 225
- UInt32
 - BoapSimple.h, 225
- UInt8
 - BoapSimple.h, 225
- unlock
 - BMutex, 82
 - BRWLock, 131
- unmapCircularBuffer
 - BFifo, 65
- update
 - BMysql, 85
- updateHead
 - BoapPacket, 97
- updateLen
 - BoapPacket, 99
- validate
 - BoapServerConnection, 109
- value
 - BCondBool, 23
 - BCondInt, 25
 - BCondValue, 28
 - BCondWrap, 31
 - BDictItem, 38

- wait
 - BCond, 21
 - BCondBool, 23
 - BCondInt, 25
 - BRtc, 126
 - BRtcThreaded, 128
 - BSema, 132
- waitForCompletion
 - BThread, 161
- waitIncrement
 - BCondInt, 25
- waitLessThan
 - BCondValue, 28
 - BCondWrap, 31
- waitLessThanOrEqual
 - BCondValue, 28
 - BCondWrap, 31
- waitMoreThanOrEqual
 - BCondValue, 28
 - BCondWrap, 31
- waitNotZero
 - BCondInt, 25
- waitNotZeroDecrement
 - BCondInt, 25
- wild
 - BDir.cpp, 196
- wildString
 - BDir.cpp, 196
- wprintf
 - BDebug.h, 192
- write
 - BConfig, 33
 - BEntryFile, 48
 - BFifo, 64
 - BFile, 70
- writeAvailable
 - BFifo, 63
- writeData
 - BBuffer, 16
 - BFifo, 64
 - BUrl, 176
- writeDone
 - BFifo, 64
- writeList
 - BEntryFile, 48
- writeString
 - BFile, 70
- writeWaitAvailable
 - BFifo, 64
- wrLock
 - BRWLock, 131
- yday
 - BTimeStamp, 168
- BTimeStampMs, 174
- year
 - BTimeStamp, 168
 - BTimeStampMs, 174