

LibBeamApi

0.3.8

Generated by Doxygen 1.5.6

Thu Dec 18 13:21:31 2008

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
6.2	BBuffer Class Reference	12
6.2.1	Constructor & Destructor Documentation	12
6.2.1.1	BBuffer	12
6.2.1.2	~BBuffer	12
6.2.2	Member Function Documentation	12
6.2.2.1	setSize	12
6.2.2.2	setData	12
6.2.2.3	writeData	13
6.2.2.4	data	13
6.2.2.5	size	13
6.2.3	Member Data Documentation	13
6.2.3.1	osize	13
6.2.3.2	odatasize	13
6.2.3.3	odata	13
6.3	BCond Class Reference	14
6.3.1	Constructor & Destructor Documentation	14
6.3.1.1	BCond	14
6.3.1.2	~BCond	14
6.3.2	Member Function Documentation	14
6.3.2.1	signal	14
6.3.2.2	wait	14
6.3.2.3	timedWait	14
6.3.3	Member Data Documentation	14
6.3.3.1	omutex	14
6.3.3.2	ocond	14
6.4	BCondBool Class Reference	15
6.4.1	Detailed Description	15
6.4.2	Constructor & Destructor Documentation	15
6.4.2.1	BCondBool	15
6.4.2.2	~BCondBool	15
6.4.3	Member Function Documentation	15
6.4.3.1	set	15
6.4.3.2	clear	16
6.4.3.3	value	16

6.4.3.4	wait	16
6.4.3.5	timedWait	16
6.4.4	Member Data Documentation	16
6.4.4.1	omutex	16
6.4.4.2	ocond	16
6.4.4.3	ovalue	16
6.5	BCondInt Class Reference	17
6.5.1	Detailed Description	18
6.5.2	Constructor & Destructor Documentation	18
6.5.2.1	BCondInt	18
6.5.2.2	~BCondInt	18
6.5.3	Member Function Documentation	18
6.5.3.1	setValue	18
6.5.3.2	increment	18
6.5.3.3	decrement	18
6.5.3.4	value	18
6.5.3.5	wait	18
6.5.3.6	waitIncrement	18
6.5.3.7	waitNotZero	18
6.5.3.8	waitNotZeroDecrement	18
6.5.3.9	tryNotZeroDecrement	18
6.5.3.10	timedWait	19
6.5.3.11	operator++	19
6.5.3.12	operator--	19
6.5.4	Member Data Documentation	19
6.5.4.1	omutex	19
6.5.4.2	ocond	19
6.5.4.3	ovalue	19
6.6	BCondValue Class Reference	20
6.6.1	Detailed Description	21
6.6.2	Constructor & Destructor Documentation	21
6.6.2.1	BCondValue	21
6.6.2.2	~BCondValue	21
6.6.3	Member Function Documentation	21
6.6.3.1	setValue	21
6.6.3.2	value	21

6.6.3.3	increment	21
6.6.3.4	decrement	21
6.6.3.5	waitMoreThanOrEqual	21
6.6.3.6	waitLessThanOrEqual	21
6.6.3.7	waitLessThan	21
6.6.3.8	operator+=	21
6.6.3.9	operator-=	21
6.6.3.10	operator++	22
6.6.3.11	operator--	22
6.6.4	Member Data Documentation	22
6.6.4.1	omutex	22
6.6.4.2	ocond	22
6.6.4.3	ovalue	22
6.7	BCondWrap Class Reference	23
6.7.1	Constructor & Destructor Documentation	24
6.7.1.1	BCondWrap	24
6.7.1.2	~BCondWrap	24
6.7.2	Member Function Documentation	24
6.7.2.1	setValue	24
6.7.2.2	value	24
6.7.2.3	increment	24
6.7.2.4	decrement	24
6.7.2.5	waitMoreThanOrEqual	24
6.7.2.6	waitLessThanOrEqual	24
6.7.2.7	waitLessThan	24
6.7.2.8	operator+=	24
6.7.2.9	operator-=	25
6.7.2.10	operator++	25
6.7.2.11	operator--	25
6.7.2.12	diff	25
6.7.3	Member Data Documentation	25
6.7.3.1	omutex	25
6.7.3.2	ocond	25
6.7.3.3	ovalue	25
6.8	BDir Class Reference	26
6.8.1	Detailed Description	27

6.8.2	Constructor & Destructor Documentation	27
6.8.2.1	BDir	27
6.8.2.2	BDir	27
6.8.2.3	~BDir	27
6.8.3	Member Function Documentation	27
6.8.3.1	open	27
6.8.3.2	error	27
6.8.3.3	read	27
6.8.3.4	clear	27
6.8.3.5	setWild	27
6.8.3.6	setSort	27
6.8.3.7	entryName	28
6.8.3.8	entryStat	28
6.8.3.9	entryStat64	28
6.8.4	Member Data Documentation	28
6.8.4.1	oerror	28
6.8.4.2	odirname	28
6.8.4.3	owild	28
6.8.4.4	osort	28
6.9	BEntry Class Reference	29
6.9.1	Detailed Description	29
6.9.2	Constructor & Destructor Documentation	30
6.9.2.1	BEntry	30
6.9.2.2	BEntry	30
6.9.2.3	BEntry	30
6.9.3	Member Function Documentation	30
6.9.3.1	getName	30
6.9.3.2	getValue	30
6.9.3.3	setLine	30
6.9.3.4	setName	30
6.9.3.5	setValue	30
6.9.3.6	line	30
6.9.3.7	print	30
6.9.4	Member Data Documentation	31
6.9.4.1	oname	31
6.9.4.2	ovalue	31

6.10 BEntryFile Class Reference	32
6.10.1 Detailed Description	32
6.10.2 Constructor & Destructor Documentation	33
6.10.2.1 BEntryFile	33
6.10.2.2 BEntryFile	33
6.10.2.3 ~BEntryFile	33
6.10.3 Member Function Documentation	33
6.10.3.1 open	33
6.10.3.2 read	33
6.10.3.3 write	33
6.10.3.4 writeList	33
6.10.3.5 clear	33
6.10.4 Member Data Documentation	33
6.10.4.1 ofilename	33
6.10.4.2 ocomments	33
6.11 BEntryList Class Reference	34
6.11.1 Detailed Description	35
6.11.2 Constructor & Destructor Documentation	35
6.11.2.1 BEntryList	35
6.11.3 Member Function Documentation	35
6.11.3.1 isSet	35
6.11.3.2 find	35
6.11.3.3 findValue	35
6.11.3.4 setValue	35
6.11.3.5 setValueRaw	35
6.11.3.6 deleteEntry	35
6.11.3.7 print	35
6.11.3.8 getString	35
6.11.3.9 insert	36
6.11.3.10 del	36
6.11.3.11 clear	36
6.11.4 Member Data Documentation	36
6.11.4.1 olastPos	36
6.12 BError Class Reference	37
6.12.1 Detailed Description	38
6.12.2 Member Enumeration Documentation	38

6.12.2.1	Type	38
6.12.3	Constructor & Destructor Documentation	38
6.12.3.1	BError	38
6.12.3.2	BError	38
6.12.4	Member Function Documentation	38
6.12.4.1	copy	38
6.12.4.2	set	38
6.12.4.3	setError	38
6.12.4.4	getString	38
6.12.4.5	getErrorNo	38
6.12.4.6	operator int	39
6.12.5	Member Data Documentation	39
6.12.5.1	oerrNo	39
6.12.5.2	oerrStr	39
6.13	BEvent Class Reference	40
6.13.1	Detailed Description	40
6.13.2	Constructor & Destructor Documentation	40
6.13.2.1	BEvent	40
6.13.2.2	~BEvent	40
6.13.3	Member Function Documentation	40
6.13.3.1	getType	40
6.13.3.2	getBinary	40
6.13.3.3	setBinary	40
6.13.4	Member Data Documentation	41
6.13.4.1	otype	41
6.14	BEventError Class Reference	42
6.14.1	Constructor & Destructor Documentation	42
6.14.1.1	BEventError	42
6.14.2	Member Function Documentation	42
6.14.2.1	getBinary	42
6.14.2.2	setBinary	42
6.15	BEventInt Class Reference	43
6.15.1	Detailed Description	43
6.15.2	Constructor & Destructor Documentation	43
6.15.2.1	BEventInt	43
6.15.2.2	~BEventInt	43

6.15.3	Member Function Documentation	43
6.15.3.1	sendEvent	43
6.15.3.2	getEvent	43
6.15.3.3	getFd	44
6.15.4	Member Data Documentation	44
6.15.4.1	ofds	44
6.16	BEventPipe Class Reference	45
6.16.1	Detailed Description	45
6.16.2	Constructor & Destructor Documentation	45
6.16.2.1	BEventPipe	45
6.16.2.2	~BEventPipe	45
6.16.3	Member Function Documentation	45
6.16.3.1	sendEvent	45
6.16.3.2	getEvent	45
6.16.3.3	getReceiveFd	46
6.16.4	Member Data Documentation	46
6.16.4.1	ofds	46
6.17	BFile Class Reference	47
6.17.1	Detailed Description	48
6.17.2	Constructor & Destructor Documentation	48
6.17.2.1	BFile	48
6.17.2.2	BFile	48
6.17.2.3	BFile	48
6.17.2.4	~BFile	48
6.17.3	Member Function Documentation	48
6.17.3.1	open	48
6.17.3.2	open	48
6.17.3.3	close	48
6.17.3.4	error	49
6.17.3.5	getFd	49
6.17.3.6	length	49
6.17.3.7	setVBuf	49
6.17.3.8	read	49
6.17.3.9	readString	49
6.17.3.10	write	49
6.17.3.11	writeString	49

6.17.3.12 seek	49
6.17.3.13 printf	49
6.17.3.14 operator=	50
6.17.4 Member Data Documentation	50
6.17.4.1 ofile	50
6.17.4.2 ofileName	50
6.17.4.3 omode	50
6.17.4.4 oerror	50
6.18 BIter Class Reference	51
6.18.1 Detailed Description	51
6.18.2 Constructor & Destructor Documentation	51
6.18.2.1 BIter	51
6.18.3 Member Function Documentation	51
6.18.3.1 operator void *	51
6.18.3.2 operator==	51
6.18.4 Member Data Documentation	51
6.18.4.1 oi	51
6.19 BList< T > Class Template Reference	52
6.19.1 Detailed Description	54
6.19.2 Member Typedef Documentation	54
6.19.2.1 SortFunc	54
6.19.3 Constructor & Destructor Documentation	55
6.19.3.1 BList	55
6.19.3.2 BList	55
6.19.3.3 ~BList	55
6.19.4 Member Function Documentation	55
6.19.4.1 start	55
6.19.4.2 begin	55
6.19.4.3 end	55
6.19.4.4 end	55
6.19.4.5 next	55
6.19.4.6 prev	55
6.19.4.7 goTo	55
6.19.4.8 position	55
6.19.4.9 number	55
6.19.4.10 isEnd	56

6.19.4.11	front	56
6.19.4.12	rear	56
6.19.4.13	get	56
6.19.4.14	get	56
6.19.4.15	append	56
6.19.4.16	insert	56
6.19.4.17	insertAfter	56
6.19.4.18	clear	56
6.19.4.19	del	56
6.19.4.20	deleteLast	57
6.19.4.21	deleteFirst	57
6.19.4.22	push	57
6.19.4.23	pop	57
6.19.4.24	queueAdd	57
6.19.4.25	queueGet	57
6.19.4.26	append	57
6.19.4.27	swap	57
6.19.4.28	sort	57
6.19.4.29	sort	57
6.19.4.30	operator=	57
6.19.4.31	operator[57
6.19.4.32	operator[58
6.19.4.33	operator[58
6.19.4.34	operator[58
6.19.4.35	operator+	58
6.19.4.36	nodeGet	58
6.19.4.37	nodeGet	58
6.19.4.38	nodeCreate	58
6.19.4.39	nodeCreate	58
6.19.5	Member Data Documentation	58
6.19.5.1	onodes	58
6.19.5.2	olength	58
6.20	BList< T >::Node Class Reference	59
6.20.1	Constructor & Destructor Documentation	59
6.20.1.1	Node	59
6.20.2	Member Data Documentation	59

6.20.2.1	next	59
6.20.2.2	prev	59
6.20.2.3	item	59
6.21	BMutex Class Reference	60
6.21.1	Detailed Description	60
6.21.2	Constructor & Destructor Documentation	60
6.21.2.1	BMutex	60
6.21.2.2	BMutex	60
6.21.2.3	~BMutex	60
6.21.3	Member Function Documentation	60
6.21.3.1	lock	60
6.21.3.2	unlock	60
6.21.3.3	tryLock	61
6.21.3.4	operator=	61
6.21.4	Member Data Documentation	61
6.21.4.1	omutex	61
6.22	BNameValue< T > Class Template Reference	62
6.22.1	Constructor & Destructor Documentation	62
6.22.1.1	BNameValue	62
6.22.1.2	BNameValue	62
6.22.2	Member Function Documentation	62
6.22.2.1	getName	62
6.22.2.2	getValue	62
6.22.3	Member Data Documentation	62
6.22.3.1	oname	62
6.22.3.2	ovalue	62
6.23	BNameValueList< T > Class Template Reference	63
6.23.1	Member Function Documentation	63
6.23.1.1	find	63
6.24	BoapClientObject Class Reference	64
6.24.1	Constructor & Destructor Documentation	65
6.24.1.1	BoapClientObject	65
6.24.1.2	BoapClientObject	65
6.24.2	Member Function Documentation	65
6.24.2.1	connectService	65
6.24.2.2	disconnectService	65

6.24.2.3	getServiceName	65
6.24.2.4	ping	65
6.24.2.5	setConnectionPriority	66
6.24.2.6	setMaxLength	66
6.24.2.7	setTimeout	66
6.24.2.8	pingLocked	66
6.24.2.9	checkApiVersion	66
6.24.2.10	performCall	66
6.24.2.11	performSend	66
6.24.2.12	performRecv	66
6.24.2.13	connectService	67
6.24.2.14	performSend	67
6.24.2.15	performRecv	67
6.24.2.16	performCall	67
6.24.3	Member Data Documentation	67
6.24.3.1	oname	67
6.24.3.2	oapiVersion	67
6.24.3.3	opriority	67
6.24.3.4	oservice	67
6.24.3.5	oconnected	67
6.24.3.6	omaxLength	67
6.24.3.7	otx	67
6.24.3.8	orx	67
6.24.3.9	oclock	67
6.24.3.10	otimeout	67
6.24.3.11	oreconnect	67
6.25	Boapns::BoapEntry Class Reference	68
6.25.1	Constructor & Destructor Documentation	68
6.25.1.1	BoapEntry	68
6.25.1.2	BoapEntry	68
6.25.2	Member Data Documentation	68
6.25.2.1	name	68
6.25.2.2	hostName	68
6.25.2.3	addressList	68
6.25.2.4	port	68
6.25.2.5	service	68

6.26	BoapFuncEntry Class Reference	69
6.26.1	Constructor & Destructor Documentation	69
6.26.1.1	BoapFuncEntry	69
6.26.1.2	BoapFuncEntry	69
6.26.2	Member Data Documentation	69
6.26.2.1	oCmd	69
6.26.2.2	oFunc	69
6.27	Boapns::Boapns Class Reference	70
6.27.1	Constructor & Destructor Documentation	70
6.27.1.1	Boapns	70
6.27.2	Member Function Documentation	70
6.27.2.1	getVersion	70
6.27.2.2	getEntryList	70
6.27.2.3	getEntry	70
6.27.2.4	addEntry	70
6.27.2.5	delEntry	70
6.27.2.6	getNewName	70
6.28	BoapPacket Class Reference	71
6.28.1	Constructor & Destructor Documentation	74
6.28.1.1	BoapPacket	74
6.28.1.2	~BoapPacket	74
6.28.1.3	BoapPacket	74
6.28.1.4	~BoapPacket	74
6.28.2	Member Function Documentation	74
6.28.2.1	resize	74
6.28.2.2	setData	74
6.28.2.3	nbytes	74
6.28.2.4	data	74
6.28.2.5	peekHead	74
6.28.2.6	getCmd	74
6.28.2.7	pushHead	74
6.28.2.8	push	74
6.28.2.9	push	74
6.28.2.10	push	74
6.28.2.11	push	74
6.28.2.12	push	74

6.28.2.13 push	74
6.28.2.14 push	74
6.28.2.15 push	74
6.28.2.16 push	74
6.28.2.17 push	74
6.28.2.18 push	74
6.28.2.19 push	74
6.28.2.20 popHead	74
6.28.2.21 pop	74
6.28.2.22 pop	74
6.28.2.23 pop	74
6.28.2.24 pop	74
6.28.2.25 pop	74
6.28.2.26 pop	74
6.28.2.27 pop	74
6.28.2.28 pop	74
6.28.2.29 pop	74
6.28.2.30 pop	74
6.28.2.31 pop	74
6.28.2.32 pop	74
6.28.2.33 copyWithSwap	74
6.28.2.34 updateLen	74
6.28.2.35 resize	74
6.28.2.36 setData	74
6.28.2.37 nbytes	74
6.28.2.38 data	74
6.28.2.39 pushHead	74
6.28.2.40 push	74
6.28.2.41 push	74
6.28.2.42 push	74
6.28.2.43 push	74
6.28.2.44 push	74
6.28.2.45 push	74
6.28.2.46 push	74
6.28.2.47 push	74
6.28.2.48 push	74

6.28.2.49	push	74
6.28.2.50	popHead	74
6.28.2.51	pop	74
6.28.2.52	pop	74
6.28.2.53	pop	74
6.28.2.54	pop	74
6.28.2.55	pop	74
6.28.2.56	pop	74
6.28.2.57	pop	74
6.28.2.58	pop	74
6.28.2.59	pop	74
6.28.2.60	pop	74
6.28.2.61	updateLen	74
6.28.3	Member Data Documentation	74
6.28.3.1	osize	74
6.28.3.2	onbytes	74
6.28.3.3	odata	74
6.28.3.4	opos	74
6.29	BoapPacketHead Struct Reference	76
6.29.1	Member Data Documentation	76
6.29.1.1	type	76
6.29.1.2	length	76
6.29.1.3	service	76
6.29.1.4	cmd	76
6.29.1.5	type	76
6.29.1.6	service	76
6.29.1.7	reserved	76
6.30	BoapServer Class Reference	77
6.30.1	Member Enumeration Documentation	78
6.30.1.1	"@0	78
6.30.2	Constructor & Destructor Documentation	79
6.30.2.1	BoapServer	79
6.30.2.2	~BoapServer	79
6.30.2.3	BoapServer	79
6.30.3	Member Function Documentation	79
6.30.3.1	init	79

6.30.3.2	run	79
6.30.3.3	processEvent	79
6.30.3.4	addObject	79
6.30.3.5	process	79
6.30.3.6	sendEvent	79
6.30.3.7	getSocket	79
6.30.3.8	getEventSocket	79
6.30.3.9	processEvent	79
6.30.3.10	getHostName	79
6.30.3.11	clientGone	79
6.30.3.12	getConnectionsNumber	79
6.30.3.13	function	79
6.30.3.14	init	80
6.30.3.15	run	80
6.30.3.16	processEvent	80
6.30.3.17	addObject	80
6.30.3.18	process	80
6.30.3.19	sendEvent	80
6.30.3.20	getSocket	80
6.30.3.21	getEventSocket	80
6.30.3.22	processEvent	80
6.30.3.23	getHostName	80
6.30.4	Member Data Documentation	80
6.30.4.1	othreaded	80
6.30.4.2	oisBoapns	80
6.30.4.3	oboapns	80
6.30.4.4	oclients	80
6.30.4.5	oclientGoneEvent	80
6.30.4.6	oservices	80
6.30.4.7	opoll	80
6.30.4.8	onet	80
6.30.4.9	onetEvent	80
6.30.4.10	onetEventAddress	80
6.30.4.11	ohostName	80
6.30.4.12	oboapNs	80
6.30.4.13	orx	80

6.30.4.14	otx	80
6.31	BoapServerConnection Class Reference	82
6.31.1	Constructor & Destructor Documentation	82
6.31.1.1	BoapServerConnection	82
6.31.2	Member Function Documentation	82
6.31.2.1	process	82
6.31.2.2	getSocket	82
6.31.2.3	setMaxLength	82
6.31.2.4	function	82
6.31.3	Member Data Documentation	83
6.31.3.1	oboapServer	83
6.31.3.2	osocket	83
6.31.3.3	orx	83
6.31.3.4	otx	83
6.31.3.5	omaxLength	83
6.32	BoapServiceEntry Class Reference	84
6.32.1	Constructor & Destructor Documentation	84
6.32.1.1	BoapServiceEntry	84
6.32.1.2	BoapServiceEntry	84
6.32.2	Member Data Documentation	84
6.32.2.1	oservice	84
6.32.2.2	oobject	84
6.33	BoapServiceObject Class Reference	85
6.33.1	Constructor & Destructor Documentation	87
6.33.1.1	BoapServiceObject	87
6.33.1.2	~BoapServiceObject	87
6.33.1.3	BoapServiceObject	87
6.33.1.4	~BoapServiceObject	87
6.33.2	Member Function Documentation	87
6.33.2.1	setName	87
6.33.2.2	sendEvent	87
6.33.2.3	processEvent	87
6.33.2.4	name	87
6.33.2.5	doPing	87
6.33.2.6	doConnectionPriority	87
6.33.2.7	process	87

6.33.2.8	processEvent	87
6.33.2.9	sendEvent	87
6.33.2.10	sendEvent	87
6.33.2.11	processEvent	87
6.33.2.12	name	87
6.33.2.13	process	87
6.33.2.14	processEvent	87
6.33.2.15	sendEvent	87
6.33.3	Member Data Documentation	87
6.33.3.1	oserver	87
6.33.3.2	oname	87
6.33.3.3	oapiVersion	87
6.33.3.4	ofuncList	87
6.34	BoapSignalObject Class Reference	89
6.34.1	Constructor & Destructor Documentation	89
6.34.1.1	BoapSignalObject	89
6.34.1.2	BoapSignalObject	89
6.34.2	Member Function Documentation	89
6.34.2.1	performSend	89
6.34.2.2	performSend	89
6.34.3	Member Data Documentation	89
6.34.3.1	otx	89
6.34.3.2	orx	89
6.35	BObject Class Reference	91
6.35.1	Constructor & Destructor Documentation	92
6.35.1.1	BObject	92
6.35.1.2	~BObject	92
6.35.2	Member Function Documentation	92
6.35.2.1	getBinary	92
6.35.2.2	setBinary	92
6.35.2.3	getString	92
6.35.2.4	setString	92
6.35.2.5	getMemberList	92
6.35.2.6	addMember	92
6.35.2.7	printIt	92
6.35.2.8	getType	92

6.35.2.9	createObj	92
6.35.3	Member Data Documentation	92
6.35.3.1	otype	92
6.36	BPoll Class Reference	93
6.36.1	Detailed Description	93
6.36.2	Member Typedef Documentation	94
6.36.2.1	PollFd	94
6.36.3	Constructor & Destructor Documentation	94
6.36.3.1	BPoll	94
6.36.3.2	~BPoll	94
6.36.4	Member Function Documentation	94
6.36.4.1	append	94
6.36.4.2	delFd	94
6.36.4.3	doPoll	94
6.36.4.4	getPollFdsNum	94
6.36.4.5	getPollFds	94
6.36.4.6	clear	94
6.36.4.7	nextFd	94
6.36.5	Member Data Documentation	94
6.36.5.1	ofdsNum	94
6.36.5.2	ofds	94
6.36.5.3	ofdsNext	94
6.37	BRefData Class Reference	96
6.37.1	Detailed Description	96
6.37.2	Constructor & Destructor Documentation	97
6.37.2.1	BRefData	97
6.37.2.2	BRefData	97
6.37.2.3	BRefData	97
6.37.2.4	~BRefData	97
6.37.3	Member Function Documentation	97
6.37.3.1	copy	97
6.37.3.2	addRef	97
6.37.3.3	deleteRef	97
6.37.3.4	refCount	97
6.37.3.5	data	97
6.37.3.6	len	97

6.37.3.7	setLen	97
6.37.3.8	operator=	97
6.37.4	Member Data Documentation	97
6.37.4.1	oData	97
6.37.4.2	oLen	97
6.37.4.3	oSize	97
6.37.4.4	oRefCount	97
6.38	BRtc Class Reference	98
6.38.1	Detailed Description	98
6.38.2	Constructor & Destructor Documentation	98
6.38.2.1	BRtc	98
6.38.2.2	~BRtc	98
6.38.3	Member Function Documentation	98
6.38.3.1	init	98
6.38.3.2	wait	98
6.38.4	Member Data Documentation	98
6.38.4.1	ofd	98
6.38.4.2	orate	98
6.39	BRtcThreaded Class Reference	100
6.39.1	Detailed Description	100
6.39.2	Constructor & Destructor Documentation	100
6.39.2.1	BRtcThreaded	100
6.39.2.2	~BRtcThreaded	100
6.39.3	Member Function Documentation	100
6.39.3.1	init	100
6.39.3.2	wait	101
6.39.3.3	function	101
6.39.4	Member Data Documentation	101
6.39.4.1	ortc	101
6.39.4.2	orate	101
6.39.4.3	ocond	101
6.40	BRWLock Class Reference	102
6.40.1	Detailed Description	102
6.40.2	Constructor & Destructor Documentation	102
6.40.2.1	BRWLock	102
6.40.2.2	BRWLock	102

6.40.2.3	~BRWLock	102
6.40.3	Member Function Documentation	102
6.40.3.1	rdLock	102
6.40.3.2	tryRdLock	103
6.40.3.3	wrLock	103
6.40.3.4	tryWrLock	103
6.40.3.5	unlock	103
6.40.3.6	operator=	103
6.40.4	Member Data Documentation	103
6.40.4.1	oLock	103
6.41	BSema Class Reference	104
6.41.1	Detailed Description	104
6.41.2	Constructor & Destructor Documentation	104
6.41.2.1	BSema	104
6.41.2.2	BSema	104
6.41.2.3	~BSema	104
6.41.3	Member Function Documentation	104
6.41.3.1	post	104
6.41.3.2	wait	105
6.41.3.3	timedWait	105
6.41.3.4	tryWait	105
6.41.3.5	getValue	105
6.41.3.6	operator=	105
6.41.4	Member Data Documentation	105
6.41.4.1	osema	105
6.42	BSocket Class Reference	106
6.42.1	Member Enumeration Documentation	107
6.42.1.1	NType	107
6.42.1.2	Priority	107
6.42.2	Constructor & Destructor Documentation	109
6.42.2.1	BSocket	109
6.42.2.2	BSocket	109
6.42.2.3	BSocket	109
6.42.2.4	~BSocket	109
6.42.3	Member Function Documentation	109
6.42.3.1	init	109

6.42.3.2	getFd	109
6.42.3.3	bind	109
6.42.3.4	connect	109
6.42.3.5	shutdown	109
6.42.3.6	close	109
6.42.3.7	listen	109
6.42.3.8	accept	109
6.42.3.9	accept	109
6.42.3.10	send	109
6.42.3.11	sendTo	109
6.42.3.12	recv	109
6.42.3.13	recvFrom	109
6.42.3.14	recvWithTimeout	109
6.42.3.15	recvFromWithTimeout	109
6.42.3.16	setSockOpt	109
6.42.3.17	getSockOpt	109
6.42.3.18	setReuseAddress	109
6.42.3.19	setBroadCast	109
6.42.3.20	setPriority	109
6.42.3.21	getMTU	109
6.42.3.22	getAddress	109
6.42.4	Member Data Documentation	109
6.42.4.1	osocket	109
6.43	BSocketAddress Class Reference	111
6.43.1	Detailed Description	111
6.43.2	Member Typedef Documentation	112
6.43.2.1	SockAddr	112
6.43.3	Constructor & Destructor Documentation	112
6.43.3.1	BSocketAddress	112
6.43.3.2	BSocketAddress	112
6.43.3.3	BSocketAddress	112
6.43.3.4	~BSocketAddress	112
6.43.4	Member Function Documentation	112
6.43.4.1	set	112
6.43.4.2	raw	112
6.43.4.3	len	112

6.43.4.4	operator=	112
6.43.4.5	operator const SockAddr *	112
6.43.4.6	operator==	112
6.43.4.7	operator"!=	112
6.43.5	Member Data Documentation	112
6.43.5.1	olen	112
6.43.5.2	oaddress	112
6.44	BSocketAddressINET Class Reference	113
6.44.1	Detailed Description	114
6.44.2	Member Typedef Documentation	114
6.44.2.1	SockAddrIP	114
6.44.3	Member Function Documentation	114
6.44.3.1	set	114
6.44.3.2	set	114
6.44.3.3	set	114
6.44.3.4	setPort	114
6.44.3.5	address	114
6.44.3.6	port	114
6.44.3.7	getString	114
6.44.3.8	getHostName	114
6.44.3.9	getIpAddresses	114
6.44.3.10	getIpAddressList	114
6.44.3.11	getIpAddressListAll	114
6.45	BString Class Reference	116
6.45.1	Constructor & Destructor Documentation	119
6.45.1.1	BString	119
6.45.1.2	BString	119
6.45.1.3	BString	119
6.45.1.4	BString	119
6.45.1.5	BString	119
6.45.1.6	BString	119
6.45.1.7	BString	119
6.45.1.8	BString	119
6.45.1.9	BString	119
6.45.1.10	~BString	119
6.45.2	Member Function Documentation	119

6.45.2.1	convert	119
6.45.2.2	convert	119
6.45.2.3	convert	119
6.45.2.4	convert	119
6.45.2.5	convert	120
6.45.2.6	convert	120
6.45.2.7	convertHex	120
6.45.2.8	convertHex	120
6.45.2.9	copy	120
6.45.2.10	strChanged	120
6.45.2.11	len	120
6.45.2.12	retStr	120
6.45.2.13	retStrDup	120
6.45.2.14	retInt	120
6.45.2.15	retDouble	120
6.45.2.16	compare	120
6.45.2.17	compareWild	121
6.45.2.18	compareWildExpression	121
6.45.2.19	add	121
6.45.2.20	truncate	121
6.45.2.21	pad	121
6.45.2.22	toUpper	121
6.45.2.23	toLower	121
6.45.2.24	removeNL	121
6.45.2.25	subString	121
6.45.2.26	del	121
6.45.2.27	insert	121
6.45.2.28	printf	122
6.45.2.29	find	122
6.45.2.30	findReverse	122
6.45.2.31	getTokenList	122
6.45.2.32	removeSeparators	122
6.45.2.33	pullToken	122
6.45.2.34	pullSeparators	122
6.45.2.35	pullWord	122
6.45.2.36	pullLine	122

6.45.2.37 field	122
6.45.2.38 fields	122
6.45.2.39 operator=	122
6.45.2.40 operator[.	122
6.45.2.41 operator==	123
6.45.2.42 operator==	123
6.45.2.43 operator>	123
6.45.2.44 operator>	123
6.45.2.45 operator<	123
6.45.2.46 operator<	123
6.45.2.47 operator>=	123
6.45.2.48 operator<=	123
6.45.2.49 operator"!=	123
6.45.2.50 operator"!=	123
6.45.2.51 operator+	123
6.45.2.52 operator+	123
6.45.2.53 operator+=	123
6.45.2.54 operator+=	123
6.45.2.55 operator+	123
6.45.2.56 operator+	123
6.45.2.57 operator+	123
6.45.2.58 operator+	123
6.45.2.59 operator const char *	123
6.45.2.60 Init	123
6.45.2.61 inString	123
6.45.2.62 isSpace	123
6.45.3 Member Data Documentation	123
6.45.3.1 ostr	123
6.46 BThread Class Reference	124
6.46.1 Constructor & Destructor Documentation	125
6.46.1.1 BThread	125
6.46.1.2 ~BThread	125
6.46.2 Member Function Documentation	125
6.46.2.1 setInitPriority	125
6.46.2.2 setInitStackSize	125
6.46.2.3 start	125

6.46.2.4	result	125
6.46.2.5	running	125
6.46.2.6	setPriority	125
6.46.2.7	cancel	125
6.46.2.8	waitForCompletion	125
6.46.2.9	getThread	125
6.46.2.10	function	125
6.46.2.11	startFunc	125
6.46.3	Member Data Documentation	125
6.46.3.1	othread	125
6.46.3.2	ostackSize	125
6.46.3.3	opolicy	125
6.46.3.4	opriority	125
6.46.3.5	orunning	125
6.46.3.6	oresult	125
6.47	BTimer Class Reference	126
6.47.1	Detailed Description	126
6.47.2	Constructor & Destructor Documentation	127
6.47.2.1	BTimer	127
6.47.2.2	~BTimer	127
6.47.3	Member Function Documentation	127
6.47.3.1	start	127
6.47.3.2	stop	127
6.47.3.3	clear	127
6.47.3.4	getElapsedTime	127
6.47.3.5	add	127
6.47.3.6	average	127
6.47.3.7	peak	127
6.47.3.8	getTime	128
6.47.4	Member Data Documentation	128
6.47.4.1	olock	128
6.47.4.2	onum	128
6.47.4.3	ostartTime	128
6.47.4.4	oendTime	128
6.47.4.5	oaverage	128
6.47.4.6	opeak	128

6.48	BUrl Class Reference	129
6.48.1	Detailed Description	129
6.48.2	Constructor & Destructor Documentation	129
6.48.2.1	BUrl	129
6.48.2.2	~BUrl	129
6.48.3	Member Function Documentation	129
6.48.3.1	readString	129
6.48.3.2	writeData	130
6.48.4	Member Data Documentation	130
6.48.4.1	oinit	130
6.48.4.2	ores	130
7	File Documentation	131
7.1	BArray.h File Reference	131
7.1.1	Define Documentation	131
7.1.1.1	BArray_H	131
7.2	BBuffer.cpp File Reference	132
7.2.1	Define Documentation	132
7.2.1.1	SIZE	132
7.3	BBuffer.h File Reference	133
7.3.1	Define Documentation	133
7.3.1.1	BBUFFER_H	133
7.4	BCond.cpp File Reference	134
7.5	BCond.h File Reference	135
7.5.1	Define Documentation	135
7.5.1.1	BCOND_H	135
7.6	BCondInt.cpp File Reference	136
7.7	BCondInt.h File Reference	137
7.7.1	Define Documentation	137
7.7.1.1	BCONDINT_H	137
7.8	BDir.cpp File Reference	138
7.8.1	Function Documentation	138
7.8.1.1	wild	138
7.8.2	Variable Documentation	138
7.8.2.1	wildString	138
7.9	BDir.h File Reference	139
7.9.1	Define Documentation	139

7.9.1.1	BDIR_H	139
7.10	BEntry.cpp File Reference	140
7.11	BEntry.h File Reference	141
7.12	BError.cpp File Reference	142
7.13	BError.h File Reference	143
7.13.1	Define Documentation	143
7.13.1.1	BERROR_H	143
7.14	BEvent.cpp File Reference	144
7.15	BEvent.h File Reference	145
7.15.1	Define Documentation	145
7.15.1.1	BEvent_H	145
7.15.2	Enumeration Type Documentation	145
7.15.2.1	BEventType	145
7.16	BFile.cpp File Reference	146
7.16.1	Define Documentation	146
7.16.1.1	STRBUF	146
7.17	BFile.h File Reference	147
7.17.1	Define Documentation	147
7.17.1.1	BFILE_H	147
7.18	BList.h File Reference	148
7.18.1	Define Documentation	148
7.18.1.1	BLIST_H	148
7.19	BList_func.h File Reference	149
7.20	BMutex.cpp File Reference	150
7.20.1	Define Documentation	150
7.20.1.1	MDEBUG	150
7.21	BMutex.h File Reference	151
7.21.1	Define Documentation	151
7.21.1.1	BMUTEX_H	151
7.22	BNameValue.h File Reference	152
7.22.1	Define Documentation	152
7.22.1.1	BNAMEVALUE_H	152
7.22.1.2	TEMPLATE_NEW	152
7.23	Boap.cpp File Reference	153
7.23.1	Define Documentation	154
7.23.1.1	APIVERSION_TEST	154

7.23.1.2	DEBUG	154
7.23.1.3	dprintf	154
7.23.1.4	IS_BIG_ENDIAN	154
7.23.2	Function Documentation	154
7.23.2.1	swap16	154
7.23.2.2	swap32	154
7.23.2.3	swap64	154
7.23.2.4	swap8	154
7.23.3	Variable Documentation	154
7.23.3.1	boapPort	154
7.23.3.2	roundSize	154
7.24	Boap.h File Reference	155
7.24.1	Typedef Documentation	156
7.24.1.1	BoapFunc	156
7.24.1.2	BoapService	156
7.24.2	Enumeration Type Documentation	156
7.24.2.1	BoapPriority	156
7.24.2.2	BoapType	156
7.24.3	Variable Documentation	156
7.24.3.1	BoapMagic	156
7.25	BoapnsC.cc File Reference	157
7.26	BoapnsC.h File Reference	158
7.26.1	Define Documentation	158
7.26.1.1	BOAPNSC_H	158
7.27	BoapnsD.cc File Reference	159
7.28	BoapnsD.h File Reference	160
7.28.1	Define Documentation	160
7.28.1.1	BOAPNSD_H	160
7.29	BoapSimple.cc File Reference	161
7.29.1	Define Documentation	161
7.29.1.1	DEBUG	161
7.29.1.2	dprintf	161
7.29.2	Variable Documentation	161
7.29.2.1	roundSize	161
7.30	BoapSimple.h File Reference	162
7.30.1	Typedef Documentation	163

7.30.1.1	BoapFunc	163
7.30.1.2	BoapService	163
7.30.1.3	Double	163
7.30.1.4	Int16	163
7.30.1.5	Int32	163
7.30.1.6	Int8	163
7.30.1.7	UInt16	163
7.30.1.8	UInt32	163
7.30.1.9	UInt8	163
7.30.2	Enumeration Type Documentation	163
7.30.2.1	BoapType	163
7.31	BObject.cc File Reference	164
7.31.1	Define Documentation	164
7.31.1.1	DEBUG	164
7.32	BObject.h File Reference	165
7.32.1	Define Documentation	165
7.32.1.1	BOBJECT_H	165
7.32.2	Typedef Documentation	165
7.32.2.1	BMember	165
7.32.2.2	BMemberList	165
7.33	BPoll-1.cpp File Reference	166
7.34	BPoll.cpp File Reference	167
7.35	BPoll.h File Reference	168
7.35.1	Define Documentation	168
7.35.1.1	B POLL_H	168
7.36	BRefData.cpp File Reference	169
7.36.1	Define Documentation	169
7.36.1.1	CHUNK	169
7.36.1.2	DEBUG	169
7.37	BRefData.h File Reference	170
7.37.1	Define Documentation	170
7.37.1.1	BREFDATA_H	170
7.38	BRtc.cpp File Reference	171
7.39	BRtc.h File Reference	172
7.40	BRWLock.cpp File Reference	173
7.41	BRWLock.h File Reference	174

7.41.1	Define Documentation	174
7.41.1.1	BRWLOCK_H	174
7.42	BSema.cpp File Reference	175
7.43	BSema.h File Reference	176
7.43.1	Define Documentation	176
7.43.1.1	BSEMA_H	176
7.44	BSocket.cpp File Reference	177
7.44.1	Define Documentation	177
7.44.1.1	IP_MTU	177
7.45	BSocket.h File Reference	178
7.45.1	Define Documentation	178
7.45.1.1	BSOCKET_H	178
7.46	BString.cpp File Reference	179
7.46.1	Define Documentation	179
7.46.1.1	DEBUG	179
7.46.1.2	MINUS	179
7.46.1.3	STRIP	179
7.46.2	Function Documentation	179
7.46.2.1	gmatch	179
7.46.2.2	operator<<	179
7.46.2.3	operator>>	179
7.47	BString.h File Reference	180
7.47.1	Define Documentation	180
7.47.1.1	BSTRING_H	180
7.47.2	Function Documentation	180
7.47.2.1	operator<<	180
7.47.2.2	operator>>	180
7.48	BThread.cpp File Reference	181
7.49	BThread.h File Reference	182
7.49.1	Define Documentation	182
7.49.1.1	BTHREAD_H	182
7.50	BTimer.cpp File Reference	183
7.51	BTimer.h File Reference	184
7.52	BTypes.h File Reference	185
7.52.1	Define Documentation	187
7.52.1.1	BTYPES_H	187

7.52.2	Typedef Documentation	187
7.52.2.1	BArrayDouble	187
7.52.2.2	BArrayFloat	187
7.52.2.3	BDouble	187
7.52.2.4	BFloat	187
7.52.2.5	BInt16	187
7.52.2.6	BInt32	187
7.52.2.7	BInt64	187
7.52.2.8	BInt8	187
7.52.2.9	BSize	187
7.52.2.10	BUInt	187
7.52.2.11	BUInt16	187
7.52.2.12	BUInt32	187
7.52.2.13	BUInt64	187
7.52.2.14	BUInt8	187
7.52.2.15	Double	187
7.52.2.16	Float	187
7.52.2.17	Int16	187
7.52.2.18	Int32	187
7.52.2.19	Int64	187
7.52.2.20	Int8	187
7.52.2.21	UInt16	187
7.52.2.22	UInt32	187
7.52.2.23	UInt64	187
7.52.2.24	UInt8	187
7.53	BUrl.cpp File Reference	188
7.54	BUrl.h File Reference	189
7.54.1	Define Documentation	189
7.54.1.1	BURL_H	189

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
BBuffer	12
BCond	14
BCondBool	15
BCondInt	17
BCondValue	20
BCondWrap	23
BEntry	29
BError	37
BEventError	42
BEvent	40
BEventError	42
BEventInt	43
BEventPipe	45
BFile	47
BIter	51
BList< T >	52
BDir	26
BList< T >::Node	59
BList< BEntry >	52
BEntryList	34
BEntryFile	32
BList< BNameValue< T > >	52
BNameValueList< T >	63
BList< dirent * >	52
BMutex	60
BNameValue< T >	62
Boapns::BoapEntry	68
BoapFuncEntry	69
BoapPacket	71
BoapPacketHead	76
BoapServiceEntry	84

BoapServiceObject	85
BObject	91
BPoll	93
BRefData	96
BRtc	98
BRWLock	102
BSema	104
BSocket	106
BoapClientObject	64
Boapns::Boapns	70
BoapClientObject	64
BoapSignalObject	89
BoapSignalObject	89
BSocketAddress	111
BSocketAddressINET	113
BString	116
BThread	124
BoapServer	77
BoapServerConnection	82
BRtcThreaded	100
BTimer	126
BUrl	129

Chapter 3

Class Index

3.1 Class List

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

BArray< T >	11
BBuffer	12
BCond	14
BCondBool (Thread conditional boolean)	15
BCondInt (Thread conditional integer)	17
BCondValue (Thread conditional value)	20
BCondWrap	23
BDir (File system directory class)	26
BEntry (Manipulate a name value pair)	29
BEntryFile (File of Entries)	32
BEntryList (List of Entries. Where an entry is a name value pair)	34
BError (Error return class)	37
BEvent (This class provides a base class for all event objects that can be sent over the events interface)	40
BEventError	42
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)	43
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)	45
BFile (File operations class)	47
BIter (Iterator for BList)	51
BList< T > (Template based list class)	52
BList< T >::Node	59
BMutex (Mutex class)	60
BNameValue< T >	62
BNameValueList< T >	63
BoapClientObject	64
Boapns::BoapEntry	68
BoapFuncEntry	69
Boapns::Boapns	70
BoapPacket	71
BoapPacketHead	76
BoapServer	77

BoapServerConnection	82
BoapServiceEntry	84
BoapServiceObject	85
BoapSignalObject	89
BObject	91
BPoll (This class provides an interface for polling a number of file descriptors. It uses round robin polling)	93
BRefData (Referenced data storage)	96
BRtc (Realtime clock)	98
BRtcThreaded (Threaded real time clock)	100
BRWLock (Thread read-write locks)	102
BSema (Sempahore class)	104
BSocket	106
BSocketAddress (Socket Address)	111
BSocketAddressINET (IP aware socket address)	113
BString	116
BThread	124
BTimer (Stopwatch style timer)	126
BUrl (Basic access to a Url)	129

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

BArray.h	131
BBuffer.cpp	132
BBuffer.h	133
BCond.cpp	134
BCond.h	135
BCondInt.cpp	136
BCondInt.h	137
BDir.cpp	138
BDir.h	139
BEntry.cpp	140
BEntry.h	141
BError.cpp	142
BError.h	143
BEvent.cpp	144
BEvent.h	145
BFile.cpp	146
BFile.h	147
BList.h	148
BList_func.h	149
BMutex.cpp	150
BMutex.h	151
BNameValue.h	152
Boap.cpp	153
Boap.h	155
BoapnsC.cc	157
BoapnsC.h	158
BoapnsD.cc	159
BoapnsD.h	160
BoapSimple.cc	161
BoapSimple.h	162
BObject.cc	164
BObject.h	165
BPoll-1.cpp	166

BPoll.cpp	167
BPoll.h	168
BRefData.cpp	169
BRefData.h	170
BRtc.cpp	171
BRtc.h	172
BRWLock.cpp	173
BRWLock.h	174
BSema.cpp	175
BSema.h	176
BSocket.cpp	177
BSocket.h	178
BString.cpp	179
BString.h	180
BThread.cpp	181
BThread.h	182
BTimer.cpp	183
BTimer.h	184
BTypes.h	185
BUrl.cpp	188
BUrl.h	189

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 \(const BArray &array\)](#)

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]`

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

- [BArray.h](#)

6.2 BBuffer Class Reference

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

Public Member Functions

- [BBuffer \(\)](#)
Create and manipulate a data buffer. On creation the buffer size defaults to 1024 bytes.
- [~BBuffer \(\)](#)
- [int setSize \(uint32_t size\)](#)
Sets the bufer size.
- [int setData \(const void *data, uint32_t size\)](#)
Sets buffer data resized to contain the data.
- [int writeData \(uint32_t pos, const void *data, uint32_t size\)](#)
Writes data into buffer from offset pos.
- [void * data \(\)](#)
The data.
- [uint32_t size \(\)](#)
Size of the buffer in bytes.

Private Attributes

- [uint32_t osize](#)
- [uint32_t odatasize](#)
- [void * odata](#)

6.2.1 Constructor & Destructor Documentation

6.2.1.1 BBuffer::BBuffer ()

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

6.2.1.2 BBuffer::~~BBuffer ()

6.2.2 Member Function Documentation

6.2.2.1 int BBuffer::setSize (uint32_t size)

Sets the bufer size.

6.2.2.2 int BBuffer::setData (const void * data, uint32_t size)

Sets buffer data resized to contain the data.

6.2.2.3 `int BBuffer::writeData (uint32_t pos, const void * data, uint32_t size)`

Writes data into buffer from offset pos.

6.2.2.4 `void * BBuffer::data ()`

The data.

6.2.2.5 `uint32_t BBuffer::size ()`

Size of the buffer in bytes.

6.2.3 Member Data Documentation**6.2.3.1** `uint32_t BBuffer::osize` [private]**6.2.3.2** `uint32_t BBuffer::odatasize` [private]**6.2.3.3** `void* BBuffer::odata` [private]

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

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

6.3 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](#)

6.3.1 Constructor & Destructor Documentation

6.3.1.1 BCond::BCond ()

Thread conditional variable.

6.3.1.2 BCond::~~BCond ()

6.3.2 Member Function Documentation

6.3.2.1 int BCond::signal ()

6.3.2.2 int BCond::wait ()

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

6.3.3 Member Data Documentation

6.3.3.1 `pthread_mutex_t` BCond::omutex [private]

6.3.3.2 `pthread_cond_t` BCond::ocond [private]

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

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

6.4 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.

Private Attributes

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

6.4.1 Detailed Description

Thread conditional boolean.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 BCondBool::BCondBool ()

6.4.2.2 BCondBool::~~BCondBool ()

6.4.3 Member Function Documentation

6.4.3.1 int BCondBool::set ()

Set value. Wakes waiting.

6.4.3.2 int BCondBool::clear ()

Clear Value.

6.4.3.3 int BCondBool::value ()

Current value.

6.4.3.4 int BCondBool::wait ()

Wait until value is true.

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

Wait until set, with timeout.

6.4.4 Member Data Documentation**6.4.4.1 pthread_mutex_t BCondBool::omutex [private]****6.4.4.2 pthread_cond_t BCondBool::ocond [private]****6.4.4.3 int BCondBool::ovalue [private]**

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

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

6.5 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](#)

6.5.1 Detailed Description

Thread conditional integer.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 BCondInt::BCondInt ()

6.5.2.2 BCondInt::~~BCondInt ()

6.5.3 Member Function Documentation

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

Set value.

6.5.3.2 int BCondInt::increment ()

Increment.

6.5.3.3 int BCondInt::decrement ()

Decrement.

6.5.3.4 int BCondInt::value ()

Current value.

6.5.3.5 int BCondInt::wait ()

Wait until value is 0.

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

Wait until value is 0 then increment.

6.5.3.7 int BCondInt::waitNotZero ()

Wait until value is not 0.

6.5.3.8 int BCondInt::waitNotZeroDecrement ()

Wait until value is not 0 and then decrement.

6.5.3.9 int BCondInt::tryNotZeroDecrement ()

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

6.5.3.10 int BCondInt::timedWait (int *timeOutUs*)

Wait for the condition, with timeout.

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

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

6.5.4 Member Data Documentation

6.5.4.1 pthread_mutex_t BCondInt::omutex [private]

6.5.4.2 pthread_cond_t BCondInt::ocond [private]

6.5.4.3 int BCondInt::ovalue [private]

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

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

6.6 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](#)

6.6.1 Detailed Description

Thread conditional value.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 BCondValue::BCondValue ()

6.6.2.2 BCondValue::~~BCondValue ()

6.6.3 Member Function Documentation

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

Set the value. Wakes waiting.

6.6.3.2 int BCondValue::value ()

Current value.

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

Increment. Wakes waiting.

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

Decrement. Wakes waiting.

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

Wait until value is at least the value given.

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

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

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

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

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

Add to value. Wakes waiting.

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

Subtract from value. Wakes waiting.

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

Increment value. Wakes waiting.

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

Decrement value. Wakes waiting.

6.6.4 Member Data Documentation**6.6.4.1 pthread_mutex_t BCondValue::omutex [private]****6.6.4.2 pthread_cond_t BCondValue::ocond [private]****6.6.4.3 int BCondValue::ovalue [private]**

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

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

6.7 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](#)

6.7.1 Constructor & Destructor Documentation

6.7.1.1 BCondWrap::BCondWrap ()

6.7.1.2 BCondWrap::~~BCondWrap ()

6.7.2 Member Function Documentation

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

Set the value. Wakes waiting.

6.7.2.2 uint32_t BCondWrap::value ()

Current value.

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

Increment. Wakes waiting.

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

Decrement. Wakes waiting.

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

Wait until value is at least the value given.

6.7.2.6 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.

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

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

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

Add to value. Wakes waiting.

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

Subtract from value. Wakes waiting.

6.7.2.10 void BCondWrap::operator++ (int) [inline]

Increment value. Wakes waiting.

6.7.2.11 void BCondWrap::operator- (int) [inline]

Decrement value. Wakes waiting.

6.7.2.12 int BCondWrap::diff (uint32_t v) [private]**6.7.3 Member Data Documentation****6.7.3.1 pthread_mutex_t BCondWrap::omutex [private]****6.7.3.2 pthread_cond_t BCondWrap::ocond [private]****6.7.3.3 uint32_t BCondWrap::ovalue [private]**

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

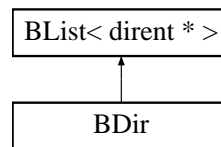
- [BCondInt.h](#)
- [BCondInt.cpp](#)

6.8 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](#)

6.8.1 Detailed Description

File system directory class.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 BDir::BDir ()

6.8.2.2 BDir::BDir (BString *name*)

6.8.2.3 BDir::~~BDir ()

6.8.3 Member Function Documentation

6.8.3.1 BError BDir::open (BString *name*)

Reads named directory.

6.8.3.2 BError BDir::error ()

Current value of error.

6.8.3.3 BError BDir::read ()

read/re-reads directory

6.8.3.4 void BDir::clear () [virtual]

Clears list.

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

6.8.3.5 void BDir::setWild (BString *wild*)

Set wildcard filter string used on read.

6.8.3.6 void BDir::setSort (int *on*)

Set alpha sort on/off.

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

Get filename.

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

Get file stats.

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

Get file stats 64.

6.8.4 Member Data Documentation

6.8.4.1 BError BDir::oerror [private]

6.8.4.2 BString BDir::odirname [private]

6.8.4.3 BString BDir::owild [private]

6.8.4.4 int BDir::osort [private]

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

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

6.9 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](#)

6.9.1 Detailed Description

Manipulate a name value pair.

6.9.2 Constructor & Destructor Documentation

6.9.2.1 BEntry::BEntry ()

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

Set name and value.

6.9.2.3 BEntry::BEntry (BString *line*)

Set name and value from white space delimited string.

6.9.3 Member Function Documentation

6.9.3.1 BString BEntry::getName ()

Get the name.

6.9.3.2 BString BEntry::getValue ()

Get the value.

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

Set name and value from white space delimited string.

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

Set the name.

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

Set the value.

6.9.3.6 BString BEntry::line ()

Return name and value as padded single string.

6.9.3.7 void BEntry::print ()

Print name and value.

6.9.4 Member Data Documentation

6.9.4.1 BString BEntry::oname [private]

6.9.4.2 BString BEntry::ovalue [private]

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

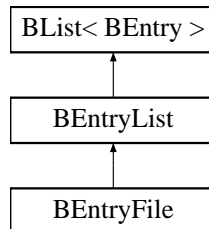
- [BEntry.h](#)
- [BEntry.cpp](#)

6.10 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](#)

6.10.1 Detailed Description

File of Entries.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 BEntryFile::BEntryFile ()

6.10.2.2 BEntryFile::BEntryFile (BString *filename*)

Opens entryfile.

6.10.2.3 BEntryFile::~~BEntryFile ()

6.10.3 Member Function Documentation

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

Opens entryfile.

6.10.3.2 int BEntryFile::read ()

Reads entry file and builds list.

6.10.3.3 int BEntryFile::write ()

Writes list to entryfile.

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

Writes specified list to file.

6.10.3.5 void BEntryFile::clear () [virtual]

Clears current list.

Reimplemented from [BEntryList](#).

6.10.4 Member Data Documentation

6.10.4.1 BString BEntryFile::ofilename [private]

6.10.4.2 BString BEntryFile::ocomments [private]

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

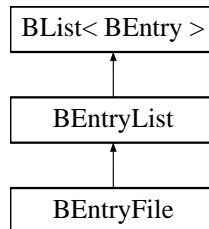
- [BEntry.h](#)
- [BEntry.cpp](#)

6.11 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.

Private Attributes

- [BIter olastPos](#)

6.11.1 Detailed Description

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

6.11.2 Constructor & Destructor Documentation

6.11.2.1 BEntryList::BEntryList ()

6.11.3 Member Function Documentation

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

1 if name is in list and value is set

6.11.3.2 BEntry * BEntryList::find (BString *name*)

Returns entry if name is found otherwise NULL.

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

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

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

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

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

Raw setting of value without looking up existing entry.

6.11.3.6 void BEntryList::deleteEntry (BString *name*)

Deletes the entry.

6.11.3.7 void BEntryList::print ()

Print list.

6.11.3.8 BString BEntryList::getString ()

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

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

Insert item before item.

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

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

Delete specified item.

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

6.11.3.11 void BEntryList::clear () [virtual]

Clear the list.

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

Reimplemented in [BEntryFile](#).

6.11.4 Member Data Documentation**6.11.4.1 BIter BEntryList::olastPos** [private]

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

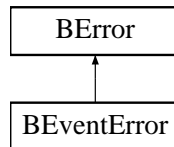
- [BEntry.h](#)
- [BEntry.cpp](#)

6.12 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 & 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`

6.12.1 Detailed Description

Error return class.

6.12.2 Member Enumeration Documentation

6.12.2.1 enum BError::Type

Enumerator:

NONE

ERROR

6.12.3 Constructor & Destructor Documentation

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

Create object.

6.12.3.2 BError::BError (BString *errStr*)

Create with error set and error string.

6.12.4 Member Function Documentation

6.12.4.1 BError BError::copy ()

Return an independant copy.

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

Set error number and message.

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

Set error type ERROR with optional message.

6.12.4.4 BString BError::getString () const

Get error message.

6.12.4.5 int BError::getErrorNo () const

Get The error number.

6.12.4.6 BError::operator int () const

Return error number.

6.12.5 Member Data Documentation

6.12.5.1 int BError::oerrNo [private]

6.12.5.2 BString BError::oerrStr [private]

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

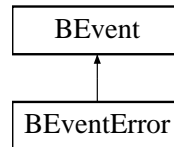
- [BError.h](#)
- [BError.cpp](#)

6.13 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.

6.13.1 Detailed Description

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

6.13.2 Constructor & Destructor Documentation

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

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

6.13.3 Member Function Documentation

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

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

Reimplemented in [BEventError](#).

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

Reimplemented in [BEventError](#).

6.13.4 Member Data Documentation

6.13.4.1 uint32_t BEvent::otype [private]

The event type.

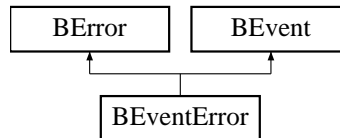
The documentation for this class was generated from the following files:

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

6.14 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)

6.14.1 Constructor & Destructor Documentation

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

6.14.2 Member Function Documentation

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

Reimplemented from [BEvent](#).

6.14.2.2 [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](#)

6.15 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 \(\)](#)
- [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.

6.15.1 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.

6.15.2 Constructor & Destructor Documentation

6.15.2.1 BEventInt::BEventInt ()

6.15.2.2 BEventInt::~~BEventInt ()

6.15.3 Member Function Documentation

6.15.3.1 BError BEventInt::sendEvent (int event)

Send an event.

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

Receive the event.

6.15.3.3 int BEventInt::getFd ()

6.15.4 Member Data Documentation

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

File descriptors for pipe.

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

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

6.16 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 \(\)](#)
- [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.

6.16.1 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.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 BEventPipe::BEventPipe ()

6.16.2.2 BEventPipe::~~BEventPipe ()

6.16.3 Member Function Documentation

6.16.3.1 BError BEventPipe::sendEvent (BEvent * event)

Send an event.

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

Receive the event.

6.16.3.3 int BEventPipe::getReceiveFd ()

returns the receive file descriptor for the poll system call

6.16.4 Member Data Documentation

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

File descriptors for pipe.

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

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

6.17 BFile Class Reference

File operations class.

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

Public Member Functions

- **BFile** ()
 - **BFile** (BString name, BString mode)
Create opened specified file.
- **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 close** ()
Close file.
- **BError error** ()
Returns current error state.
- FILE * **getFd** ()
File descriptor.
- int **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).
- int **write** (const void *buf, int nbytes)
Write to file.
- int **writeString** (const BString &str)
Write string to file.
- int **seek** (int pos, int whence)

Set seek position.

- int `printf` (const char *fmt,...)
Formatted print into the file.
- `BFile & operator=` (const `BFile` &file)

Private Attributes

- FILE * `ofile`
- `BString` `ofilename`
- `BString` `omode`
- `BError` `oerror`

6.17.1 Detailed Description

File operations class.

6.17.2 Constructor & Destructor Documentation

6.17.2.1 `BFile::BFile ()`

6.17.2.2 `BFile::BFile (BString name, BString mode)`

Create opened specified file.

6.17.2.3 `BFile::BFile (const BFile &file)`

Create opened specified file.

6.17.2.4 `BFile::~~BFile ()`

6.17.3 Member Function Documentation

6.17.3.1 `BError BFile::open (BString name, BString mode)`

Open file.

6.17.3.2 `BError BFile::open (FILE *file)`

Assign object to opened file handle.

6.17.3.3 `BError BFile::close ()`

Close file.

6.17.3.4 BError BFile::error ()

Returns current error state.

6.17.3.5 FILE * BFile::getFd ()

File descriptor.

6.17.3.6 int BFile::length ()

File size in bytes.

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

Set stream buffering options.

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

Read from file.

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

Read string. (ref fgets).

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

Write to file.

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

Write string to file.

6.17.3.12 int BFile::seek (int *pos*, int *whence*)

Set seek position.

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

Formatted print into the file.

6.17.3.14 BFile & BFile::operator= (const BFile & *file*)

6.17.4 Member Data Documentation

6.17.4.1 FILE* BFile::ofile [private]

6.17.4.2 BString BFile::ofilename [private]

6.17.4.3 BString BFile::omode [private]

6.17.4.4 BError BFile::oerror [private]

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

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

6.18 BIter Class Reference

Iterator for [BList](#).

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

Public Member Functions

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

Private Attributes

- void * [oi](#)

6.18.1 Detailed Description

Iterator for [BList](#).

6.18.2 Constructor & Destructor Documentation

6.18.2.1 [BIter::BIter](#) (void * *i* = 0) [inline]

6.18.3 Member Function Documentation

6.18.3.1 [BIter::operator void *](#) () [inline]

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

6.18.4 Member Data Documentation

6.18.4.1 void* [BIter::oi](#) [private]

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

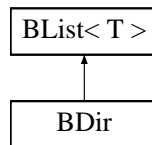
- [BList.h](#)

6.19 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)
Postition in list item with iterator i.
- unsigned int [number](#) ()

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[]` (`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`

6.19.1 Detailed Description

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

Template based list class.

6.19.2 Member Typedef Documentation

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

Prototype for sorting function.

6.19.3 Constructor & Destructor Documentation

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

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

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

6.19.4 Member Function Documentation

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

Iterator to start of list.

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

Iterator for start of list.

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

Iterator for end of list.

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

Iterator for end of list.

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

Iterator for next item in list.

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

Iterator for previous item in list.

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

Iterator for pos item in list.

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

Position in list item with iterator i.

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

Number of items in list.

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

True if iterator refers to last item.

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

Get first item in list.

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

Get last item in list.

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

Get item specified by iterator in list.

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

Get item specified by iterator in list.

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

Append item to list.

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

Insert item before item.

Reimplemented in [BEntryList](#).

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

Insert item after item.

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

Clear the list.

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

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

Delete specified item.

Reimplemented in [BEntryList](#).

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

Delete last item.

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

Delete first item.

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

Push item onto list.

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

Pop item from list deleting item.

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

Add item to end of list.

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

Get item from front of list deleting item.

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

Append list to list.

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

Swap two items in list.

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

Sort list based on get(i) values.

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

Sort list based on Sort func.

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

6.19.4.31]

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

6.19.4.32]

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

6.19.4.33]

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

6.19.4.34]

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

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

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

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

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

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

6.19.5 Member Data Documentation

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

6.19.5.2 `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](#)

6.20 BList< T >::Node Class Reference

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

Public Member Functions

- [Node](#) (const T &i)

Public Attributes

- [Node](#) * [next](#)
- [Node](#) * [prev](#)
- T [item](#)

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

6.20.1 Constructor & Destructor Documentation

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

6.20.2 Member Data Documentation

6.20.2.1 `template<class T> Node* BList< T >::Node::next`

6.20.2.2 `template<class T> Node* BList< T >::Node::prev`

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

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

- [BList.h](#)

6.21 BMutex Class Reference

Mutex class.

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

Public Member Functions

- [BMutex \(\)](#)
- [BMutex \(const \[BMutex\]\(#\) &mutex\)](#)
- [~BMutex \(\)](#)
- [int lock \(\)](#)
Set lock, wait in necessary.
- [int unlock \(\)](#)
Unlock the lock.
- [int tryLock \(\)](#)
Test the lock.
- [BMutex & operator= \(const \[BMutex\]\(#\) &mutex\)](#)

Private Attributes

- pthread_mutex_t [omutex](#)

6.21.1 Detailed Description

Mutex class.

6.21.2 Constructor & Destructor Documentation

6.21.2.1 BMutex::BMutex ()

6.21.2.2 BMutex::BMutex (const BMutex & mutex)

6.21.2.3 BMutex::~~BMutex ()

6.21.3 Member Function Documentation

6.21.3.1 int BMutex::lock ()

Set lock, wait in necessary.

6.21.3.2 int BMutex::unlock ()

Unlock the lock.

6.21.3.3 int BMutex::tryLock ()

Test the lock.

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

6.21.4 Member Data Documentation

6.21.4.1 pthread_mutex_t BMutex::omutex [private]

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

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

6.22 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 >
```

6.22.1 Constructor & Destructor Documentation

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

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

6.22.2 Member Function Documentation

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

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

6.22.3 Member Data Documentation

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

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

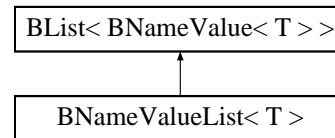
The documentation for this class was generated from the following file:

- [BNameValue.h](#)

6.23 BNameValueList< T > Class Template Reference

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

Inheritance diagram for BNameValueList< T >::



Public Member Functions

- T * [find](#) (BString name)

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

6.23.1 Member Function Documentation

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

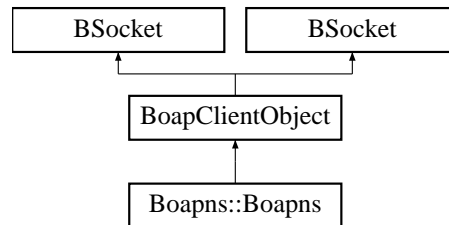
The documentation for this class was generated from the following file:

- [BNameValue.h](#)

6.24 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

6.24.1 Constructor & Destructor Documentation

6.24.1.1 [BoapClientObject::BoapClientObject](#) ([BString](#) name = " ")

6.24.1.2 [BoapClientObject::BoapClientObject](#) ([BString](#) name)

6.24.2 Member Function Documentation

6.24.2.1 [BError BoapClientObject::connectService](#) ([BString](#) name)

Connects to the named service.

6.24.2.2 [BError BoapClientObject::disconnectService](#) ()

Disconnects from the named service.

6.24.2.3 [BString BoapClientObject::getServiceName](#) ()

Get the name of the service.

6.24.2.4 [BError BoapClientObject::ping](#) ([BUInt32](#) & apiVersion)

Pings the connection and finds the remotes version number.

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

Sets the connection priority.

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

Sets the maximum packet length.

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

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

6.24.2.8 BError BoapClientObject::pingLocked (BUInt32 & *apiVersion*) [protected]**6.24.2.9 BError BoapClientObject::checkApiVersion ()** [protected]**6.24.2.10 BError BoapClientObject::performCall (BoapPacket & *tx*, BoapPacket & *rx*)**
[protected]

Performs a RPC call to the named service.

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

Performs a send to the named service.

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

Performs a receive.

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

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

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

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

6.24.3 Member Data Documentation

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

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

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

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

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

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

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

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

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

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

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

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

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

6.25 Boapns::BoapEntry Class Reference

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

Public Member Functions

- [BoapEntry \(\)](#)
- [BoapEntry \(BString pname, BString phostName, BList< BString > paddressList, UInt32 pport, UInt32 pservice\)](#)

Public Attributes

- [BString name](#)
- [BString hostName](#)
- [BList< BString > addressList](#)
- [UInt32 port](#)
- [UInt32 service](#)

6.25.1 Constructor & Destructor Documentation

6.25.1.1 Boapns::BoapEntry::BoapEntry ()

6.25.1.2 Boapns::BoapEntry::BoapEntry (BString *pname*, BString *phostName*, BList< BString > *paddressList*, UInt32 *pport*, UInt32 *pservice*)

6.25.2 Member Data Documentation

6.25.2.1 BString Boapns::BoapEntry::name

6.25.2.2 BString Boapns::BoapEntry::hostName

6.25.2.3 BList<BString> Boapns::BoapEntry::addressList

6.25.2.4 UInt32 Boapns::BoapEntry::port

6.25.2.5 UInt32 Boapns::BoapEntry::service

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

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

6.26 BoapFuncEntry Class Reference

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

Public Member Functions

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

Public Attributes

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

6.26.1 Constructor & Destructor Documentation

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

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

6.26.2 Member Data Documentation

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

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

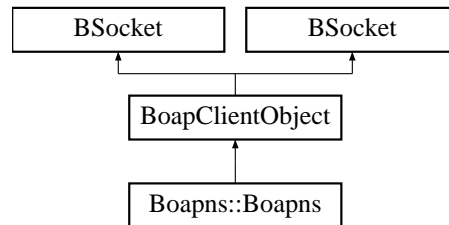
The documentation for this class was generated from the following files:

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

6.27 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 \(BString &name\)](#)

6.27.1 Constructor & Destructor Documentation

6.27.1.1 [Boapns::Boapns::Boapns \(BString *name* = " "\)](#)

6.27.2 Member Function Documentation

6.27.2.1 [BError Boapns::Boapns::getVersion \(BString & *version*\)](#)

6.27.2.2 [BError Boapns::Boapns::getEntryList \(BList< BoapEntry > & *entryList*\)](#)

6.27.2.3 [BError Boapns::Boapns::getEntry \(BString *name*, BoapEntry & *entry*\)](#)

6.27.2.4 [BError Boapns::Boapns::addEntry \(BoapEntry *entry*\)](#)

6.27.2.5 [BError Boapns::Boapns::delEntry \(BString *name*\)](#)

6.27.2.6 [BError Boapns::Boapns::getNewName \(BString & *name*\)](#)

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

- [BoapnsC.h](#)

6.28 BoapPacket Class Reference

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

Public Member Functions

- [BoapPacket](#) ()
- [~BoapPacket](#) ()
- [int](#) [resize](#) (int size)
- [BError](#) [setData](#) (void *data, int nbytes)
- [int](#) [nbytes](#) ()
- [char *](#) [data](#) ()
- [int](#) [peekHead](#) ([BoapPacketHead](#) &head)
- [UInt32](#) [getCmd](#) ()
- [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](#) ([Int64](#) v)
- [int](#) [push](#) ([UInt64](#) v)
- [int](#) [push](#) (const [BString](#) &v)
- [int](#) [push](#) ([Double](#) v)
- [int](#) [push](#) (const [BError](#) &v)
- [int](#) [push](#) ([UInt32](#) nBytes, const void *data, char *swapType="1")
- [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](#) ([Int64](#) &v)
- [int](#) [pop](#) ([UInt64](#) &v)
- [int](#) [pop](#) ([BString](#) &v)
- [int](#) [pop](#) ([Double](#) &v)
- [int](#) [pop](#) ([BError](#) &v)
- [int](#) [pop](#) ([UInt32](#) nBytes, void *data, char *swapType="1")
- [BoapPacket](#) ()
- [~BoapPacket](#) ()
- [int](#) [resize](#) (int size)
- [BError](#) [setData](#) (void *data, int nbytes)
- [int](#) [nbytes](#) ()
- [char *](#) [data](#) ()
- [int](#) [pushHead](#) ([BoapPacketHead](#) &head)
- [int](#) [push](#) ([Int8](#) v)
- [int](#) [push](#) ([UInt8](#) v)

- int [push](#) ([Int16](#) v)
- int [push](#) ([UInt16](#) v)
- int [push](#) ([Int32](#) v)
- int [push](#) ([UInt32](#) v)
- int [push](#) ([BString](#) &v)
- int [push](#) ([Double](#) v)
- int [push](#) ([BError](#) &v)
- int [push](#) ([UInt32](#) nBytes, const void *data)
- int [popHead](#) ([BoapPacketHead](#) &head)
- int [pop](#) ([Int8](#) &v)
- int [pop](#) ([UInt8](#) &v)
- int [pop](#) ([Int16](#) &v)
- int [pop](#) ([UInt16](#) &v)
- int [pop](#) ([Int32](#) &v)
- int [pop](#) ([UInt32](#) &v)
- int [pop](#) ([BString](#) &v)
- int [pop](#) ([Double](#) &v)
- int [pop](#) ([BError](#) &v)
- int [pop](#) ([UInt32](#) nBytes, void *data)

Private Member Functions

- void [copyWithSwap](#) (void *dst, const void *src, [UInt32](#) nBytes, char *swapType)
- void [updateLen](#) ()
- void [updateLen](#) ()

Private Attributes

- int [osize](#)
- int [onbytes](#)
- char * [odata](#)
- int [opos](#)

6.28.1 Constructor & Destructor Documentation

6.28.1.1 `BoapPacket::BoapPacket ()`

6.28.1.2 `BoapPacket::~~BoapPacket ()`

6.28.1.3 `BoapPacket::BoapPacket ()`

6.28.1.4 `BoapPacket::~~BoapPacket ()`

6.28.2 Member Function Documentation

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

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

6.28.2.3 `int BoapPacket::nbytes ()`

6.28.2.4 `char * BoapPacket::data ()`

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

6.28.2.6 `UInt32 BoapPacket::getCmd ()`

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

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

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

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

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

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

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

6.28.2.14 `int BoapPacket::push (Int64 v)`

6.28.2.15 `int BoapPacket::push (UInt64 v)`

6.28.2.16 `int BoapPacket::push (const BString & v)`

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

6.28.2.18 `int BoapPacket::push (const BError & v)`

6.28.2.19 `int BoapPacket::push (UInt32 nBytes, const void * data, char * swapType = "1")`

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

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

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

Generated on Thu Dec 18 13:21:31 2008 for LibBeamApi by Doxygen

6.28.2.23 `int BoapPacket::pop (Int16 & v)`

6.28.2.24 `int BoapPacket::pop (UInt16 & v)`

6.28.2.25 `int BoapPacket::pop (Int32 & v)`

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

6.29 BoapPacketHead Struct Reference

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

Public Attributes

- [UInt32 type](#)
- [UInt32 length](#)
- [UInt32 service](#)
- [UInt32 cmd](#)
- [BoapType type](#)
- [BoapService service](#)
- [UInt32 reserved](#) [12]

6.29.1 Member Data Documentation

6.29.1.1 [UInt32 BoapPacketHead::type](#)

6.29.1.2 [UInt32 BoapPacketHead::length](#)

6.29.1.3 [UInt32 BoapPacketHead::service](#)

6.29.1.4 [UInt32 BoapPacketHead::cmd](#)

6.29.1.5 [BoapType BoapPacketHead::type](#)

6.29.1.6 [BoapService BoapPacketHead::service](#)

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

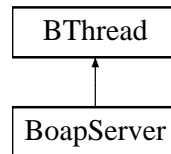
The documentation for this struct was generated from the following files:

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

6.30 BoapServer Class Reference

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

Inheritance diagram for BoapServer::



Public Types

- enum { **NOTHEADS** = 0, **THREADED** = 1 }

Public Member Functions

- **BoapServer** ()
- **~BoapServer** ()
- **BError** **init** (**BString** boapNsHost="", 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** ()
- **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** ()

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`

6.30.1 Member Enumeration Documentation

6.30.1.1 anonymous enum

Enumerator:

NOTHREADS

THREADED

6.30.2 Constructor & Destructor Documentation

6.30.2.1 **BoapServer::BoapServer ()**

6.30.2.2 **BoapServer::~~BoapServer ()**

6.30.2.3 **BoapServer::BoapServer ()**

6.30.3 Member Function Documentation

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

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

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

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

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

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

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

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

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

6.30.3.10 **BString BoapServer::getHostName ()**

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

6.30.3.12 **int BoapServer::getConnectionsNumber ()**

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

Reimplemented from [BThread](#).

- 6.30.3.14 **BError** BoapServer::init (int *boapNs* = 0)
- 6.30.3.15 **BError** BoapServer::run ()
- 6.30.3.16 **BError** BoapServer::processEvent (BoapPacket & *rx*)
- 6.30.3.17 **BError** BoapServer::addObject (BoapServiceObject * *object*)
- 6.30.3.18 **BError** BoapServer::process (int *fd*)
- 6.30.3.19 **BError** BoapServer::sendEvent (BoapPacket & *tx*)
- 6.30.3.20 **BSocket&** BoapServer::getSocket ()
- 6.30.3.21 **BSocket&** BoapServer::getEventSocket ()
- 6.30.3.22 **BError** BoapServer::processEvent (int *fd*)
- 6.30.3.23 **BString** BoapServer::getHostName ()

6.30.4 Member Data Documentation

- 6.30.4.1 **int** BoapServer::othreaded [private]
- 6.30.4.2 **int** BoapServer::oisBoapns [private]
- 6.30.4.3 **Boapns::Boapns*** BoapServer::oboapns [private]
- 6.30.4.4 **BList<BoapServerConnection*>** BoapServer::oclients [private]
- 6.30.4.5 **BEventInt** BoapServer::oclientGoneEvent [private]
- 6.30.4.6 **BList< BoapServiceEntry >** BoapServer::oservices [private]
- 6.30.4.7 **BPoll** BoapServer::opoll [private]
- 6.30.4.8 **BSocket** BoapServer::onet [private]
- 6.30.4.9 **BSocket** BoapServer::onetEvent [private]
- 6.30.4.10 **BSocketAddressINET** BoapServer::onetEventAddress [private]
- 6.30.4.11 **BString** BoapServer::ohostName [private]
- 6.30.4.12 **int** BoapServer::oboapNs [private]
- 6.30.4.13 **BoapPacket** BoapServer::orx [private]
- 6.30.4.14 **BoapPacket** BoapServer::otx [private]

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

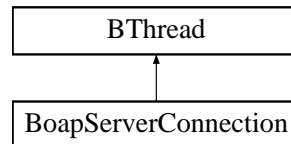
- [Boap.h](#)

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

6.31 BoapServerConnection Class Reference

```
#include <Boap.h>
```

Inheritance diagram for BoapServerConnection::



Public Member Functions

- [BoapServerConnection](#) ([BoapServer](#) &boapServer, int fd)
- [BError](#) process ()
- [BSocket](#) & [getSocket](#) ()
- void [setMaxLength](#) ([BUInt32](#) maxLength)

Private Member Functions

- void * [function](#) ()

Private Attributes

- [BoapServer](#) & oboapServer
- [BSocket](#) osocket
- [BoapPacket](#) orx
- [BoapPacket](#) otx
- [BUInt32](#) omaxLength

6.31.1 Constructor & Destructor Documentation

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

6.31.2 Member Function Documentation

6.31.2.1 [BError](#) [BoapServerConnection::process](#) ()

6.31.2.2 [BSocket](#) & [BoapServerConnection::getSocket](#) ()

6.31.2.3 void [BoapServerConnection::setMaxLength](#) ([BUInt32](#) *maxLength*)

6.31.2.4 void * [BoapServerConnection::function](#) () [private, virtual]

Reimplemented from [BThread](#).

6.31.3 Member Data Documentation

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

6.31.3.2 `BSocket BoapServerConnection::osocket` [private]

6.31.3.3 `BoapPacket BoapServerConnection::orx` [private]

6.31.3.4 `BoapPacket BoapServerConnection::otx` [private]

6.31.3.5 `BUInt32 BoapServerConnection::omaxLength` [private]

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

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

6.32 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

6.32.1 Constructor & Destructor Documentation

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

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

6.32.2 Member Data Documentation

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

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

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

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

6.33 BoapServiceObject Class Reference

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

Public Member Functions

- [BoapServiceObject](#) ([BoapServer](#) &server, [BString](#) name="")
- virtual [~BoapServiceObject](#) ()
- [BError](#) setName ([BString](#) name)
- [BError](#) sendEvent ([BString](#) signalName, [Int32](#) arg)
- virtual [BError](#) processEvent ([BString](#) objectName, [BString](#) name, [Int32](#) 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

6.33.1 Constructor & Destructor Documentation

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

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

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

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

6.33.2 Member Function Documentation

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

6.33.2.2 **BError BoapServiceObject::sendEvent** (BString *signalName*, Int32 *arg*)

6.33.2.3 **BError BoapServiceObject::processEvent** (BString *objectName*, BString *name*, Int32 *arg*) [virtual]

6.33.2.4 **BString BoapServiceObject::name** ()

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

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

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

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

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

6.33.2.10 **BError BoapServiceObject::sendEvent** (BString *signalName*, Int32 *arg*)

6.33.2.11 **virtual BError BoapServiceObject::processEvent** (BString *objectName*, BString *name*, Int32 *arg*) [virtual]

6.33.2.12 **BString BoapServiceObject::name** ()

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

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

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

6.33.3 Member Data Documentation

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

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

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

Generated on Thu Dec 18 13:21:31 2008 for LibBeamApi by Doxygen

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

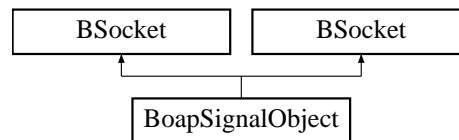
The documentation for this class was generated from the following files:

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

6.34 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](#)

6.34.1 Constructor & Destructor Documentation

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

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

6.34.2 Member Function Documentation

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

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

6.34.3 Member Data Documentation

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

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

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

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

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

6.35 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))

6.35.1 Constructor & Destructor Documentation

6.35.1.1 `BObject::BObject ()`

6.35.1.2 `BObject::~~BObject ()` [virtual]

6.35.2 Member Function Documentation

6.35.2.1 `BError BObject::getBinary (BDataBuf & buf)` [virtual]

6.35.2.2 `BError BObject::setBinary (BDataBuf & buf)` [virtual]

6.35.2.3 `BString BObject::getString ()` [virtual]

6.35.2.4 `BError BObject::setString (BString str)` [virtual]

6.35.2.5 `BMemberList BObject::getMemberList ()` [virtual]

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

6.35.2.7 `void BObject::printIt ()` [virtual]

6.35.2.8 `BType & BObject::getType ()` [virtual]

6.35.2.9 `BObject * BObject::createObj ()` [static]

6.35.3 Member Data Documentation

6.35.3.1 `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](#)

6.36 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.

6.36.1 Detailed Description

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

6.36.2 Member Typedef Documentation

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

6.36.3 Constructor & Destructor Documentation

6.36.3.1 `BPoll::BPoll ()`

6.36.3.2 `BPoll::~~BPoll ()`

6.36.4 Member Function Documentation

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

Append a file descriptor to polling list.

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

Remove a file descriptor from polling list.

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

Perform polling operation.

6.36.4.4 `int BPoll::getPollFdsNum ()`

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

6.36.4.6 `void BPoll::clear ()`

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

6.36.5 Member Data Documentation

6.36.5.1 `int BPoll::ofdsNum` `[private]`

The number of FD's in list.

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

The list of poll fd's.

6.36.5.3 `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-1.cpp](#)
 - [BPoll.cpp](#)

6.37 BRefData Class Reference

Referenced data storage.

```
#include <BRefData.h>
```

Public Member Functions

- [BRefData](#) ()
- [BRefData](#) (int len)
- [BRefData](#) (const [BRefData](#) &refData)
- [~BRefData](#) ()
- [BRefData](#) * [copy](#) ()
- [BRefData](#) * [addRef](#) ()
- int [deleteRef](#) ()
- int [refCount](#) ()
- char * [data](#) ()
- int [len](#) ()
- void [setLen](#) (int len)
- [BRefData](#) & [operator=](#) ([BRefData](#) &refData)

Private Attributes

- void * [oData](#)
- int [oLen](#)
- int [oSize](#)
- int [oRefCount](#)

6.37.1 Detailed Description

Referenced data storage.

6.37.2 Constructor & Destructor Documentation

6.37.2.1 `BRefData::BRefData ()`

6.37.2.2 `BRefData::BRefData (int len)`

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

6.37.2.4 `BRefData::~~BRefData ()`

6.37.3 Member Function Documentation

6.37.3.1 `BRefData * BRefData::copy ()`

6.37.3.2 `BRefData * BRefData::addRef ()`

6.37.3.3 `int BRefData::deleteRef ()`

6.37.3.4 `int BRefData::refCount ()` `[inline]`

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

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

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

6.37.3.8 `BRefData & BRefData::operator= (BRefData & refData)`

6.37.4 Member Data Documentation

6.37.4.1 `void* BRefData::oData` `[private]`

6.37.4.2 `int BRefData::oLen` `[private]`

6.37.4.3 `int BRefData::oSize` `[private]`

6.37.4.4 `int BRefData::oRefCount` `[private]`

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

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

6.38 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](#)

6.38.1 Detailed Description

Realtime clock.

6.38.2 Constructor & Destructor Documentation

6.38.2.1 BRtc::BRtc ()

6.38.2.2 BRtc::~~BRtc ()

6.38.3 Member Function Documentation

6.38.3.1 BError BRtc::init (int rate)

Setup interrupt rate.

6.38.3.2 void BRtc::wait (int delayUs)

Wait specified uS.

6.38.4 Member Data Documentation

6.38.4.1 int BRtc::ofd [private]

6.38.4.2 int BRtc::orate [private]

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

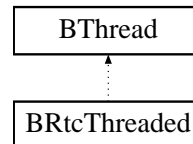
-
- [BRtc.h](#)
 - [BRtc.cpp](#)

6.39 BRtcThreaded Class Reference

Threaded real time clock.

```
#include <BRtc.h>
```

Inheritance diagram for BRtcThreaded::



Public Member Functions

- [BRtcThreaded \(\)](#)
- [~BRtcThreaded \(\)](#)
- [BError init](#) (int rate)
Setup interrupt rate.
- void [wait](#) (int delayUs)
Wait specified uS.

Private Member Functions

- void * [function](#) ()

Private Attributes

- [BRtc orte](#)
- int [orate](#)
- [BCond ocond](#)

6.39.1 Detailed Description

Threaded real time clock.

6.39.2 Constructor & Destructor Documentation

6.39.2.1 BRtcThreaded::BRtcThreaded ()

6.39.2.2 BRtcThreaded::~~BRtcThreaded ()

6.39.3 Member Function Documentation

6.39.3.1 BError BRtcThreaded::init (int rate)

Setup interrupt rate.

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

Wait specified uS.

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

Reimplemented from [BThread](#).

6.39.4 Member Data Documentation

6.39.4.1 BRtc BRtcThreaded::ortc [private]

6.39.4.2 int BRtcThreaded::orate [private]

6.39.4.3 BCond BRtcThreaded::ocond [private]

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

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

6.40 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](#)

6.40.1 Detailed Description

thread read-write locks

6.40.2 Constructor & Destructor Documentation

6.40.2.1 BRWLock::BRWLock ()

6.40.2.2 BRWLock::BRWLock (const BRWLock & rwlock)

6.40.2.3 BRWLock::~~BRWLock ()

6.40.3 Member Function Documentation

6.40.3.1 int BRWLock::rdLock ()

Set lock, wait if necessary.

6.40.3.2 int BRWLock::tryRdLock ()

Test the lock.

6.40.3.3 int BRWLock::wrLock ()

Set lock, wait if necessary.

6.40.3.4 int BRWLock::tryWrLock ()

Test the lock.

6.40.3.5 int BRWLock::unlock ()

Unlock the lock.

6.40.3.6 BRWLock & BRWLock::operator= (const BRWLock & *rwlock*)**6.40.4 Member Data Documentation****6.40.4.1 pthread_rwlock_t BRWLock::olock [private]**

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

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

6.41 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](#)

6.41.1 Detailed Description

Sempahore class.

6.41.2 Constructor & Destructor Documentation

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

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

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

6.41.3 Member Function Documentation

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

Post condition.

6.41.3.2 int BSema::wait ()

Wait for contition.

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

Wait for condition with timeout.

6.41.3.4 int BSema::tryWait ()

Test for the condition.

6.41.3.5 int BSema::getValue () const**6.41.3.6 BSema & BSema::operator= (const BSema & *sema*)****6.41.4 Member Data Documentation****6.41.4.1 sem_t BSema::osema [private]**

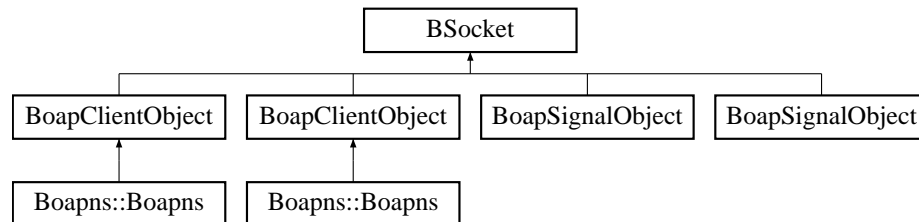
The documentation for this class was generated from the following files:

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

6.42 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](#)

6.42.1 Member Enumeration Documentation

6.42.1.1 enum BSocket::NType

Enumerator:

STREAM

DGRAM

6.42.1.2 enum BSocket::Priority

Enumerator:

PriorityLow

PriorityNormal

PriorityHigh

6.42.2 Constructor & Destructor Documentation

6.42.2.1 BSocket::BSocket ()

6.42.2.2 BSocket::BSocket (int *fd*)

6.42.2.3 BSocket::BSocket (NType *type*)

6.42.2.4 BSocket::~~BSocket ()

6.42.3 Member Function Documentation

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

6.42.3.2 int BSocket::getFd ()

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

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

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

6.42.3.6 BError BSocket::close ()

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

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

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

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

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

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

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

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

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

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

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

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

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

Generated on Thu Dec 18 13:21:31 2008 for LibBeamApi by Doxygen

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

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

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

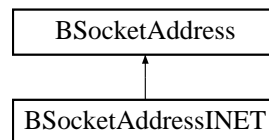
- [BSocket.h](#)
- [BSocket.cpp](#)

6.43 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](#)

6.43.1 Detailed Description

Socket Address.

6.43.2 Member Typedef Documentation

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

6.43.3 Constructor & Destructor Documentation

6.43.3.1 `BSocketAddress::BSocketAddress ()`

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

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

6.43.3.4 `BSocketAddress::~~BSocketAddress ()`

6.43.4 Member Function Documentation

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

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

6.43.4.3 `int BSocketAddress::len () const`

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

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

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

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

6.43.5 Member Data Documentation

6.43.5.1 `int BSocketAddress::olen` [private]

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

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

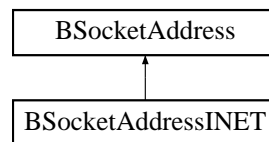
- [BSocket.h](#)
- [BSocket.cpp](#)

6.44 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.

6.44.1 Detailed Description

IP aware socket address.

6.44.2 Member Typedef Documentation

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

6.44.3 Member Function Documentation

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

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

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

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

6.44.3.5 `uint32_t BSocketAddressINET::address ()`

Returns socket ip address.

6.44.3.6 `uint32_t BSocketAddressINET::port ()`

Returns socket port.

6.44.3.7 `BString BSocketAddressINET::getString ()`

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

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

Get this hosts network name.

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

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

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

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

6.44.3.11 `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](#)

6.45 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)
- virtual [~BString](#) ()
- [BString copy](#) ()
Return an independant copy.
- virtual void [strChanged](#) ()
- 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.
- 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 add](#) (const [BString](#) &str) const
Add two strings.
- [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 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 [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 field](#) (int field) const

- char ** fields ()
- 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

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

6.45.1 Constructor & Destructor Documentation

6.45.1.1 BString::BString ()

6.45.1.2 BString::BString (const BString & *string*)

6.45.1.3 BString::BString (const char * *str*)

6.45.1.4 BString::BString (char *ch*)

6.45.1.5 BString::BString (int *v*)

6.45.1.6 BString::BString (unsigned int *v*)

6.45.1.7 BString::BString (long *v*)

6.45.1.8 BString::BString (unsigned long long *value*)

6.45.1.9 BString::BString (double *v*)

6.45.1.10 BString::~BString () [virtual]

6.45.2 Member Function Documentation

6.45.2.1 BString BString::convert (char *ch*) [static]

Converts char to string.

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

Converts int to string.

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

Converts uint to string.

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

Converts long to string.

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

Converts double to string.

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

Converts u long long to string.

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

Converts int to string as hex value.

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

Converts uint to string as hex value.

6.45.2.9 BString BString::copy ()

Return an independant copy.

6.45.2.10 void BString::strChanged () [virtual]**6.45.2.11 int BString::len () const**

Length of string.

6.45.2.12 const char * BString::retStr () const

Ptr to char* representation.

6.45.2.13 char * BString::retStrDup () const

Ptr to newly malloc'd char*.

6.45.2.14 int BString::retInt () const

Return string as a int.

6.45.2.15 double BString::retDouble () const

Return string as a double.

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

Compare strings.

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

Compare string to string with wildcards.

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

Compare string to space delimited patterns.

6.45.2.19 BString BString::add (const BString & *str*) const

Add two strings.

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

Truncate to length len.

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

Pad to length len.

6.45.2.22 BString & BString::toUpper ()

Convert to uppercase.

6.45.2.23 BString & BString::toLower ()

Convert to lowercase.

6.45.2.24 void BString::removeNL ()

Remove if present NL from last char.

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

Returns substring.

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

Delete substring.

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

Insert substring.

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

Formatted print into the string.

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

Find *ch* in string searching forwards.

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

Find *ch* in string searching backwards.

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

Break string into tokens.

6.45.2.32 BString BString::removeSeparators (BString *separators*)

Remove any char from separators from string.

6.45.2.33 BString BString::pullToken (BString *terminators*)

Pull token from start of string.

6.45.2.34 BString BString::pullSeparators (BString *separators*)

Pull separators from start of string.

6.45.2.35 BString BString::pullWord ()

Pull a word out of the head of the string.

6.45.2.36 BString BString::pullLine ()

Pull a line out of the head of the string.

6.45.2.37 BString BString::field (int *field*) const**6.45.2.38 char ** BString::fields ()****6.45.2.39 BString & BString::operator= (const BString & *string*)****6.45.2.40]**

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

- 6.45.2.41 `int BString::operator==(const BString & s) const` [inline]
- 6.45.2.42 `int BString::operator==(const char * s) const` [inline]
- 6.45.2.43 `int BString::operator>(const BString & s) const` [inline]
- 6.45.2.44 `int BString::operator>(const char * s) const` [inline]
- 6.45.2.45 `int BString::operator<(const BString & s) const` [inline]
- 6.45.2.46 `int BString::operator<(const char * s) const` [inline]
- 6.45.2.47 `int BString::operator>=(const BString & s) const` [inline]
- 6.45.2.48 `int BString::operator<=(const BString & s) const` [inline]
- 6.45.2.49 `int BString::operator!=(const BString & s) const` [inline]
- 6.45.2.50 `int BString::operator!=(const char * s) const` [inline]
- 6.45.2.51 `BString BString::operator+(const BString & s) const` [inline]
- 6.45.2.52 `BString BString::operator+(const char * s) const` [inline]
- 6.45.2.53 `BString BString::operator+=(const BString & s)` [inline]
- 6.45.2.54 `BString BString::operator+=(const char * s)` [inline]
- 6.45.2.55 `BString BString::operator+(char ch) const` [inline]
- 6.45.2.56 `BString BString::operator+(int i) const` [inline]
- 6.45.2.57 `BString BString::operator+(unsigned int i) const` [inline]
- 6.45.2.58 `BString BString::operator+(unsigned long long i) const` [inline]
- 6.45.2.59 `BString::operator const char * () const` [inline]
- 6.45.2.60 `void BString::Init(const char * str)` [private]
- 6.45.2.61 `int BString::inString(int pos) const` [private]
- 6.45.2.62 `int BString::isSpace(char ch) const` [private]

6.45.3 Member Data Documentation

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

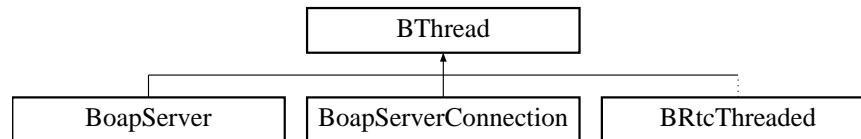
The documentation for this class was generated from the following files:

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

6.46 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](#)

6.46.1 Constructor & Destructor Documentation

6.46.1.1 `BThread::BThread ()`

6.46.1.2 `BThread::~~BThread ()` [virtual]

6.46.2 Member Function Documentation

6.46.2.1 `int BThread::setInitPriority (int policy, int priority)`

6.46.2.2 `int BThread::setInitStackSize (size_t stackSize)`

6.46.2.3 `int BThread::start ()`

6.46.2.4 `void * BThread::result ()`

6.46.2.5 `int BThread::running ()`

6.46.2.6 `int BThread::setPriority (int policy, int priority)`

6.46.2.7 `int BThread::cancel ()`

6.46.2.8 `void * BThread::waitForCompletion ()`

6.46.2.9 `pthread_t BThread::getThread ()`

6.46.2.10 `void * BThread::function ()` [virtual]

Reimplemented in [BoapServerConnection](#), [BoapServer](#), and [BRtcThreaded](#).

6.46.2.11 `void * BThread::startFunc (void * arg)` [static, private]

6.46.3 Member Data Documentation

6.46.3.1 `pthread_t BThread::othread` [private]

6.46.3.2 `size_t BThread::ostackSize` [private]

6.46.3.3 `int BThread::opolicy` [private]

6.46.3.4 `int BThread::opriority` [private]

6.46.3.5 `int BThread::orunning` [private]

6.46.3.6 `void* BThread::oresult` [private]

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

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

6.47 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](#)

6.47.1 Detailed Description

Stopwatch style timer.

6.47.2 Constructor & Destructor Documentation

6.47.2.1 BTimer::BTimer ()

6.47.2.2 BTimer::~~BTimer ()

6.47.3 Member Function Documentation

6.47.3.1 void BTimer::start ()

Start timer.

6.47.3.2 void BTimer::stop ()

Stop timer.

6.47.3.3 void BTimer::clear ()

Clear timer.

6.47.3.4 double BTimer::getElapsedTime ()

Returns the elapsed time from the last start.

6.47.3.5 void BTimer::add (BTimer & *timer*)

Add two timers.

6.47.3.6 double BTimer::average ()

Average time is duration between [start\(\)](#) and [stop\(\)](#) / number of stops.

6.47.3.7 double BTimer::peak ()

Peak time.

6.47.3.8 `double BTimer::getTime ()` [static, private]

6.47.4 Member Data Documentation

6.47.4.1 `BMutex BTimer::olock` [private]

6.47.4.2 `unsigned int BTimer::onum` [private]

6.47.4.3 `double BTimer::ostartTime` [private]

6.47.4.4 `double BTimer::oendTime` [private]

6.47.4.5 `double BTimer::oaverage` [private]

6.47.4.6 `double BTimer::opeak` [private]

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

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

6.48 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](#)

6.48.1 Detailed Description

Basic access to a Url.

6.48.2 Constructor & Destructor Documentation

6.48.2.1 BUrl::BUrl ()

6.48.2.2 BUrl::~~BUrl ()

6.48.3 Member Function Documentation

6.48.3.1 BError BUrl::readString (BString url, BString & str)

Reads URL.

6.48.3.2 `size_t BUrl::writeData (void * data, size_t size, size_t elSize, void * stream)` [static, private]

6.48.4 Member Data Documentation

6.48.4.1 `int BUrl::oinit` [static, private]

6.48.4.2 `BString BUrl::ores` [private]

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

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

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 BBuffer.cpp File Reference

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

Defines

- #define [SIZE](#) 1024

7.2.1 Define Documentation

7.2.1.1 #define SIZE 1024

7.3 BBuffer.h File Reference

```
#include <stdint.h>
```

Classes

- class [BBuffer](#)

Defines

- #define [BBUFFER_H](#) 1

7.3.1 Define Documentation

7.3.1.1 #define BBUFFER_H 1

7.4 BCond.cpp File Reference

```
#include <BCond.h>  
#include <sys/time.h>  
#include <stdio.h>
```


7.5 BCond.h File Reference

```
#include <pthread.h>
```

Classes

- class [BCond](#)

Defines

- #define [BCOND_H](#) 1

7.5.1 Define Documentation

7.5.1.1 #define BCOND_H 1

7.6 BCondInt.cpp File Reference

```
#include <BCondInt.h>
#include <sys/time.h>
#include <stdio.h>
#include <errno.h>
```

7.7 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

7.7.1 Define Documentation

7.7.1.1 #define BCONDINT_H 1

7.8 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](#)

7.8.1 Function Documentation

7.8.1.1 static int wild (const dirent *e) [static]

7.8.2 Variable Documentation

7.8.2.1 BString wildString [static]

7.9 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

7.9.1 Define Documentation

7.9.1.1 #define BDIR_H 1

7.10 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>
```

7.11 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.

7.12 BError.cpp File Reference

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


7.13 BError.h File Reference

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

Classes

- class [BError](#)
Error return class.

Defines

- #define [BERROR_H](#) 1

7.13.1 Define Documentation

7.13.1.1 #define BERROR_H 1

7.14 BEvent.cpp File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <BEvent.h>
#include <BPoll.h>
```

7.15 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](#) }

7.15.1 Define Documentation

7.15.1.1 #define BEvent_H 1

7.15.2 Enumeration Type Documentation

7.15.2.1 enum BEventType

Enumerator:

BEventTypeNone

BEventTypeInt

BEventTypeError

7.16 BFile.cpp File Reference

```
#include <BFile.h>
#include <sys/stat.h>
#include <string.h>
#include <stdarg.h>
#include <errno.h>
```

Defines

- #define [STRBUF](#) 10240

7.16.1 Define Documentation

7.16.1.1 #define STRBUF 10240

7.17 BFile.h File Reference

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

Classes

- class [BFile](#)
File operations class.

Defines

- #define [BFILE_H](#) 1

7.17.1 Define Documentation

7.17.1.1 #define BFILE_H 1

7.18 BList.h File Reference

```
#include <BList_func.h>
```

Classes

- class [BIter](#)
Iterator for [BList](#).
- class [BList< T >](#)
Template based list class.
- class [BList< T >::Node](#)

Defines

- #define [BLIST_H](#) 1

7.18.1 Define Documentation

7.18.1.1 #define BLIST_H 1

7.19 BList_func.h File Reference

```
#include <stdlib.h>  
#include <stdio.h>  
#include <memory.h>
```

7.20 BMutex.cpp File Reference

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

Defines

- #define [MDEBUG](#) 0

7.20.1 Define Documentation

7.20.1.1 #define MDEBUG 0

7.21 BMutex.h File Reference

```
#include <pthread.h>
```

Classes

- class [BMutex](#)
Mutex class.

Defines

- #define [BMUTEX_H](#) 1

7.21.1 Define Documentation

7.21.1.1 #define BMUTEX_H 1

7.22 BNameValue.h File Reference

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

Classes

- class [BNameValue< T >](#)
- class [BNameValueList< T >](#)

Defines

- #define [BNAMEVALUE_H](#) 1
- #define [TEMPLATE_NEW](#) 1

7.22.1 Define Documentation

7.22.1.1 [#define BNAMEVALUE_H](#) 1

7.22.1.2 [#define TEMPLATE_NEW](#) 1

7.23 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

Functions

- static void [swap8](#) (char *d, char *s)
- static void [swap16](#) (char *d, char *s)
- static void [swap32](#) (char *d, char *s)
- static void [swap64](#) (char *d, char *s)

Variables

- const int [boapPort](#) = 12000

The default BOAP connection port.

- const int [roundSize](#) = 256

7.23.1 Define Documentation

7.23.1.1 `#define APIVERSION_TEST 1`

7.23.1.2 `#define DEBUG 0`

7.23.1.3 `#define dprintf(fmt, a...)`

7.23.1.4 `#define IS_BIG_ENDIAN 1`

7.23.2 Function Documentation

7.23.2.1 `static void swap16 (char * d, char * s)` [inline, static]

7.23.2.2 `static void swap32 (char * d, char * s)` [inline, static]

7.23.2.3 `static void swap64 (char * d, char * s)` [inline, static]

7.23.2.4 `static void swap8 (char * d, char * s)` [inline, static]

7.23.3 Variable Documentation

7.23.3.1 `const int boapPort = 12000`

The default BOAP connection port.

7.23.3.2 `const int roundSize = 256`

7.24 Boap.h File Reference

```
#include <stdint.h>
#include <BPoll.h>
#include <BSocket.h>
#include <BThread.h>
#include <BError.h>
#include <BEvent.h>
#include <BMutex.h>
#include <BTypes.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 [UInt32](#) [BoapService](#)
- typedef [BError](#)([BoapServiceObject](#)::* [BoapFunc](#))([BoapServerConnection](#) *conn, [BoapPacket](#) &rx, [BoapPacket](#) &tx)

Enumerations

- enum [BoapType](#) {
 [BoapTypeRpc](#), [BoapTypeRpcReply](#), [BoapTypeSignal](#), [BoapTypeRpc](#),
 [BoapTypeSignal](#) }
- enum [BoapPriority](#) { [BoapPriorityLow](#), [BoapPriorityNormal](#), [BoapPriorityHigh](#) }

Variables

- const [UInt32](#) [BoapMagic](#) = 0x424F4100

7.24.1 Typedef Documentation

7.24.1.1 `typedef BError(BoapServiceObject::* BoapFunc)(BoapServerConnection *conn, BoapPacket &rx, BoapPacket &tx)`

7.24.1.2 `typedef UInt32 BoapService`

7.24.2 Enumeration Type Documentation

7.24.2.1 `enum BoapPriority`

Enumerator:

BoapPriorityLow

BoapPriorityNormal

BoapPriorityHigh

7.24.2.2 `enum BoapType`

Enumerator:

BoapTypeRpc

BoapTypeRpcReply

BoapTypeSignal

BoapTypeRpc

BoapTypeSignal

7.24.3 Variable Documentation

7.24.3.1 `const UInt32 BoapMagic = 0x424F4100`

7.25 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)

7.26 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

7.26.1 Define Documentation

7.26.1.1 #define BOAPNSC_H 1

7.27 BoapnsD.cc File Reference

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

Namespaces

- namespace [Boapns](#)

7.28 BoapnsD.h File Reference

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

Namespaces

- namespace [Boapns](#)

Classes

- class [Boapns::BoapEntry](#)

Defines

- #define [BOAPNSD_H](#) 1

7.28.1 Define Documentation

7.28.1.1 #define BOAPNSD_H 1

7.29 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

7.29.1 Define Documentation

7.29.1.1 #define `DEBUG` 0

7.29.1.2 #define `dprintf`(fmt, a...)

7.29.2 Variable Documentation

7.29.2.1 const int `roundSize` = 256

7.30 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](#), [BoapTypeRpc](#),
 [BoapTypeSignal](#) }

7.30.1 Typedef Documentation

7.30.1.1 `typedef BError(BoapServiceObject::* BoapFunc)(BoapPacket &rx, BoapPacket &tx)`

7.30.1.2 `typedef uint32_t BoapService`

7.30.1.3 `typedef double Double`

7.30.1.4 `typedef int16_t Int16`

7.30.1.5 `typedef int32_t Int32`

7.30.1.6 `typedef int8_t Int8`

7.30.1.7 `typedef uint16_t UInt16`

7.30.1.8 `typedef uint32_t UInt32`

7.30.1.9 `typedef uint8_t UInt8`

7.30.2 Enumeration Type Documentation

7.30.2.1 `enum BoapType`

Enumerator:

BoapTypeRpc

BoapTypeRpcReply

BoapTypeSignal

BoapTypeRpc

BoapTypeSignal

7.31 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

7.31.1 Define Documentation

7.31.1.1 #define DEBUG 0

7.32 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](#)

7.32.1 Define Documentation

7.32.1.1 #define BOBJECT_H 1

7.32.2 Typedef Documentation

7.32.2.1 typedef BNameValue<BObject*> BMember

7.32.2.2 typedef BNameValueList<BObject*> BMemberList

7.33 BPoll-1.cpp File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <BPoll.h>
```


7.34 BPoll.cpp File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <BPoll.h>
```

7.35 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

7.35.1 Define Documentation

7.35.1.1 #define BPOLL_H 1

7.36 BRefData.cpp File Reference

```
#include <stdlib.h>
#include <string.h>
#include <BRefData.h>
```

Defines

- #define [DEBUG](#) 0
- #define [CHUNK](#) 16

7.36.1 Define Documentation

7.36.1.1 #define [CHUNK](#) 16

7.36.1.2 #define [DEBUG](#) 0

7.37 BRefData.h File Reference

Classes

- class [BRefData](#)
Referenced data storage.

Defines

- `#define BREFDATA_H 1`

7.37.1 Define Documentation

7.37.1.1 `#define BREFDATA_H 1`

7.38 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>
```

7.39 BRtc.h File Reference

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

Classes

- class [BRtc](#)
Realtime clock.
- class [BRtcThreaded](#)
Threaded real time clock.

7.40 BRWLock.cpp File Reference

```
#include <BRWLock.h>
```

7.41 BRWLock.h File Reference

```
#include <pthread.h>
```

Classes

- class [BRWLock](#)
thread read-write locks

Defines

- #define [BRWLOCK_H](#) 1

7.41.1 Define Documentation

7.41.1.1 #define BRWLOCK_H 1

7.42 BSema.cpp File Reference

```
#include <BSema.h>  
#include <errno.h>  
#include <sys/time.h>
```

7.43 BSema.h File Reference

```
#include <semaphore.h>
```

Classes

- class [BSema](#)
Sempahore class.

Defines

- #define [BSEMA_H](#) 1

7.43.1 Define Documentation

7.43.1.1 #define BSEMA_H 1

7.44 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

7.44.1 Define Documentation

7.44.1.1 #define IP_MTU 14

7.45 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

7.45.1 Define Documentation

7.45.1.1 #define BSOCKET_H 1

7.46 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)

7.46.1 Define Documentation

7.46.1.1 #define [DEBUG](#) 0

7.46.1.2 #define [MINUS](#) '-'

7.46.1.3 #define [STRIP](#) 0x7f

7.46.2 Function Documentation

7.46.2.1 static int [gmatch](#) (const char *s, const char *p) [static]

7.46.2.2 std::ostream& [operator<<](#) (std::ostream &o, [BString](#) &s)

7.46.2.3 std::istream& [operator>>](#) (std::istream &i, [BString](#) &s)

7.47 BString.h File Reference

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

7.47.1 Define Documentation

7.47.1.1 #define BSTRING_H 1

7.47.2 Function Documentation

7.47.2.1 std::ostream& operator<< (std::ostream & o, BString & s)

7.47.2.2 std::istream& operator>> (std::istream & i, BString & s)

7.48 BThread.cpp File Reference

```
#include <BThread.h>
#include <unistd.h>
#include <errno.h>
#include <sys/types.h>
```

7.49 BThread.h File Reference

```
#include <pthread.h>
```

Classes

- class [BThread](#)

Defines

- #define [BTHREAD_H](#) 1

7.49.1 Define Documentation

7.49.1.1 #define BTHREAD_H 1

7.50 BTimer.cpp File Reference

```
#include <BTimer.h>  
#include <sys/time.h>
```

7.51 BTimer.h File Reference

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

Classes

- class [BTimer](#)
Stopwatch style timer.

7.52 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 [BFloat](#)
- typedef double [BDouble](#)
- typedef size_t [BSize](#)
- typedef uint32_t [BUInt](#)
- typedef std::vector< float > [BArrayFloat](#)
- typedef std::vector< double > [BArrayDouble](#)
- 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 int64_t [Int64](#)
- typedef uint64_t [UInt64](#)
- typedef float [Float](#)
- typedef double [Double](#)

7.52.1 Define Documentation

7.52.1.1 `#define BTYPES_H 1`

7.52.2 Typedef Documentation

7.52.2.1 `typedef std::vector<double> BArrayDouble`

7.52.2.2 `typedef std::vector<float> BArrayFloat`

7.52.2.3 `typedef double BDouble`

7.52.2.4 `typedef float BFloat`

7.52.2.5 `typedef int16_t BInt16`

7.52.2.6 `typedef int32_t BInt32`

7.52.2.7 `typedef int64_t BInt64`

7.52.2.8 `typedef int8_t BInt8`

7.52.2.9 `typedef size_t BSize`

7.52.2.10 `typedef uint32_t BUInt`

7.52.2.11 `typedef uint16_t BUInt16`

7.52.2.12 `typedef uint32_t BUInt32`

7.52.2.13 `typedef uint64_t BUInt64`

7.52.2.14 `typedef uint8_t BUInt8`

7.52.2.15 `typedef double Double`

7.52.2.16 `typedef float Float`

7.52.2.17 `typedef int16_t Int16`

7.52.2.18 `typedef int32_t Int32`

7.52.2.19 `typedef int64_t Int64`

7.52.2.20 `typedef int8_t Int8`

7.52.2.21 `typedef uint16_t UInt16`

7.52.2.22 `typedef uint32_t UInt32`

7.52.2.23 `typedef uint64_t UInt64`

7.52.2.24 `typedef uint8_t UInt8`

7.53 BUrl.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
#include <BUrl.h>
#include <curl/curl.h>
```

7.54 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

7.54.1 Define Documentation

7.54.1.1 #define BURL_H 1

Index

- ~BBuffer
 - BBuffer, [12](#)
- ~BCond
 - BCond, [14](#)
- ~BCondBool
 - BCondBool, [15](#)
- ~BCondInt
 - BCondInt, [18](#)
- ~BCondValue
 - BCondValue, [21](#)
- ~BCondWrap
 - BCondWrap, [24](#)
- ~BDir
 - BDir, [27](#)
- ~BEntryFile
 - BEntryFile, [33](#)
- ~BEvent
 - BEvent, [40](#)
- ~BEventInt
 - BEventInt, [43](#)
- ~BEventPipe
 - BEventPipe, [45](#)
- ~BFile
 - BFile, [48](#)
- ~BList
 - BList, [55](#)
- ~BMutex
 - BMutex, [60](#)
- ~BObject
 - BObject, [92](#)
- ~BPoll
 - BPoll, [94](#)
- ~BRWLock
 - BRWLock, [102](#)
- ~BRefData
 - BRefData, [97](#)
- ~BRtc
 - BRtc, [98](#)
- ~BRtcThreaded
 - BRtcThreaded, [100](#)
- ~BSema
 - BSema, [104](#)
- ~BSocket
 - BSocket, [109](#)
- ~BSocketAddress
 - BSocketAddress, [112](#)
- ~BString
 - BString, [119](#)
- ~BThread
 - BThread, [125](#)
- ~BTimer
 - BTimer, [127](#)
- ~BUrl
 - BUrl, [129](#)
- ~BoapPacket
 - BoapPacket, [74](#)
- ~BoapServer
 - BoapServer, [79](#)
- ~BoapServiceObject
 - BoapServiceObject, [87](#)
- accept
 - BSocket, [109](#)
- add
 - BString, [121](#)
 - BTimer, [127](#)
- addEntry
 - Boapns, [10](#)
 - Boapns::Boapns, [70](#)
- addMember
 - BObject, [92](#)
- addObject
 - BoapServer, [79](#), [80](#)
- addRef
 - BRefData, [97](#)
- address
 - BSocketAddressINET, [114](#)
- addressList
 - Boapns::BoapEntry, [68](#)
- apiVersion
 - Boapns, [10](#)
- APIVERSION_TEST
 - Boap.cpp, [154](#)
- append
 - BList, [56](#), [57](#)
 - BPoll, [94](#)
- average
 - BTimer, [127](#)
- BArray, [11](#)

- BArray, 11
- BArray.h, 131
 - BArray_H, 131
- BArray_H
 - BArray.h, 131
- BArrayDouble
 - BTypes.h, 187
- BArrayFloat
 - BTypes.h, 187
- BBuffer, 12
 - ~BBuffer, 12
 - BBuffer, 12
 - data, 13
 - odata, 13
 - odatasize, 13
 - osize, 13
 - setData, 12
 - setSize, 12
 - size, 13
 - writeData, 12
- BBuffer.cpp, 132
- SIZE, 132
- BBuffer.h, 133
 - BBUFFER_H, 133
- BBUFFER_H
 - BBuffer.h, 133
- BCond, 14
 - ~BCond, 14
 - BCond, 14
 - ocond, 14
 - omutex, 14
 - signal, 14
 - timedWait, 14
 - wait, 14
- BCond.cpp, 134
- BCond.h, 135
 - BCOND_H, 135
- BCOND_H
 - BCond.h, 135
- BCondBool, 15
 - ~BCondBool, 15
 - BCondBool, 15
 - clear, 15
 - ocond, 16
 - omutex, 16
 - ovalue, 16
 - set, 15
 - timedWait, 16
 - value, 16
 - wait, 16
- BCondInt, 17
 - ~BCondInt, 18
 - BCondInt, 18
 - decrement, 18
 - increment, 18
 - ocond, 19
 - omutex, 19
 - operator++, 19
 - operator-, 19
 - ovalue, 19
 - setValue, 18
 - timedWait, 18
 - tryNotZeroDecrement, 18
 - value, 18
 - wait, 18
 - waitIncrement, 18
 - waitNotZero, 18
 - waitNotZeroDecrement, 18
- BCondInt.cpp, 136
- BCondInt.h, 137
 - BCONDINT_H, 137
- BCONDINT_H
 - BCondInt.h, 137
- BCondValue, 20
 - ~BCondValue, 21
 - BCondValue, 21
 - decrement, 21
 - increment, 21
 - ocond, 22
 - omutex, 22
 - operator++, 21
 - operator+=", 21
 - operator-, 22
 - operator=, 21
 - ovalue, 22
 - setValue, 21
 - value, 21
 - waitLessThan, 21
 - waitLessThanOrEqual, 21
 - waitMoreThanOrEqual, 21
- BCondWrap, 23
 - ~BCondWrap, 24
 - BCondWrap, 24
 - decrement, 24
 - diff, 25
 - increment, 24
 - ocond, 25
 - omutex, 25
 - operator++, 25
 - operator+=", 24
 - operator-, 25
 - operator=, 24
 - ovalue, 25
 - setValue, 24
 - value, 24
 - waitLessThan, 24
 - waitLessThanOrEqual, 24
 - waitMoreThanOrEqual, 24

- BDir, 26
 - ~BDir, 27
 - BDir, 27
 - clear, 27
 - entryName, 27
 - entryStat, 28
 - entryStat64, 28
 - error, 27
 - odirname, 28
 - oerror, 28
 - open, 27
 - osort, 28
 - owild, 28
 - read, 27
 - setSort, 27
 - setWild, 27
- BDir.cpp, 138
 - wild, 138
 - wildString, 138
- BDir.h, 139
 - BDir_H, 139
- BDir_H
 - BDir.h, 139
- BDouble
 - BTypes.h, 187
- begin
 - BList, 55
- BEntry, 29
 - BEntry, 30
 - getName, 30
 - getValue, 30
 - line, 30
 - oname, 31
 - ovalue, 31
 - print, 30
 - setLine, 30
 - setName, 30
 - setValue, 30
- BEntry.cpp, 140
- BEntry.h, 141
- BEntryFile, 32
 - ~BEntryFile, 33
 - BEntryFile, 33
 - clear, 33
 - ocomments, 33
 - ofilename, 33
 - open, 33
 - read, 33
 - write, 33
 - writeList, 33
- BEntryList, 34
 - BEntryList, 35
 - clear, 36
 - del, 36
 - deleteEntry, 35
 - find, 35
 - findValue, 35
 - getString, 35
 - insert, 35
 - isSet, 35
 - olastPos, 36
 - print, 35
 - setValue, 35
 - setValueRaw, 35
- BError, 37
 - BError, 38
 - copy, 38
 - ERROR, 38
 - getErrorNo, 38
 - getString, 38
 - NONE, 38
 - oerrNo, 39
 - oerrStr, 39
 - operator int, 38
 - set, 38
 - setError, 38
 - Type, 38
- BError.cpp, 142
- BError.h, 143
 - BERROR_H, 143
- BERROR_H
 - BError.h, 143
- BEvent, 40
 - ~BEvent, 40
 - BEvent, 40
 - getBinary, 40
 - getType, 40
 - otype, 41
 - setBinary, 40
- BEvent.cpp, 144
- BEvent.h, 145
 - BEvent_H, 145
 - BEventType, 145
 - BEventTypeError, 145
 - BEventTypeInt, 145
 - BEventTypeNone, 145
- BEvent_H
 - BEvent.h, 145
- BEventError, 42
 - BEventError, 42
 - getBinary, 42
 - setBinary, 42
- BEventInt, 43
 - ~BEventInt, 43
 - BEventInt, 43
 - getEvent, 43
 - getFd, 43
 - ofds, 44

- sendEvent, [43](#)
- BEventPipe, [45](#)
 - ~BEventPipe, [45](#)
 - BEventPipe, [45](#)
 - getEvent, [45](#)
 - getReceiveFd, [45](#)
 - ofds, [46](#)
 - sendEvent, [45](#)
- BEventType
 - BEvent.h, [145](#)
- BEventTypeError
 - BEvent.h, [145](#)
- BEventTypeInt
 - BEvent.h, [145](#)
- BEventTypeNone
 - BEvent.h, [145](#)
- BFile, [47](#)
 - ~BFile, [48](#)
 - BFile, [48](#)
 - close, [48](#)
 - error, [48](#)
 - getFd, [49](#)
 - length, [49](#)
 - oerror, [50](#)
 - ofile, [50](#)
 - ofilename, [50](#)
 - omode, [50](#)
 - open, [48](#)
 - operator=, [49](#)
 - printf, [49](#)
 - read, [49](#)
 - readString, [49](#)
 - seek, [49](#)
 - setVBuf, [49](#)
 - write, [49](#)
 - writeString, [49](#)
- BFile.cpp, [146](#)
 - STRBUF, [146](#)
- BFile.h, [147](#)
 - BFILE_H, [147](#)
- BFILE_H
 - BFile.h, [147](#)
- BFloat
 - BTypes.h, [187](#)
- bind
 - BSocket, [109](#)
- BInt16
 - BTypes.h, [187](#)
- BInt32
 - BTypes.h, [187](#)
- BInt64
 - BTypes.h, [187](#)
- BInt8
 - BTypes.h, [187](#)
- BIter, [51](#)
 - BIter, [51](#)
 - oi, [51](#)
 - operator void *, [51](#)
 - operator==, [51](#)
- BList, [52](#)
 - ~BList, [55](#)
 - append, [56, 57](#)
 - begin, [55](#)
 - BList, [55](#)
 - clear, [56](#)
 - del, [56](#)
 - deleteFirst, [57](#)
 - deleteLast, [56](#)
 - end, [55](#)
 - front, [56](#)
 - get, [56](#)
 - goTo, [55](#)
 - insert, [56](#)
 - insertAfter, [56](#)
 - isEnd, [55](#)
 - next, [55](#)
 - nodeCreate, [58](#)
 - nodeGet, [58](#)
 - number, [55](#)
 - olength, [58](#)
 - onodes, [58](#)
 - operator+, [58](#)
 - operator=, [57](#)
 - pop, [57](#)
 - position, [55](#)
 - prev, [55](#)
 - push, [57](#)
 - queueAdd, [57](#)
 - queueGet, [57](#)
 - rear, [56](#)
 - sort, [57](#)
 - SortFunc, [54](#)
 - start, [55](#)
 - swap, [57](#)
- BList.h, [148](#)
 - BLIST_H, [148](#)
- BList::Node, [59](#)
 - item, [59](#)
 - next, [59](#)
 - Node, [59](#)
 - prev, [59](#)
- BList_func.h, [149](#)
- BLIST_H
 - BList.h, [148](#)
- BMember
 - BObject.h, [165](#)
- BMemberList
 - BObject.h, [165](#)

- BMutex, 60
 - ~BMutex, 60
 - BMutex, 60
 - lock, 60
 - omutex, 61
 - operator=, 61
 - tryLock, 60
 - unlock, 60
- BMutex.cpp, 150
 - MDEBUG, 150
- BMutex.h, 151
 - BMUTEX_H, 151
- BMUTEX_H
 - BMutex.h, 151
- BNameValue, 62
 - BNameValue, 62
 - getName, 62
 - getValue, 62
 - oname, 62
 - ovalue, 62
- BNameValue.h, 152
 - BNAMEVALUE_H, 152
 - TEMPLATE_NEW, 152
- BNAMEVALUE_H
 - BNameValue.h, 152
- BNameValueList, 63
 - find, 63
- Boap.cpp, 153
 - APIVERSION_TEST, 154
 - boapPort, 154
 - DEBUG, 154
 - dprintf, 154
 - IS_BIG_ENDIAN, 154
 - roundSize, 154
 - swap16, 154
 - swap32, 154
 - swap64, 154
 - swap8, 154
- Boap.h, 155
 - BoapFunc, 156
 - BoapMagic, 156
 - BoapPriority, 156
 - BoapPriorityHigh, 156
 - BoapPriorityLow, 156
 - BoapPriorityNormal, 156
 - BoapService, 156
 - BoapType, 156
 - BoapTypeRpc, 156
 - BoapTypeRpcReply, 156
 - BoapTypeSignal, 156
- BoapClientObject, 64
 - BoapClientObject, 65
 - checkApiVersion, 66
 - connectService, 65, 66
 - disconnectService, 65
 - getServiceName, 65
 - oapiVersion, 67
 - oconnected, 67
 - olock, 67
 - omaxLength, 67
 - oname, 67
 - opriority, 67
 - oreconnect, 67
 - orx, 67
 - oservice, 67
 - otimeout, 67
 - otx, 67
 - performCall, 66, 67
 - performRecv, 66, 67
 - performSend, 66, 67
 - ping, 65
 - pingLocked, 66
 - setConnectionPriority, 65
 - setMaxLength, 66
 - setTimeout, 66
- BoapEntry
 - Boapns::BoapEntry, 68
- BoapFunc
 - Boap.h, 156
 - BoapSimple.h, 163
- BoapFuncEntry, 69
 - BoapFuncEntry, 69
 - ocmd, 69
 - ofunc, 69
- BoapMagic
 - Boap.h, 156
- Boapns, 9
 - addEntry, 10
 - apiVersion, 10
 - Boapns, 10
 - Boapns::Boapns, 70
 - delEntry, 10
 - getEntry, 10
 - getEntryList, 10
 - getNewName, 10
 - getVersion, 10
- Boapns::BoapEntry, 68
 - addressList, 68
 - BoapEntry, 68
 - hostName, 68
 - name, 68
 - port, 68
 - service, 68
- Boapns::Boapns, 70
 - addEntry, 70
 - Boapns, 70
 - delEntry, 70
 - getEntry, 70

- [getEntryList](#), 70
 - [getNewName](#), 70
 - [getVersion](#), 70
- [BoapnsC.cc](#), 157
- [BoapnsC.h](#), 158
 - [BOAPNSC_H](#), 158
- [BOAPNSC_H](#)
 - [BoapnsC.h](#), 158
- [BoapnsD.cc](#), 159
- [BoapnsD.h](#), 160
 - [BOAPNSD_H](#), 160
- [BOAPNSD_H](#)
 - [BoapnsD.h](#), 160
- [BoapPacket](#), 71
 - [~BoapPacket](#), 74
 - [BoapPacket](#), 74
 - [copyWithSwap](#), 74
 - [data](#), 74
 - [getCmd](#), 74
 - [nbytes](#), 74
 - [odata](#), 74
 - [onbytes](#), 74
 - [opos](#), 74
 - [osize](#), 74
 - [peekHead](#), 74
 - [pop](#), 74
 - [popHead](#), 74
 - [push](#), 74
 - [pushHead](#), 74
 - [resize](#), 74
 - [setData](#), 74
 - [updateLen](#), 74
- [BoapPacketHead](#), 76
 - [cmd](#), 76
 - [length](#), 76
 - [reserved](#), 76
 - [service](#), 76
 - [type](#), 76
- [boapPort](#)
 - [Boap.cpp](#), 154
- [BoapPriority](#)
 - [Boap.h](#), 156
- [BoapPriorityHigh](#)
 - [Boap.h](#), 156
- [BoapPriorityLow](#)
 - [Boap.h](#), 156
- [BoapPriorityNormal](#)
 - [Boap.h](#), 156
- [BoapServer](#), 77
 - [~BoapServer](#), 79
 - [addObject](#), 79, 80
 - [BoapServer](#), 79
 - [clientGone](#), 79
 - [function](#), 79
 - [getConnectionsNumber](#), 79
 - [getEventSocket](#), 79, 80
 - [getHostName](#), 79, 80
 - [getSocket](#), 79, 80
 - [init](#), 79
 - [NOTHEADS](#), 78
 - [oboapNs](#), 80
 - [oboapns](#), 80
 - [oclientGoneEvent](#), 80
 - [oclients](#), 80
 - [ohostName](#), 80
 - [oisBoapns](#), 80
 - [onet](#), 80
 - [onetEvent](#), 80
 - [onetEventAddress](#), 80
 - [opoll](#), 80
 - [orx](#), 80
 - [oservices](#), 80
 - [othreaded](#), 80
 - [otx](#), 80
 - [process](#), 79, 80
 - [processEvent](#), 79, 80
 - [run](#), 79, 80
 - [sendEvent](#), 79, 80
 - [THREADED](#), 78
- [BoapServerConnection](#), 82
 - [BoapServerConnection](#), 82
 - [function](#), 82
 - [getSocket](#), 82
 - [oboapServer](#), 83
 - [omaxLength](#), 83
 - [orx](#), 83
 - [osocket](#), 83
 - [otx](#), 83
 - [process](#), 82
 - [setMaxLength](#), 82
- [BoapService](#)
 - [Boap.h](#), 156
 - [BoapSimple.h](#), 163
- [BoapServiceEntry](#), 84
 - [BoapServiceEntry](#), 84
 - [oobject](#), 84
 - [oservice](#), 84
- [BoapServiceObject](#), 85
 - [~BoapServiceObject](#), 87
 - [BoapServiceObject](#), 87
 - [doConnectionPriority](#), 87
 - [doPing](#), 87
 - [name](#), 87
 - [oapiVersion](#), 87
 - [ofuncList](#), 87
 - [oname](#), 87
 - [oserver](#), 87
 - [process](#), 87

- processEvent, 87
- sendEvent, 87
- setName, 87
- BoapSignalObject, 89
 - BoapSignalObject, 89
 - orx, 89
 - otx, 89
 - performSend, 89
- BoapSimple.cc, 161
 - DEBUG, 161
 - dprintf, 161
 - roundSize, 161
- BoapSimple.h, 162
 - BoapFunc, 163
 - BoapService, 163
 - BoapType, 163
 - BoapTypeRpc, 163
 - BoapTypeRpcReply, 163
 - BoapTypeSignal, 163
 - Double, 163
 - Int16, 163
 - Int32, 163
 - Int8, 163
 - UInt16, 163
 - UInt32, 163
 - UInt8, 163
- BoapType
 - Boap.h, 156
 - BoapSimple.h, 163
- BoapTypeRpc
 - Boap.h, 156
 - BoapSimple.h, 163
- BoapTypeRpcReply
 - Boap.h, 156
 - BoapSimple.h, 163
- BoapTypeSignal
 - Boap.h, 156
 - BoapSimple.h, 163
- BObject, 91
 - ~BObject, 92
 - addMember, 92
 - BObject, 92
 - createObj, 92
 - getBinary, 92
 - getMemberList, 92
 - getString, 92
 - getType, 92
 - otype, 92
 - printIt, 92
 - setBinary, 92
 - setString, 92
- BObject.cc, 164
 - DEBUG, 164
- BObject.h, 165
 - BMember, 165
 - BMemberList, 165
 - BOBJECT_H, 165
- BOBJECT_H
 - BObject.h, 165
- BPoll, 93
 - ~BPoll, 94
 - append, 94
 - BPoll, 94
 - clear, 94
 - delFd, 94
 - doPoll, 94
 - getPollFds, 94
 - getPollFdsNum, 94
 - nextFd, 94
 - ofds, 94
 - ofdsNext, 94
 - ofdsNum, 94
 - PollFd, 94
- BPoll-1.cpp, 166
- BPoll.cpp, 167
- BPoll.h, 168
 - B POLL_H, 168
- B POLL_H
 - BPoll.h, 168
- BRefData, 96
 - ~BRefData, 97
 - addRef, 97
 - BRefData, 97
 - copy, 97
 - data, 97
 - deleteRef, 97
 - len, 97
 - oData, 97
 - oLen, 97
 - operator=, 97
 - oRefCount, 97
 - oSize, 97
 - refCount, 97
 - setLen, 97
- BRefData.cpp, 169
 - CHUNK, 169
 - DEBUG, 169
- BRefData.h, 170
 - BREFDATA_H, 170
- BREFDATA_H
 - BRefData.h, 170
- BRtc, 98
 - ~BRtc, 98
 - BRtc, 98
 - init, 98
 - ofd, 98
 - orate, 98
 - wait, 98

- BRtc.cpp, [171](#)
- BRtc.h, [172](#)
- BRtcThreaded, [100](#)
 - ~BRtcThreaded, [100](#)
 - BRtcThreaded, [100](#)
 - function, [101](#)
 - init, [100](#)
 - ocond, [101](#)
 - orate, [101](#)
 - ortc, [101](#)
 - wait, [100](#)
- BRWLock, [102](#)
 - ~BRWLock, [102](#)
 - BRWLock, [102](#)
 - olock, [103](#)
 - operator=, [103](#)
 - rdLock, [102](#)
 - tryRdLock, [102](#)
 - tryWrLock, [103](#)
 - unlock, [103](#)
 - wrLock, [103](#)
- BRWLock.cpp, [173](#)
- BRWLock.h, [174](#)
 - BRWLOCK_H, [174](#)
- BRWLOCK_H
 - BRWLock.h, [174](#)
- BSema, [104](#)
 - ~BSema, [104](#)
 - BSema, [104](#)
 - getValue, [105](#)
 - operator=, [105](#)
 - osema, [105](#)
 - post, [104](#)
 - timedWait, [105](#)
 - tryWait, [105](#)
 - wait, [104](#)
- BSema.cpp, [175](#)
- BSema.h, [176](#)
 - BSEMA_H, [176](#)
- BSEMA_H
 - BSema.h, [176](#)
- BSize
 - BTypes.h, [187](#)
- BSocket, [106](#)
 - ~BSocket, [109](#)
 - accept, [109](#)
 - bind, [109](#)
 - BSocket, [109](#)
 - close, [109](#)
 - connect, [109](#)
 - DGRAM, [107](#)
 - getAddress, [109](#)
 - getFd, [109](#)
 - getMTU, [109](#)
 - getSockOpt, [109](#)
 - init, [109](#)
 - listen, [109](#)
 - NType, [107](#)
 - osocket, [109](#)
 - Priority, [107](#)
 - PriorityHigh, [107](#)
 - PriorityLow, [107](#)
 - PriorityNormal, [107](#)
 - recv, [109](#)
 - recvFrom, [109](#)
 - recvFromWithTimeout, [109](#)
 - recvWithTimeout, [109](#)
 - send, [109](#)
 - sendTo, [109](#)
 - setBroadCast, [109](#)
 - setPriority, [109](#)
 - setReuseAddress, [109](#)
 - setSockOpt, [109](#)
 - shutdown, [109](#)
 - STREAM, [107](#)
- BSocket.cpp, [177](#)
 - IP_MTU, [177](#)
- BSocket.h, [178](#)
 - BSOCKET_H, [178](#)
- BSOCKET_H
 - BSocket.h, [178](#)
- BSocketAddress, [111](#)
 - ~BSocketAddress, [112](#)
 - BSocketAddress, [112](#)
 - len, [112](#)
 - oaddress, [112](#)
 - olen, [112](#)
 - operator const SockAddr *, [112](#)
 - operator!=, [112](#)
 - operator=, [112](#)
 - operator==, [112](#)
 - raw, [112](#)
 - set, [112](#)
 - SockAddr, [112](#)
- BSocketAddressINET, [113](#)
 - address, [114](#)
 - getHostName, [114](#)
 - getIpAddresses, [114](#)
 - getIpAddressList, [114](#)
 - getIpAddressListAll, [114](#)
 - getString, [114](#)
 - port, [114](#)
 - set, [114](#)
 - setPort, [114](#)
 - SockAddrIP, [114](#)
- BString, [116](#)
 - ~BString, [119](#)
 - add, [121](#)

- BString, 119
- compare, 120
- compareWild, 120
- compareWildExpression, 121
- convert, 119, 120
- convertHex, 120
- copy, 120
- del, 121
- field, 122
- fields, 122
- find, 122
- findReverse, 122
- getTokenList, 122
- Init, 123
- insert, 121
- inString, 123
- isSpace, 123
- len, 120
- operator const char *, 123
- operator!=, 123
- operator<, 123
- operator<=, 123
- operator>, 123
- operator>=, 123
- operator+, 123
- operator+=, 123
- operator=, 122
- operator==, 122, 123
- ostr, 123
- pad, 121
- printf, 121
- pullLine, 122
- pullSeparators, 122
- pullToken, 122
- pullWord, 122
- removeNL, 121
- removeSeparators, 122
- retDouble, 120
- retInt, 120
- retStr, 120
- retStrDup, 120
- strChanged, 120
- subString, 121
- toLower, 121
- toUpper, 121
- truncate, 121
- BString.cpp, 179
- DEBUG, 179
- gmatch, 179
- MINUS, 179
- operator<<, 179
- operator>>, 179
- STRIP, 179
- BString.h, 180
- BSTRING_H, 180
- operator<<, 180
- operator>>, 180
- BSTRING_H
- BString.h, 180
- BThread, 124
- ~BThread, 125
- BThread, 125
- cancel, 125
- function, 125
- getThread, 125
- opolicy, 125
- opriority, 125
- oresult, 125
- orunning, 125
- ostackSize, 125
- othread, 125
- result, 125
- running, 125
- setInitPriority, 125
- setInitStackSize, 125
- setPriority, 125
- start, 125
- startFunc, 125
- waitForCompletion, 125
- BThread.cpp, 181
- BThread.h, 182
- BTHREAD_H, 182
- BTHREAD_H
- BThread.h, 182
- BTimer, 126
- ~BTimer, 127
- add, 127
- average, 127
- BTimer, 127
- clear, 127
- getElapsedTime, 127
- getTime, 127
- oaverage, 128
- oendTime, 128
- oclock, 128
- onum, 128
- opeak, 128
- ostartTime, 128
- peak, 127
- start, 127
- stop, 127
- BTimer.cpp, 183
- BTimer.h, 184
- BTypes.h, 185
- BArrayDouble, 187
- BArrayFloat, 187
- BDouble, 187
- BFloat, 187

- BInt16, [187](#)
- BInt32, [187](#)
- BInt64, [187](#)
- BInt8, [187](#)
- BSize, [187](#)
- BTYPES_H, [187](#)
- BUInt, [187](#)
- BUInt16, [187](#)
- BUInt32, [187](#)
- BUInt64, [187](#)
- BUInt8, [187](#)
- Double, [187](#)
- Float, [187](#)
- Int16, [187](#)
- Int32, [187](#)
- Int64, [187](#)
- Int8, [187](#)
- UInt16, [187](#)
- UInt32, [187](#)
- UInt64, [187](#)
- UInt8, [187](#)
- BTYPES_H
 - BTypes.h, [187](#)
- BUInt
 - BTypes.h, [187](#)
- BUInt16
 - BTypes.h, [187](#)
- BUInt32
 - BTypes.h, [187](#)
- BUInt64
 - BTypes.h, [187](#)
- BUInt8
 - BTypes.h, [187](#)
- BUrl, [129](#)
 - ~BUrl, [129](#)
 - BUrl, [129](#)
 - oinit, [130](#)
 - ores, [130](#)
 - readString, [129](#)
 - writeData, [129](#)
- BUrl.cpp, [188](#)
- BUrl.h, [189](#)
 - BURL_H, [189](#)
- BURL_H
 - BUrl.h, [189](#)
- cancel
 - BThread, [125](#)
- checkApiVersion
 - BoapClientObject, [66](#)
- CHUNK
 - BRefData.cpp, [169](#)
- clear
 - BCondBool, [15](#)
- BDir, [27](#)
- BEntryFile, [33](#)
- BEntryList, [36](#)
- BList, [56](#)
- BPoll, [94](#)
- BTimer, [127](#)
- clientGone
 - BoapServer, [79](#)
- close
 - BFile, [48](#)
 - BSocket, [109](#)
- cmd
 - BoapPacketHead, [76](#)
- compare
 - BString, [120](#)
- compareWild
 - BString, [120](#)
- compareWildExpression
 - BString, [121](#)
- connect
 - BSocket, [109](#)
- connectService
 - BoapClientObject, [65](#), [66](#)
- convert
 - BString, [119](#), [120](#)
- convertHex
 - BString, [120](#)
- copy
 - BError, [38](#)
 - BRefData, [97](#)
 - BString, [120](#)
- copyWithSwap
 - BoapPacket, [74](#)
- createObj
 - BObject, [92](#)
- data
 - BBuffer, [13](#)
 - BoapPacket, [74](#)
 - BRefData, [97](#)
- DEBUG
 - Boap.cpp, [154](#)
 - BoapSimple.cc, [161](#)
 - BObject.cc, [164](#)
 - BRefData.cpp, [169](#)
 - BString.cpp, [179](#)
- decrement
 - BCondInt, [18](#)
 - BCondValue, [21](#)
 - BCondWrap, [24](#)
- del
 - BEntryList, [36](#)
 - BList, [56](#)
 - BString, [121](#)

- delEntry
 - Boapns, 10
 - Boapns::Boapns, 70
- deleteEntry
 - BEntryList, 35
- deleteFirst
 - BList, 57
- deleteLast
 - BList, 56
- deleteRef
 - BRefData, 97
- delFd
 - BPoll, 94
- DGRAM
 - BSocket, 107
- diff
 - BCondWrap, 25
- disconnectService
 - BoapClientObject, 65
- doConnectionPriority
 - BoapServiceObject, 87
- doPing
 - BoapServiceObject, 87
- doPoll
 - BPoll, 94
- Double
 - BoapSimple.h, 163
 - BTypes.h, 187
- dprintf
 - Boap.cpp, 154
 - BoapSimple.cc, 161
- end
 - BList, 55
- entryName
 - BDir, 27
- entryStat
 - BDir, 28
- entryStat64
 - BDir, 28
- ERROR
 - BError, 38
- error
 - BDir, 27
 - BFile, 48
- field
 - BString, 122
- fields
 - BString, 122
- find
 - BEntryList, 35
 - BNameValueList, 63
 - BString, 122
- findReverse
 - BString, 122
- findValue
 - BEntryList, 35
- Float
 - BTypes.h, 187
- front
 - BList, 56
- function
 - BoapServer, 79
 - BoapServerConnection, 82
 - BRtcThreaded, 101
 - BThread, 125
- get
 - BList, 56
- getAddress
 - BSocket, 109
- getBinary
 - BEvent, 40
 - BEventError, 42
 - BObject, 92
- getCmd
 - BoapPacket, 74
- getConnectionsNumber
 - BoapServer, 79
- getElapsedTime
 - BTimer, 127
- getEntry
 - Boapns, 10
 - Boapns::Boapns, 70
- getEntryList
 - Boapns, 10
 - Boapns::Boapns, 70
- getErrorNo
 - BError, 38
- getEvent
 - BEventInt, 43
 - BEventPipe, 45
- getEventSocket
 - BoapServer, 79, 80
- getFd
 - BEventInt, 43
 - BFile, 49
 - BSocket, 109
- getHostName
 - BoapServer, 79, 80
 - BSocketAddressINET, 114
- getIpAddresses
 - BSocketAddressINET, 114
- getIpAddressList
 - BSocketAddressINET, 114
- getIpAddressListAll
 - BSocketAddressINET, 114

- getMemberList
 - BObject, [92](#)
- getMTU
 - BSocket, [109](#)
- getName
 - BEntry, [30](#)
 - BNameValue, [62](#)
- getNewName
 - Boapns, [10](#)
 - Boapns::Boapns, [70](#)
- getPollFds
 - BPoll, [94](#)
- getPollFdsNum
 - BPoll, [94](#)
- getReceiveFd
 - BEventPipe, [45](#)
- getServiceName
 - BoapClientObject, [65](#)
- getSocket
 - BoapServer, [79](#), [80](#)
 - BoapServerConnection, [82](#)
- getSockOpt
 - BSocket, [109](#)
- getString
 - BEntryList, [35](#)
 - BError, [38](#)
 - BObject, [92](#)
 - BSocketAddressINET, [114](#)
- getThread
 - BThread, [125](#)
- getTime
 - BTimer, [127](#)
- getTokenList
 - BString, [122](#)
- getType
 - BEvent, [40](#)
 - BObject, [92](#)
- getValue
 - BEntry, [30](#)
 - BNameValue, [62](#)
 - BSema, [105](#)
- getVersion
 - Boapns, [10](#)
 - Boapns::Boapns, [70](#)
- gmatch
 - BString.cpp, [179](#)
- goTo
 - BList, [55](#)
- hostName
 - Boapns::BoapEntry, [68](#)
- increment
 - BCondInt, [18](#)
 - BCondValue, [21](#)
 - BCondWrap, [24](#)
- Init
 - BString, [123](#)
- init
 - BoapServer, [79](#)
 - BRtc, [98](#)
 - BRtcThreaded, [100](#)
 - BSocket, [109](#)
- insert
 - BEntryList, [35](#)
 - BList, [56](#)
 - BString, [121](#)
- insertAfter
 - BList, [56](#)
- inString
 - BString, [123](#)
- Int16
 - BoapSimple.h, [163](#)
 - BTypes.h, [187](#)
- Int32
 - BoapSimple.h, [163](#)
 - BTypes.h, [187](#)
- Int64
 - BTypes.h, [187](#)
- Int8
 - BoapSimple.h, [163](#)
 - BTypes.h, [187](#)
- IP_MTU
 - BSocket.cpp, [177](#)
- IS_BIG_ENDIAN
 - Boap.cpp, [154](#)
- isEnd
 - BList, [55](#)
- isSet
 - BEntryList, [35](#)
- isSpace
 - BString, [123](#)
- item
 - BList::Node, [59](#)
- len
 - BRefData, [97](#)
 - BSocketAddress, [112](#)
 - BString, [120](#)
- length
 - BFile, [49](#)
 - BoapPacketHead, [76](#)
- line
 - BEntry, [30](#)
- listen
 - BSocket, [109](#)
- lock
 - BMutex, [60](#)

- MDEBUG
 - BMutex.cpp, 150
- MINUS
 - BString.cpp, 179
- name
 - Boapns::BoapEntry, 68
 - BoapServiceObject, 87
- nbytes
 - BoapPacket, 74
- next
 - BList, 55
 - BList::Node, 59
- nextFd
 - BPoll, 94
- Node
 - BList::Node, 59
- nodeCreate
 - BList, 58
- nodeGet
 - BList, 58
- NONE
 - BError, 38
- NOTHEADS
 - BoapServer, 78
- NType
 - BSocket, 107
- number
 - BList, 55
- oaddress
 - BSocketAddress, 112
- oapiVersion
 - BoapClientObject, 67
 - BoapServiceObject, 87
- oaverage
 - BTimer, 128
- oboapNs
 - BoapServer, 80
- oboapns
 - BoapServer, 80
- oboapServer
 - BoapServerConnection, 83
- oclientGoneEvent
 - BoapServer, 80
- oclients
 - BoapServer, 80
- ocmd
 - BoapFuncEntry, 69
- ocomments
 - BEntryFile, 33
- ocond
 - BCond, 14
 - BCondBool, 16
 - BCondInt, 19
 - BCondValue, 22
 - BCondWrap, 25
 - BRtcThreaded, 101
- oconnected
 - BoapClientObject, 67
- oData
 - BRefData, 97
- odata
 - BBuffer, 13
 - BoapPacket, 74
- odatasize
 - BBuffer, 13
- odirname
 - BDir, 28
- oendTime
 - BTimer, 128
- oerrNo
 - BError, 39
- oerror
 - BDir, 28
 - BFile, 50
- oerrStr
 - BError, 39
- ofd
 - BRtc, 98
- ofds
 - BEventInt, 44
 - BEventPipe, 46
 - BPoll, 94
- ofdsNext
 - BPoll, 94
- ofdsNum
 - BPoll, 94
- ofile
 - BFile, 50
- ofilename
 - BFile, 50
- ofilename
 - BEntryFile, 33
- ofunc
 - BoapFuncEntry, 69
- ofuncList
 - BoapServiceObject, 87
- ohostName
 - BoapServer, 80
- oi
 - BIter, 51
- oinit
 - BUrl, 130
- oisBoapns
 - BoapServer, 80
- olastPos
 - BEntryList, 36

- oLen
 - BRefData, 97
- olen
 - BSocketAddress, 112
- olength
 - BList, 58
- olock
 - BoapClientObject, 67
 - BRWLock, 103
 - BTimer, 128
- omaxLength
 - BoapClientObject, 67
 - BoapServerConnection, 83
- omode
 - BFile, 50
- omutex
 - BCond, 14
 - BCondBool, 16
 - BCondInt, 19
 - BCondValue, 22
 - BCondWrap, 25
 - BMutex, 61
- oname
 - BEntry, 31
 - BNameValue, 62
 - BoapClientObject, 67
 - BoapServiceObject, 87
- onbytes
 - BoapPacket, 74
- onet
 - BoapServer, 80
- onetEvent
 - BoapServer, 80
- onetEventAddress
 - BoapServer, 80
- onodes
 - BList, 58
- onum
 - BTimer, 128
- oobject
 - BoapServiceEntry, 84
- opeak
 - BTimer, 128
- open
 - BDir, 27
 - BEntryFile, 33
 - BFile, 48
- operator const char *
 - BString, 123
- operator const SockAddr *
 - BSocketAddress, 112
- operator int
 - BError, 38
- operator void *
 - BIter, 51
- operator!=
 - BSocketAddress, 112
 - BString, 123
- operator<
 - BString, 123
- operator<<
 - BString.cpp, 179
 - BString.h, 180
- operator<=
 - BString, 123
- operator>
 - BString, 123
- operator>>
 - BString.cpp, 179
 - BString.h, 180
- operator>=
 - BString, 123
- operator+
 - BList, 58
 - BString, 123
- operator++
 - BCondInt, 19
 - BCondValue, 21
 - BCondWrap, 25
- operator+=
 - BCondValue, 21
 - BCondWrap, 24
 - BString, 123
- operator-
 - BCondInt, 19
 - BCondValue, 22
 - BCondWrap, 25
- operator-=
 - BCondValue, 21
 - BCondWrap, 24
- operator=
 - BFile, 49
 - BList, 57
 - BMutex, 61
 - BRefData, 97
 - BRWLock, 103
 - BSema, 105
 - BSocketAddress, 112
 - BString, 122
- operator==
 - BIter, 51
 - BSocketAddress, 112
 - BString, 122, 123
- opolicy
 - BThread, 125
- opoll
 - BoapServer, 80
- opos

- BoapPacket, [74](#)
- opriority
 - BoapClientObject, [67](#)
 - BThread, [125](#)
- orate
 - BRtc, [98](#)
 - BRtcThreaded, [101](#)
- oreconnect
 - BoapClientObject, [67](#)
- oRefCount
 - BRefData, [97](#)
- ores
 - BUrl, [130](#)
- orresult
 - BThread, [125](#)
- ortc
 - BRtcThreaded, [101](#)
- orunning
 - BThread, [125](#)
- orx
 - BoapClientObject, [67](#)
 - BoapServer, [80](#)
 - BoapServerConnection, [83](#)
 - BoapSignalObject, [89](#)
- osema
 - BSema, [105](#)
- oserver
 - BoapServiceObject, [87](#)
- oservice
 - BoapClientObject, [67](#)
 - BoapServiceEntry, [84](#)
- oservices
 - BoapServer, [80](#)
- oSize
 - BRefData, [97](#)
- osize
 - BBuffer, [13](#)
 - BoapPacket, [74](#)
- osocket
 - BoapServerConnection, [83](#)
 - BSocket, [109](#)
- osort
 - BDir, [28](#)
- ostackSize
 - BThread, [125](#)
- ostartTime
 - BTimer, [128](#)
- ostr
 - BString, [123](#)
- othread
 - BThread, [125](#)
- othreaded
 - BoapServer, [80](#)
- otimeout
 - BoapClientObject, [67](#)
- otx
 - BoapClientObject, [67](#)
 - BoapServer, [80](#)
 - BoapServerConnection, [83](#)
 - BoapSignalObject, [89](#)
- otype
 - BEvent, [41](#)
 - BObject, [92](#)
- ovalue
 - BCondBool, [16](#)
 - BCondInt, [19](#)
 - BCondValue, [22](#)
 - BCondWrap, [25](#)
 - BEntry, [31](#)
 - BNameValue, [62](#)
- owild
 - BDir, [28](#)
- pad
 - BString, [121](#)
- peak
 - BTimer, [127](#)
- peekHead
 - BoapPacket, [74](#)
- performCall
 - BoapClientObject, [66, 67](#)
- performRecv
 - BoapClientObject, [66, 67](#)
- performSend
 - BoapClientObject, [66, 67](#)
 - BoapSignalObject, [89](#)
- ping
 - BoapClientObject, [65](#)
- pingLocked
 - BoapClientObject, [66](#)
- PollFd
 - BPoll, [94](#)
- pop
 - BList, [57](#)
 - BoapPacket, [74](#)
- popHead
 - BoapPacket, [74](#)
- port
 - Boapns::BoapEntry, [68](#)
 - BSocketAddressINET, [114](#)
- position
 - BList, [55](#)
- post
 - BSema, [104](#)
- prev
 - BList, [55](#)
 - BList::Node, [59](#)
- print

- BEntry, 30
- BEntryList, 35
- printf
 - BFile, 49
 - BString, 121
- printf_t
 - BObject, 92
- Priority
 - BSocket, 107
- PriorityHigh
 - BSocket, 107
- PriorityLow
 - BSocket, 107
- PriorityNormal
 - BSocket, 107
- process
 - BoapServer, 79, 80
 - BoapServerConnection, 82
 - BoapServiceObject, 87
- processEvent
 - BoapServer, 79, 80
 - BoapServiceObject, 87
- pullLine
 - BString, 122
- pullSeparators
 - BString, 122
- pullToken
 - BString, 122
- pullWord
 - BString, 122
- push
 - BList, 57
 - BoapPacket, 74
- pushHead
 - BoapPacket, 74
- queueAdd
 - BList, 57
- queueGet
 - BList, 57
- raw
 - BSocketAddress, 112
- rdLock
 - BRWLock, 102
- read
 - BDir, 27
 - BEntryFile, 33
 - BFile, 49
- readString
 - BFile, 49
 - BUrl, 129
- rear
 - BList, 56
- recv
 - BSocket, 109
- recvFrom
 - BSocket, 109
- recvFromWithTimeout
 - BSocket, 109
- recvWithTimeout
 - BSocket, 109
- refCount
 - BRefData, 97
- removeNL
 - BString, 121
- removeSeparators
 - BString, 122
- reserved
 - BoapPacketHead, 76
- resize
 - BoapPacket, 74
- result
 - BThread, 125
- retDouble
 - BString, 120
- retInt
 - BString, 120
- retStr
 - BString, 120
- retStrDup
 - BString, 120
- roundSize
 - Boap.cpp, 154
 - BoapSimple.cc, 161
- run
 - BoapServer, 79, 80
- running
 - BThread, 125
- seek
 - BFile, 49
- send
 - BSocket, 109
- sendEvent
 - BEventInt, 43
 - BEventPipe, 45
 - BoapServer, 79, 80
 - BoapServiceObject, 87
- sendTo
 - BSocket, 109
- service
 - Boapns::BoapEntry, 68
 - BoapPacketHead, 76
- set
 - BCondBool, 15
 - BError, 38
 - BSocketAddress, 112

- BSocketAddressINET, 114
- setBinary
 - BEvent, 40
 - BEventError, 42
 - BObject, 92
- setBroadCast
 - BSocket, 109
- setConnectionPriority
 - BoapClientObject, 65
- setData
 - BBuffer, 12
 - BoapPacket, 74
- setError
 - BError, 38
- setInitPriority
 - BThread, 125
- setInitStackSize
 - BThread, 125
- setLen
 - BRefData, 97
- setLine
 - BEntry, 30
- setMaxLength
 - BoapClientObject, 66
 - BoapServerConnection, 82
- setName
 - BEntry, 30
 - BoapServiceObject, 87
- setPort
 - BSocketAddressINET, 114
- setPriority
 - BSocket, 109
 - BThread, 125
- setReuseAddress
 - BSocket, 109
- setSize
 - BBuffer, 12
- setSockOpt
 - BSocket, 109
- setSort
 - BDir, 27
- setString
 - BObject, 92
- setTimeout
 - BoapClientObject, 66
- setValue
 - BCondInt, 18
 - BCondValue, 21
 - BCondWrap, 24
 - BEntry, 30
 - BEntryList, 35
- setValueRaw
 - BEntryList, 35
- setVBuf
 - BFile, 49
- setWild
 - BDir, 27
- shutdown
 - BSocket, 109
- signal
 - BCond, 14
- SIZE
 - BBuffer.cpp, 132
- size
 - BBuffer, 13
- SockAddr
 - BSocketAddress, 112
- SockAddrIP
 - BSocketAddressINET, 114
- sort
 - BList, 57
- SortFunc
 - BList, 54
- start
 - BList, 55
 - BThread, 125
 - BTimer, 127
- startFunc
 - BThread, 125
- stop
 - BTimer, 127
- STRBUF
 - BFile.cpp, 146
- strChanged
 - BString, 120
- STREAM
 - BSocket, 107
- STRIP
 - BString.cpp, 179
- subString
 - BString, 121
- swap
 - BList, 57
- swap16
 - Boap.cpp, 154
- swap32
 - Boap.cpp, 154
- swap64
 - Boap.cpp, 154
- swap8
 - Boap.cpp, 154
- TEMPLATE_NEW
 - BNameValue.h, 152
- THREADED
 - BoapServer, 78
- timedWait
 - BCond, 14

- BCondBool, [16](#)
- BCondInt, [18](#)
- BSema, [105](#)
- toLower
 - BString, [121](#)
- toUpper
 - BString, [121](#)
- truncate
 - BString, [121](#)
- tryLock
 - BMutex, [60](#)
- tryNotZeroDecrement
 - BCondInt, [18](#)
- tryRdLock
 - BRWLock, [102](#)
- tryWait
 - BSema, [105](#)
- tryWrLock
 - BRWLock, [103](#)
- Type
 - BError, [38](#)
- type
 - BoapPacketHead, [76](#)
- UInt16
 - BoapSimple.h, [163](#)
 - BTypes.h, [187](#)
- UInt32
 - BoapSimple.h, [163](#)
 - BTypes.h, [187](#)
- UInt64
 - BTypes.h, [187](#)
- UInt8
 - BoapSimple.h, [163](#)
 - BTypes.h, [187](#)
- unlock
 - BMutex, [60](#)
 - BRWLock, [103](#)
- updateLen
 - BoapPacket, [74](#)
- value
 - BCondBool, [16](#)
 - BCondInt, [18](#)
 - BCondValue, [21](#)
 - BCondWrap, [24](#)
- wait
 - BCond, [14](#)
 - BCondBool, [16](#)
 - BCondInt, [18](#)
 - BRtc, [98](#)
 - BRtcThreaded, [100](#)
 - BSema, [104](#)
 - waitForCompletion
 - BThread, [125](#)
 - waitIncrement
 - BCondInt, [18](#)
 - waitLessThan
 - BCondValue, [21](#)
 - BCondWrap, [24](#)
 - waitLessThanOrEqual
 - BCondValue, [21](#)
 - BCondWrap, [24](#)
 - waitMoreThanOrEqual
 - BCondValue, [21](#)
 - BCondWrap, [24](#)
 - waitNotZero
 - BCondInt, [18](#)
 - waitNotZeroDecrement
 - BCondInt, [18](#)
 - wild
 - BDir.cpp, [138](#)
 - wildString
 - BDir.cpp, [138](#)
 - write
 - BEntryFile, [33](#)
 - BFile, [49](#)
 - writeData
 - BBuffer, [12](#)
 - BUrl, [129](#)
 - writeList
 - BEntryFile, [33](#)
 - writeString
 - BFile, [49](#)
 - wrLock
 - BRWLock, [103](#)