

LibTmsApi
2.0.0

Generated by Doxygen 1.8.5

Mon Dec 11 2017 13:24:55

Contents

- 1 Main Page** **1**
 - 1.1 Introduction 1
 - 1.2 Overview 1
 - 1.3 BEAM class library 2
 - 1.4 Examples 3

- 2 Namespace Index** **7**
 - 2.1 Namespace List 7

- 3 Hierarchical Index** **9**
 - 3.1 Class Hierarchy 9

- 4 Class Index** **13**
 - 4.1 Class List 13

- 5 File Index** **17**
 - 5.1 File List 17

- 6 Namespace Documentation** **21**
 - 6.1 Boapns Namespace Reference 21
 - 6.1.1 Variable Documentation 21
 - 6.1.1.1 apiVersion 21
 - 6.2 Tms Namespace Reference 21
 - 6.2.1 Enumeration Type Documentation 23
 - 6.2.1.1 CaptureClock 23
 - 6.2.1.2 CyclePeriod 23
 - 6.2.1.3 DataFunction 24
 - 6.2.1.4 DataType 24
 - 6.2.1.5 Priority 24
 - 6.2.1.6 TestOutput 24
 - 6.2.1.7 TimingSig 24
 - 6.2.2 Variable Documentation 25
 - 6.2.2.1 apiVersion 25

6.2.2.2	tmsNumPickups	25
6.2.2.3	tmsPhaseTableSize	25
7	Class Documentation	27
7.1	BArray< T > Class Template Reference	27
7.1.1	Detailed Description	27
7.1.2	Member Typedef Documentation	28
7.1.2.1	SortFunc	28
7.1.3	Constructor & Destructor Documentation	28
7.1.3.1	BArray	28
7.1.3.2	BArray	28
7.1.3.3	BArray	28
7.1.4	Member Function Documentation	28
7.1.4.1	append	28
7.1.4.2	append	28
7.1.4.3	del	28
7.1.4.4	insert	28
7.1.4.5	number	28
7.1.4.6	rear	28
7.1.4.7	sort	28
7.2	BAtomic< Type > Class Template Reference	28
7.2.1	Detailed Description	29
7.2.2	Constructor & Destructor Documentation	29
7.2.2.1	BAtomic	29
7.2.3	Member Function Documentation	29
7.2.3.1	add	29
7.2.3.2	getValue	29
7.2.3.3	operator Type	29
7.2.3.4	operator++	29
7.2.3.5	operator++	29
7.2.3.6	operator--	29
7.2.3.7	operator--	29
7.2.4	Member Data Documentation	29
7.2.4.1	ovalue	29
7.3	BAtomicCount Class Reference	29
7.3.1	Detailed Description	30
7.3.2	Constructor & Destructor Documentation	30
7.3.2.1	BAtomicCount	30
7.3.3	Member Function Documentation	30
7.3.3.1	add	30

7.3.3.2	getValue	30
7.3.3.3	operator long	30
7.3.3.4	operator++	30
7.3.3.5	operator++	30
7.3.3.6	operator--	30
7.3.3.7	operator--	30
7.3.4	Member Data Documentation	30
7.3.4.1	ovalue	30
7.4	BBuffer Class Reference	30
7.4.1	Constructor & Destructor Documentation	31
7.4.1.1	BBuffer	31
7.4.1.2	~BBuffer	31
7.4.2	Member Function Documentation	31
7.4.2.1	data	31
7.4.2.2	resize	31
7.4.2.3	setData	31
7.4.2.4	setSize	31
7.4.2.5	size	31
7.4.2.6	writeData	32
7.4.3	Member Data Documentation	32
7.4.3.1	odata	32
7.4.3.2	odataSize	32
7.4.3.3	osize	32
7.5	BBufferStore Class Reference	32
7.5.1	Constructor & Destructor Documentation	33
7.5.1.1	BBufferStore	33
7.5.1.2	~BBufferStore	33
7.5.2	Member Function Documentation	33
7.5.2.1	getHexString	33
7.5.2.2	getPos	33
7.5.2.3	pop	33
7.5.2.4	pop	33
7.5.2.5	pop	33
7.5.2.6	pop	33
7.5.2.7	pop	33
7.5.2.8	pop	33
7.5.2.9	pop	33
7.5.2.10	pop	33
7.5.2.11	pop	33
7.5.2.12	pop	33

7.5.2.13	pop	34
7.5.2.14	pop	34
7.5.2.15	pop	34
7.5.2.16	pop	34
7.5.2.17	pop	34
7.5.2.18	push	34
7.5.2.19	push	34
7.5.2.20	push	34
7.5.2.21	push	34
7.5.2.22	push	34
7.5.2.23	push	34
7.5.2.24	push	34
7.5.2.25	push	34
7.5.2.26	push	34
7.5.2.27	push	34
7.5.2.28	push	34
7.5.2.29	push	34
7.5.2.30	push	34
7.5.2.31	push	34
7.5.2.32	push	34
7.5.2.33	setHexString	34
7.5.2.34	setPos	34
7.5.3	Member Data Documentation	34
7.5.3.1	opos	34
7.5.3.2	oswapBytes	34
7.6	BComms Class Reference	35
7.6.1	Member Enumeration Documentation	35
7.6.1.1	Wait	35
7.6.2	Constructor & Destructor Documentation	36
7.6.2.1	BComms	36
7.6.2.2	~BComms	36
7.6.3	Member Function Documentation	36
7.6.3.1	eventQueue	36
7.6.3.2	init	36
7.6.3.3	packetMode	36
7.6.3.4	read	36
7.6.3.5	readAvailable	36
7.6.3.6	setPacketMode	36
7.6.3.7	setTimeout	36
7.6.3.8	wait	36

7.6.3.9	write	36
7.6.3.10	writeAvailable	36
7.6.4	Member Data Documentation	36
7.6.4.1	oevent	36
7.6.4.2	oeventNum	36
7.6.4.3	oeventQueue	36
7.6.4.4	opacketMode	36
7.6.4.5	otimeout	36
7.7	BCond Class Reference	37
7.7.1	Constructor & Destructor Documentation	37
7.7.1.1	BCond	37
7.7.1.2	~BCond	37
7.7.2	Member Function Documentation	37
7.7.2.1	signal	37
7.7.2.2	timedWait	37
7.7.2.3	wait	37
7.7.3	Member Data Documentation	37
7.7.3.1	ocond	37
7.7.3.2	omutex	37
7.8	BCondBool Class Reference	37
7.8.1	Detailed Description	38
7.8.2	Constructor & Destructor Documentation	38
7.8.2.1	BCondBool	38
7.8.2.2	~BCondBool	38
7.8.3	Member Function Documentation	38
7.8.3.1	clear	38
7.8.3.2	operator int	38
7.8.3.3	set	38
7.8.3.4	timedWait	38
7.8.3.5	value	38
7.8.3.6	wait	39
7.8.4	Member Data Documentation	39
7.8.4.1	ocond	39
7.8.4.2	omutex	39
7.8.4.3	ovalue	39
7.9	BCondInt Class Reference	39
7.9.1	Detailed Description	40
7.9.2	Constructor & Destructor Documentation	40
7.9.2.1	BCondInt	40
7.9.2.2	~BCondInt	40

7.9.3	Member Function Documentation	40
7.9.3.1	decrement	40
7.9.3.2	increment	40
7.9.3.3	operator++	40
7.9.3.4	operator+=	40
7.9.3.5	operator--	40
7.9.3.6	operator-=	40
7.9.3.7	setValue	40
7.9.3.8	value	40
7.9.3.9	waitLessThan	41
7.9.3.10	waitLessThanOrEqual	41
7.9.3.11	waitMoreThanOrEqual	41
7.9.4	Member Data Documentation	41
7.9.4.1	ocond	41
7.9.4.2	omutex	41
7.9.4.3	ovalue	41
7.10	BCondResource Class Reference	41
7.10.1	Detailed Description	42
7.10.2	Constructor & Destructor Documentation	42
7.10.2.1	BCondResource	42
7.10.2.2	~BCondResource	42
7.10.3	Member Function Documentation	42
7.10.3.1	end	42
7.10.3.2	inUse	42
7.10.3.3	lock	42
7.10.3.4	locked	42
7.10.3.5	start	42
7.10.3.6	unlock	42
7.10.4	Member Data Documentation	42
7.10.4.1	ocond	42
7.10.4.2	olock	42
7.10.4.3	omutex	42
7.10.4.4	ouse	42
7.11	BCondValue Class Reference	42
7.11.1	Detailed Description	43
7.11.2	Constructor & Destructor Documentation	43
7.11.2.1	BCondValue	43
7.11.2.2	~BCondValue	43
7.11.3	Member Function Documentation	43
7.11.3.1	decrement	43

7.11.3.2	increment	43
7.11.3.3	operator++	44
7.11.3.4	operator+=	44
7.11.3.5	operator--	44
7.11.3.6	operator-=	44
7.11.3.7	setValue	44
7.11.3.8	value	44
7.11.3.9	waitLessThan	44
7.11.3.10	waitLessThanOrEqual	44
7.11.3.11	waitMoreThanOrEqual	44
7.11.4	Member Data Documentation	44
7.11.4.1	ocond	44
7.11.4.2	omutex	44
7.11.4.3	ovalue	44
7.12	BCondWrap Class Reference	45
7.12.1	Constructor & Destructor Documentation	45
7.12.1.1	BCondWrap	45
7.12.1.2	~BCondWrap	45
7.12.2	Member Function Documentation	45
7.12.2.1	decrement	45
7.12.2.2	diff	46
7.12.2.3	increment	46
7.12.2.4	operator++	46
7.12.2.5	operator+=	46
7.12.2.6	operator--	46
7.12.2.7	operator-=	46
7.12.2.8	setValue	46
7.12.2.9	value	46
7.12.2.10	waitLessThan	46
7.12.2.11	waitLessThanOrEqual	46
7.12.2.12	waitMoreThanOrEqual	46
7.12.3	Member Data Documentation	46
7.12.3.1	ocond	46
7.12.3.2	omutex	46
7.12.3.3	ovalue	47
7.13	BConfig Class Reference	47
7.13.1	Detailed Description	47
7.13.2	Member Function Documentation	47
7.13.2.1	close	47
7.13.2.2	fileName	47

7.13.2.3	findValue	47
7.13.2.4	open	48
7.13.2.5	read	48
7.13.2.6	write	48
7.13.3	Member Data Documentation	48
7.13.3.1	ofile	48
7.13.3.2	ofilename	48
7.13.3.3	olock	48
7.14	BDate Class Reference	48
7.14.1	Constructor & Destructor Documentation	49
7.14.1.1	BDate	49
7.14.1.2	BDate	49
7.14.1.3	~BDate	49
7.14.2	Member Function Documentation	49
7.14.2.1	clear	49
7.14.2.2	compare	49
7.14.2.3	day	49
7.14.2.4	daysInMonth	49
7.14.2.5	getDate	49
7.14.2.6	getString	49
7.14.2.7	getStringFormatted	50
7.14.2.8	isLeap	50
7.14.2.9	isSet	50
7.14.2.10	month	50
7.14.2.11	operator BString	50
7.14.2.12	operator!=	50
7.14.2.13	operator<	50
7.14.2.14	operator<=	50
7.14.2.15	operator==	50
7.14.2.16	operator>	50
7.14.2.17	operator>=	50
7.14.2.18	set	50
7.14.2.19	set	50
7.14.2.20	setFirst	50
7.14.2.21	setLast	50
7.14.2.22	setNow	50
7.14.2.23	setString	50
7.14.2.24	setYDay	50
7.14.2.25	yday	50
7.14.2.26	year	50

7.14.3	Member Data Documentation	51
7.14.3.1	oyday	51
7.14.3.2	oyear	51
7.15	BDebugBacktrace Class Reference	51
7.15.1	Constructor & Destructor Documentation	51
7.15.1.1	BDebugBacktrace	51
7.15.1.2	~BDebugBacktrace	51
7.15.2	Member Function Documentation	51
7.15.2.1	dumpBacktrace	51
7.15.2.2	dumpBacktraceFile	51
7.15.2.3	dumpBacktraceStdout	51
7.15.2.4	dumpBacktraceSyslog	51
7.16	BDict< Type > Class Template Reference	52
7.16.1	Member Typedef Documentation	53
7.16.1.1	iterator	53
7.16.2	Constructor & Destructor Documentation	53
7.16.2.1	BDict	53
7.16.2.2	BDict	53
7.16.3	Member Function Documentation	53
7.16.3.1	append	53
7.16.3.2	append	53
7.16.3.3	clear	53
7.16.3.4	del	53
7.16.3.5	del	53
7.16.3.6	find	53
7.16.3.7	hashAdd	53
7.16.3.8	hashDelete	53
7.16.3.9	hashFind	53
7.16.3.10	hashPrint	53
7.16.3.11	hasKey	53
7.16.3.12	insert	53
7.16.3.13	key	53
7.16.3.14	operator+	53
7.16.3.15	operator=	54
7.16.3.16	operator[]	54
7.16.3.17	operator[]	54
7.16.3.18	operator[]	54
7.16.4	Member Data Documentation	54
7.16.4.1	ohashLists	54
7.16.4.2	ohashSize	54

7.17	BDictItem< Type > Class Template Reference	54
7.17.1	Detailed Description	54
7.17.2	Constructor & Destructor Documentation	54
7.17.2.1	BDictItem	54
7.17.3	Member Data Documentation	54
7.17.3.1	key	54
7.17.3.2	value	54
7.18	BDictMap< Value > Class Template Reference	55
7.18.1	Detailed Description	55
7.18.2	Member Typedef Documentation	55
7.18.2.1	iterator	55
7.18.3	Member Function Documentation	55
7.18.3.1	clear	55
7.18.3.2	del	55
7.18.3.3	del	55
7.18.3.4	hasKey	55
7.18.3.5	isEnd	55
7.18.3.6	key	56
7.18.3.7	next	56
7.18.3.8	operator[]	56
7.18.3.9	operator[]	56
7.18.3.10	size	56
7.18.3.11	start	56
7.19	BDir Class Reference	56
7.19.1	Detailed Description	57
7.19.2	Constructor & Destructor Documentation	57
7.19.2.1	BDir	57
7.19.2.2	BDir	57
7.19.2.3	~BDir	57
7.19.3	Member Function Documentation	57
7.19.3.1	clear	57
7.19.3.2	entryName	57
7.19.3.3	entryStat	57
7.19.3.4	entryStat64	57
7.19.3.5	error	57
7.19.3.6	open	57
7.19.3.7	read	57
7.19.3.8	setSort	58
7.19.3.9	setWild	58
7.19.4	Member Data Documentation	58

7.19.4.1	odirname	58
7.19.4.2	oerror	58
7.19.4.3	osort	58
7.19.4.4	owild	58
7.20	BDuration Class Reference	58
7.20.1	Constructor & Destructor Documentation	59
7.20.1.1	BDuration	59
7.20.1.2	BDuration	59
7.20.1.3	~BDuration	59
7.20.2	Member Function Documentation	59
7.20.2.1	addMicroSeconds	59
7.20.2.2	addMilliSeconds	59
7.20.2.3	addSeconds	59
7.20.2.4	clear	59
7.20.2.5	getMicroSeconds	59
7.20.2.6	getSeconds	59
7.20.2.7	getString	59
7.20.2.8	hour	60
7.20.2.9	microSecond	60
7.20.2.10	minute	60
7.20.2.11	second	60
7.20.2.12	set	60
7.20.2.13	setString	60
7.20.3	Member Data Documentation	60
7.20.3.1	ohour	60
7.20.3.2	omicroSecond	60
7.20.3.3	omminute	60
7.20.3.4	osecond	60
7.20.3.5	ospare	60
7.21	BEntry Class Reference	60
7.21.1	Detailed Description	61
7.21.2	Constructor & Destructor Documentation	61
7.21.2.1	BEntry	61
7.21.2.2	BEntry	61
7.21.2.3	BEntry	61
7.21.3	Member Function Documentation	61
7.21.3.1	getName	61
7.21.3.2	getValue	61
7.21.3.3	line	62
7.21.3.4	print	62

7.21.3.5	setLine	62
7.21.3.6	setName	62
7.21.3.7	setValue	62
7.21.4	Member Data Documentation	62
7.21.4.1	oname	62
7.21.4.2	ovalue	62
7.22	BEntryFile Class Reference	62
7.22.1	Detailed Description	63
7.22.2	Constructor & Destructor Documentation	63
7.22.2.1	BEntryFile	63
7.22.2.2	BEntryFile	63
7.22.2.3	~BEntryFile	63
7.22.3	Member Function Documentation	63
7.22.3.1	clear	63
7.22.3.2	filename	63
7.22.3.3	open	63
7.22.3.4	read	64
7.22.3.5	write	64
7.22.3.6	writeList	64
7.22.4	Member Data Documentation	64
7.22.4.1	ocomments	64
7.22.4.2	ofilename	64
7.23	BEntryList Class Reference	64
7.23.1	Detailed Description	65
7.23.2	Constructor & Destructor Documentation	65
7.23.2.1	BEntryList	65
7.23.3	Member Function Documentation	65
7.23.3.1	clear	65
7.23.3.2	del	65
7.23.3.3	deleteEntry	65
7.23.3.4	find	65
7.23.3.5	findValue	66
7.23.3.6	getString	66
7.23.3.7	insert	66
7.23.3.8	isSet	66
7.23.3.9	operator=	66
7.23.3.10	print	66
7.23.3.11	setValue	66
7.23.3.12	setValueRaw	66
7.23.4	Member Data Documentation	66

7.23.4.1	olastPos	66
7.24	BError Class Reference	66
7.24.1	Detailed Description	67
7.24.2	Constructor & Destructor Documentation	67
7.24.2.1	BError	67
7.24.2.2	BError	68
7.24.3	Member Function Documentation	68
7.24.3.1	clear	68
7.24.3.2	copy	68
7.24.3.3	getErrorNo	68
7.24.3.4	getNumber	68
7.24.3.5	getString	68
7.24.3.6	num	68
7.24.3.7	operator int	68
7.24.3.8	set	68
7.24.3.9	setError	68
7.24.3.10	str	68
7.24.4	Member Data Documentation	69
7.24.4.1	oerrNo	69
7.24.4.2	oerrStr	69
7.25	BErrorTime Class Reference	69
7.25.1	Detailed Description	69
7.25.2	Member Enumeration Documentation	70
7.25.2.1	Type	70
7.25.3	Constructor & Destructor Documentation	70
7.25.3.1	BErrorTime	70
7.25.4	Member Function Documentation	70
7.25.4.1	clear	70
7.25.4.2	copy	70
7.25.4.3	getErrorNo	70
7.25.4.4	getString	70
7.25.4.5	getTime	70
7.25.4.6	operator int	70
7.25.4.7	set	70
7.25.5	Member Data Documentation	70
7.25.5.1	oerrNo	70
7.25.5.2	oerrStr	70
7.25.5.3	oerrTime	71
7.26	BEvent Class Reference	71
7.26.1	Constructor & Destructor Documentation	71

7.26.1.1	BEvent	71
7.26.2	Member Function Documentation	71
7.26.2.1	arg	71
7.26.2.2	type	71
7.26.3	Member Data Documentation	71
7.26.3.1	oarg	71
7.26.3.2	otype	71
7.27	BEvent1 Class Reference	72
7.27.1	Detailed Description	72
7.27.2	Constructor & Destructor Documentation	72
7.27.2.1	BEvent1	72
7.27.2.2	~BEvent1	72
7.27.3	Member Function Documentation	72
7.27.3.1	getBinary	72
7.27.3.2	getType	72
7.27.3.3	setBinary	72
7.27.4	Member Data Documentation	73
7.27.4.1	otype	73
7.28	BEvent1Error Class Reference	73
7.28.1	Constructor & Destructor Documentation	73
7.28.1.1	BEvent1Error	73
7.28.2	Member Function Documentation	73
7.28.2.1	getBinary	73
7.28.2.2	setBinary	73
7.29	BEvent1Int Class Reference	73
7.29.1	Detailed Description	74
7.29.2	Constructor & Destructor Documentation	74
7.29.2.1	BEvent1Int	74
7.29.2.2	~BEvent1Int	74
7.29.3	Member Function Documentation	74
7.29.3.1	clear	74
7.29.3.2	getEvent	74
7.29.3.3	getFd	74
7.29.3.4	sendEvent	74
7.29.4	Member Data Documentation	74
7.29.4.1	ofds	74
7.30	BEvent1Pipe Class Reference	75
7.30.1	Detailed Description	75
7.30.2	Constructor & Destructor Documentation	75
7.30.2.1	BEvent1Pipe	75

7.30.2.2	~BEvent1Pipe	75
7.30.3	Member Function Documentation	75
7.30.3.1	clear	75
7.30.3.2	getEvent	75
7.30.3.3	getReceiveFd	76
7.30.3.4	sendEvent	76
7.30.4	Member Data Documentation	76
7.30.4.1	ofds	76
7.31	BEventPipe Class Reference	76
7.31.1	Detailed Description	76
7.31.2	Constructor & Destructor Documentation	77
7.31.2.1	BEventPipe	77
7.31.2.2	~BEventPipe	77
7.31.3	Member Function Documentation	77
7.31.3.1	clear	77
7.31.3.2	getFd	77
7.31.3.3	read	77
7.31.3.4	readAvailable	77
7.31.3.5	write	77
7.31.3.6	writeAvailable	77
7.31.4	Member Data Documentation	77
7.31.4.1	ofds	77
7.32	BFifo< Type > Class Template Reference	77
7.32.1	Constructor & Destructor Documentation	79
7.32.1.1	BFifo	79
7.32.1.2	~BFifo	79
7.32.2	Member Function Documentation	79
7.32.2.1	clear	79
7.32.2.2	operator[]	79
7.32.2.3	read	79
7.32.2.4	read	79
7.32.2.5	readAvailable	79
7.32.2.6	readAvailableChunk	79
7.32.2.7	readData	79
7.32.2.8	readData	79
7.32.2.9	readDone	79
7.32.2.10	readPos	79
7.32.2.11	resize	79
7.32.2.12	size	79
7.32.2.13	write	80

7.32.2.14	write	80
7.32.2.15	writeAvailable	80
7.32.2.16	writeAvailableChunk	80
7.32.2.17	writeBackup	80
7.32.2.18	writeData	80
7.32.2.19	writeData	80
7.32.2.20	writeDone	80
7.32.3	Member Data Documentation	80
7.32.3.1	odata	80
7.32.3.2	olock	80
7.32.3.3	oreadPos	80
7.32.3.4	osize	80
7.32.3.5	owritePos	81
7.33	BFifoCirc< Type > Class Template Reference	81
7.33.1	Detailed Description	82
7.33.2	Member Enumeration Documentation	82
7.33.2.1	anonymous enum	82
7.33.3	Constructor & Destructor Documentation	82
7.33.3.1	BFifoCirc	82
7.33.3.2	~BFifoCirc	82
7.33.4	Member Function Documentation	82
7.33.4.1	clear	82
7.33.4.2	mapCircularBuffer	82
7.33.4.3	operator[]	82
7.33.4.4	read	82
7.33.4.5	readAvailable	82
7.33.4.6	readData	83
7.33.4.7	readDone	83
7.33.4.8	readWaitAvailable	83
7.33.4.9	size	83
7.33.4.10	unmapCircularBuffer	83
7.33.4.11	write	83
7.33.4.12	writeAvailable	83
7.33.4.13	writeData	83
7.33.4.14	writeDone	83
7.33.4.15	writeWaitAvailable	83
7.33.5	Member Data Documentation	83
7.33.5.1	odata	83
7.33.5.2	olock	83
7.33.5.3	oreadPos	83

7.33.5.4	osize	84
7.33.5.5	ovmSize	84
7.33.5.6	owriteNumFifoSamples	84
7.33.5.7	owritePos	84
7.34	BFifoCircPos Class Reference	84
7.34.1	Detailed Description	84
7.34.2	Constructor & Destructor Documentation	85
7.34.2.1	BFifoCircPos	85
7.34.3	Member Function Documentation	85
7.34.3.1	difference	85
7.34.3.2	increment	85
7.34.3.3	operator int	85
7.34.3.4	operator!=	85
7.34.3.5	operator+=	85
7.34.3.6	operator==	85
7.34.3.7	pos	85
7.34.3.8	set	85
7.34.3.9	setSize	85
7.34.4	Member Data Documentation	85
7.34.4.1	opos	85
7.34.4.2	osize	85
7.35	BFile Class Reference	85
7.35.1	Detailed Description	87
7.35.2	Constructor & Destructor Documentation	87
7.35.2.1	BFile	87
7.35.2.2	BFile	87
7.35.2.3	~BFile	87
7.35.3	Member Function Documentation	87
7.35.3.1	close	87
7.35.3.2	fgets	87
7.35.3.3	fileName	87
7.35.3.4	flush	87
7.35.3.5	getFd	87
7.35.3.6	isEnd	87
7.35.3.7	isOpen	87
7.35.3.8	length	87
7.35.3.9	open	88
7.35.3.10	open	88
7.35.3.11	open	88
7.35.3.12	operator=	88

7.35.3.13	position	88
7.35.3.14	printf	88
7.35.3.15	read	88
7.35.3.16	readString	88
7.35.3.17	seek	88
7.35.3.18	setVBuf	88
7.35.3.19	truncate	88
7.35.3.20	write	88
7.35.3.21	writeString	88
7.35.4	Member Data Documentation	89
7.35.4.1	ofile	89
7.35.4.2	ofilename	89
7.35.4.3	omode	89
7.36	BFileCsv Class Reference	89
7.36.1	Constructor & Destructor Documentation	89
7.36.1.1	BFileCsv	89
7.36.2	Member Function Documentation	89
7.36.2.1	readCsv	89
7.36.2.2	writeCsv	89
7.36.3	Member Data Documentation	89
7.36.3.1	oseparator	89
7.37	BFileData Class Reference	90
7.37.1	Member Function Documentation	90
7.37.1.1	del	90
7.37.1.2	find	90
7.37.1.3	getNextId	90
7.37.1.4	open	90
7.37.1.5	read	90
7.37.1.6	write	90
7.37.1.7	write	90
7.37.2	Member Data Documentation	90
7.37.2.1	ofilename	90
7.38	BIter Class Reference	91
7.38.1	Detailed Description	91
7.38.2	Constructor & Destructor Documentation	91
7.38.2.1	BIter	91
7.38.3	Member Function Documentation	91
7.38.3.1	operator BNode *	91
7.38.3.2	operator==	91
7.38.3.3	valid	91

7.38.4	Member Data Documentation	91
7.38.4.1	oi	91
7.39	BList< T > Class Template Reference	91
7.39.1	Detailed Description	94
7.39.2	Member Typedef Documentation	94
7.39.2.1	SortFunc	94
7.39.3	Constructor & Destructor Documentation	94
7.39.3.1	BList	94
7.39.3.2	BList	94
7.39.3.3	~BList	94
7.39.4	Member Function Documentation	94
7.39.4.1	append	94
7.39.4.2	append	94
7.39.4.3	begin	94
7.39.4.4	clear	94
7.39.4.5	del	94
7.39.4.6	deleteFirst	95
7.39.4.7	deleteLast	95
7.39.4.8	end	95
7.39.4.9	end	95
7.39.4.10	front	95
7.39.4.11	get	95
7.39.4.12	get	95
7.39.4.13	goTo	95
7.39.4.14	has	95
7.39.4.15	insert	95
7.39.4.16	insertAfter	95
7.39.4.17	isEnd	95
7.39.4.18	next	96
7.39.4.19	nodeCreate	96
7.39.4.20	nodeCreate	96
7.39.4.21	nodeGet	96
7.39.4.22	nodeGet	96
7.39.4.23	number	96
7.39.4.24	operator+	96
7.39.4.25	operator=	96
7.39.4.26	operator[]	96
7.39.4.27	operator[]	96
7.39.4.28	operator[]	96
7.39.4.29	operator[]	96

7.39.4.30 pop	96
7.39.4.31 position	96
7.39.4.32 prev	96
7.39.4.33 push	96
7.39.4.34 queueAdd	96
7.39.4.35 queueGet	97
7.39.4.36 rear	97
7.39.4.37 size	97
7.39.4.38 sort	97
7.39.4.39 sort	97
7.39.4.40 start	97
7.39.4.41 swap	97
7.39.5 Member Data Documentation	97
7.39.5.1 olength	97
7.39.5.2 onodes	97
7.40 BMutex Class Reference	97
7.40.1 Detailed Description	98
7.40.2 Member Enumeration Documentation	98
7.40.2.1 Type	98
7.40.3 Constructor & Destructor Documentation	98
7.40.3.1 BMutex	98
7.40.3.2 BMutex	98
7.40.3.3 ~BMutex	98
7.40.4 Member Function Documentation	98
7.40.4.1 lock	98
7.40.4.2 operator=	99
7.40.4.3 timedLock	99
7.40.4.4 tryLock	99
7.40.4.5 unlock	99
7.40.5 Member Data Documentation	99
7.40.5.1 omutex	99
7.41 BMutexLock Class Reference	99
7.41.1 Constructor & Destructor Documentation	99
7.41.1.1 BMutexLock	99
7.41.1.2 ~BMutexLock	99
7.41.2 Member Function Documentation	99
7.41.2.1 lock	99
7.41.2.2 unlock	99
7.41.3 Member Data Documentation	100
7.41.3.1 olock	100

7.42 Bmysql Class Reference	100
7.42.1 Constructor & Destructor Documentation	100
7.42.1.1 Bmysql	100
7.42.1.2 ~Bmysql	100
7.42.2 Member Function Documentation	100
7.42.2.1 close	100
7.42.2.2 db	100
7.42.2.3 del	100
7.42.2.4 escapeString	101
7.42.2.5 flush	101
7.42.2.6 get	101
7.42.2.7 insert	101
7.42.2.8 open	101
7.42.2.9 query	101
7.42.2.10 setDebug	101
7.42.2.11 update	101
7.42.3 Member Data Documentation	101
7.42.3.1 odb	101
7.42.3.2 odebug	101
7.42.3.3 olock	101
7.42.3.4 oopened	101
7.43 BNameValue< T > Class Template Reference	101
7.43.1 Constructor & Destructor Documentation	102
7.43.1.1 BNameValue	102
7.43.1.2 BNameValue	102
7.43.2 Member Function Documentation	102
7.43.2.1 getName	102
7.43.2.2 getValue	102
7.43.3 Member Data Documentation	102
7.43.3.1 oname	102
7.43.3.2 ovalue	102
7.44 BNameValueList< T > Class Template Reference	102
7.44.1 Member Function Documentation	102
7.44.1.1 find	102
7.44.1.2 findPos	102
7.45 BNode Class Reference	103
7.45.1 Constructor & Destructor Documentation	103
7.45.1.1 BNode	103
7.45.2 Member Data Documentation	103
7.45.2.1 next	103

7.45.2.2	prev	103
7.46	BoapClientObject Class Reference	103
7.46.1	Constructor & Destructor Documentation	105
7.46.1.1	BoapClientObject	105
7.46.1.2	~BoapClientObject	105
7.46.1.3	BoapClientObject	105
7.46.2	Member Function Documentation	105
7.46.2.1	checkApiVersion	105
7.46.2.2	connectService	105
7.46.2.3	connectService	105
7.46.2.4	disconnectService	105
7.46.2.5	getServiceName	105
7.46.2.6	handleReconnect	105
7.46.2.7	performCall	105
7.46.2.8	performCall	105
7.46.2.9	performRecv	105
7.46.2.10	performRecv	105
7.46.2.11	performSend	105
7.46.2.12	performSend	105
7.46.2.13	ping	106
7.46.2.14	pingLocked	106
7.46.2.15	setConnectionPriority	106
7.46.2.16	setMaxLength	106
7.46.2.17	setTimeout	106
7.46.3	Member Data Documentation	106
7.46.3.1	oapiVersion	106
7.46.3.2	oconnected	106
7.46.3.3	olock	106
7.46.3.4	omaxLength	106
7.46.3.5	oname	106
7.46.3.6	opriority	106
7.46.3.7	oreconnect	106
7.46.3.8	orx	106
7.46.3.9	oservice	106
7.46.3.10	otimeout	106
7.46.3.11	otx	106
7.47	Boapns::BoapEntry Class Reference	107
7.47.1	Constructor & Destructor Documentation	107
7.47.1.1	BoapEntry	107
7.47.2	Member Data Documentation	107

7.47.2.1	addressList	107
7.47.2.2	hostName	107
7.47.2.3	name	107
7.47.2.4	port	107
7.47.2.5	service	107
7.48	BoapFuncEntry Class Reference	107
7.48.1	Constructor & Destructor Documentation	108
7.48.1.1	BoapFuncEntry	108
7.48.1.2	BoapFuncEntry	108
7.48.2	Member Data Documentation	108
7.48.2.1	ocmd	108
7.48.2.2	ocmd	108
7.48.2.3	ofunc	108
7.49	BoapMcClientObject Class Reference	108
7.49.1	Constructor & Destructor Documentation	109
7.49.1.1	BoapMcClientObject	109
7.49.1.2	~BoapMcClientObject	109
7.49.2	Member Function Documentation	109
7.49.2.1	getApiVersion	109
7.49.2.2	performCall	109
7.49.2.3	performRecv	109
7.49.2.4	performSend	109
7.49.2.5	setAddress	109
7.49.3	Member Data Documentation	109
7.49.3.1	oaddressFrom	109
7.49.3.2	oaddressTo	109
7.49.3.3	oapiVersion	109
7.49.3.4	ocomms	109
7.49.3.5	opacket	109
7.50	BoapMcComms Class Reference	110
7.50.1	Constructor & Destructor Documentation	111
7.50.1.1	BoapMcComms	111
7.50.1.2	~BoapMcComms	111
7.50.2	Member Function Documentation	111
7.50.2.1	getApiVersion	111
7.50.2.2	packetRecv	111
7.50.2.3	packetSend	111
7.50.2.4	performCall	111
7.50.2.5	performSend	112
7.50.2.6	processPacket	112

7.50.2.7	processRequest	112
7.50.2.8	processRequests	112
7.50.2.9	processRx	112
7.50.2.10	setAddress	112
7.50.2.11	setComms	112
7.50.2.12	setComms	112
7.50.2.13	setCommsMode	112
7.50.2.14	setTimeout	112
7.50.3	Member Data Documentation	112
7.50.3.1	oaddressFrom	112
7.50.3.2	oaddressTo	112
7.50.3.3	oapiVersion	112
7.50.3.4	ocomms	113
7.50.3.5	oclockCall	113
7.50.3.6	oclockTx	113
7.50.3.7	opacket	113
7.50.3.8	opacketReqQueue	113
7.50.3.9	opacketReqRx	113
7.50.3.10	opacketReqTx	113
7.50.3.11	opacketRx	113
7.50.3.12	opacketRxSema	113
7.50.3.13	opacketTx	113
7.50.3.14	opacketTxQueue	113
7.50.3.15	opacketTxQueueWriteNum	113
7.50.3.16	opacketTxSema	113
7.50.3.17	oslave	114
7.50.3.18	othreaded	114
7.50.3.19	otimeout	114
7.51	BoapMcPacket Class Reference	114
7.51.1	Member Data Documentation	114
7.51.1.1	data	114
7.51.1.2	head	114
7.52	BoapMcPacketHead Struct Reference	114
7.52.1	Member Data Documentation	115
7.52.1.1	addressFrom	115
7.52.1.2	addressTo	115
7.52.1.3	checksum	115
7.52.1.4	cmd	115
7.52.1.5	error	115
7.52.1.6	length	115

7.53	BoapMcServiceObject Class Reference	115
7.53.1	Constructor & Destructor Documentation	115
7.53.1.1	BoapMcServiceObject	115
7.53.1.2	~BoapMcServiceObject	115
7.53.2	Member Function Documentation	115
7.53.2.1	process	115
7.53.2.2	processEvent	115
7.53.2.3	sendEvent	115
7.53.3	Member Data Documentation	116
7.53.3.1	oapiVersion	116
7.54	BoapMcSignalObject Class Reference	116
7.54.1	Constructor & Destructor Documentation	116
7.54.1.1	BoapMcSignalObject	116
7.54.2	Member Function Documentation	116
7.54.2.1	performSend	116
7.54.3	Member Data Documentation	116
7.54.3.1	ocomms	116
7.55	Boapns::Boapns Class Reference	116
7.55.1	Constructor & Destructor Documentation	117
7.55.1.1	Boapns	117
7.55.2	Member Function Documentation	117
7.55.2.1	addEntry	117
7.55.2.2	delEntry	117
7.55.2.3	getEntry	117
7.55.2.4	getEntryList	117
7.55.2.5	getNewName	117
7.55.2.6	getVersion	117
7.56	BoapPacket Class Reference	117
7.56.1	Constructor & Destructor Documentation	119
7.56.1.1	BoapPacket	119
7.56.1.2	~BoapPacket	119
7.56.1.3	BoapPacket	119
7.56.1.4	~BoapPacket	119
7.56.2	Member Function Documentation	119
7.56.2.1	data	119
7.56.2.2	getCmd	119
7.56.2.3	nbytes	119
7.56.2.4	peekHead	119
7.56.2.5	pop	119
7.56.2.6	pop	119

7.56.2.7	pop	119
7.56.2.8	pop	119
7.56.2.9	pop	119
7.56.2.10	pop	119
7.56.2.11	pop	119
7.56.2.12	pop	119
7.56.2.13	pop	119
7.56.2.14	pop	119
7.56.2.15	popHead	119
7.56.2.16	popHead	119
7.56.2.17	push	119
7.56.2.18	push	120
7.56.2.19	push	120
7.56.2.20	push	120
7.56.2.21	push	120
7.56.2.22	push	120
7.56.2.23	push	120
7.56.2.24	push	120
7.56.2.25	push	120
7.56.2.26	push	120
7.56.2.27	pushHead	120
7.56.2.28	pushHead	120
7.56.2.29	resize	120
7.56.2.30	setData	120
7.56.2.31	updateHead	120
7.56.2.32	updateLen	120
7.56.3	Member Data Documentation	120
7.56.3.1	odata	120
7.56.3.2	onbytes	120
7.56.3.3	opos	120
7.56.3.4	osize	120
7.57	BoapPacketHead Struct Reference	120
7.57.1	Member Data Documentation	121
7.57.1.1	cmd	121
7.57.1.2	cmd	121
7.57.1.3	length	121
7.57.1.4	length	121
7.57.1.5	reserved	121
7.57.1.6	service	121
7.57.1.7	service	121

7.57.1.8	type	121
7.57.1.9	type	121
7.58	BoapServer Class Reference	121
7.58.1	Member Enumeration Documentation	123
7.58.1.1	anonymous enum	123
7.58.2	Constructor & Destructor Documentation	123
7.58.2.1	BoapServer	123
7.58.2.2	~BoapServer	123
7.58.2.3	BoapServer	123
7.58.3	Member Function Documentation	123
7.58.3.1	addObject	123
7.58.3.2	addObject	123
7.58.3.3	clientGone	123
7.58.3.4	function	123
7.58.3.5	getConnectionsNumber	123
7.58.3.6	getEventSocket	123
7.58.3.7	getEventSocket	123
7.58.3.8	getHostName	123
7.58.3.9	getHostName	123
7.58.3.10	getSocket	123
7.58.3.11	getSocket	123
7.58.3.12	init	123
7.58.3.13	init	123
7.58.3.14	newConnection	123
7.58.3.15	process	123
7.58.3.16	process	123
7.58.3.17	processEvent	124
7.58.3.18	processEvent	124
7.58.3.19	processEvent	124
7.58.3.20	processEvent	124
7.58.3.21	run	124
7.58.3.22	run	124
7.58.3.23	sendEvent	124
7.58.3.24	sendEvent	124
7.58.4	Member Data Documentation	124
7.58.4.1	oboapNs	124
7.58.4.2	oboapns	124
7.58.4.3	oclientGoneEvent	124
7.58.4.4	oclients	124
7.58.4.5	ohostName	124

7.58.4.6	oisBoapns	124
7.58.4.7	onet	124
7.58.4.8	onetEvent	124
7.58.4.9	onetEventAddress	124
7.58.4.10	onumOperations	124
7.58.4.11	opoll	124
7.58.4.12	orx	124
7.58.4.13	oservices	124
7.58.4.14	othreaded	124
7.58.4.15	otx	124
7.59	BoapServerConnection Class Reference	125
7.59.1	Constructor & Destructor Documentation	125
7.59.1.1	BoapServerConnection	125
7.59.1.2	~BoapServerConnection	125
7.59.2	Member Function Documentation	125
7.59.2.1	function	125
7.59.2.2	getHead	126
7.59.2.3	getSocket	126
7.59.2.4	init	126
7.59.2.5	process	126
7.59.2.6	setMaxLength	126
7.59.2.7	validate	126
7.59.3	Member Data Documentation	126
7.59.3.1	oboapServer	126
7.59.3.2	omaxLength	126
7.59.3.3	orx	126
7.59.3.4	osocket	126
7.59.3.5	otx	126
7.60	BoapServiceEntry Class Reference	126
7.60.1	Constructor & Destructor Documentation	126
7.60.1.1	BoapServiceEntry	126
7.60.1.2	BoapServiceEntry	127
7.60.2	Member Data Documentation	127
7.60.2.1	oobject	127
7.60.2.2	oservice	127
7.61	BoapServiceObject Class Reference	127
7.61.1	Constructor & Destructor Documentation	128
7.61.1.1	BoapServiceObject	128
7.61.1.2	~BoapServiceObject	128
7.61.1.3	BoapServiceObject	128

7.61.1.4	~BoapServiceObject	128
7.61.2	Member Function Documentation	128
7.61.2.1	doConnectionPriority	128
7.61.2.2	doPing	128
7.61.2.3	name	128
7.61.2.4	name	128
7.61.2.5	process	128
7.61.2.6	process	128
7.61.2.7	processEvent	128
7.61.2.8	processEvent	128
7.61.2.9	processEvent	128
7.61.2.10	processEvent	128
7.61.2.11	sendEvent	128
7.61.2.12	sendEvent	128
7.61.2.13	sendEvent	128
7.61.2.14	sendEvent	128
7.61.2.15	setName	128
7.61.3	Member Data Documentation	128
7.61.3.1	oapiVersion	128
7.61.3.2	ofuncList	128
7.61.3.3	oname	128
7.61.3.4	oserver	128
7.62	BoapSignalObject Class Reference	129
7.62.1	Constructor & Destructor Documentation	129
7.62.1.1	BoapSignalObject	129
7.62.1.2	BoapSignalObject	129
7.62.2	Member Function Documentation	129
7.62.2.1	performSend	129
7.62.2.2	performSend	129
7.62.3	Member Data Documentation	129
7.62.3.1	orx	129
7.62.3.2	otx	130
7.63	BObj Class Reference	130
7.63.1	Constructor & Destructor Documentation	130
7.63.1.1	BObj	130
7.63.1.2	~BObj	130
7.63.2	Member Function Documentation	130
7.63.2.1	getDebugString	130
7.63.2.2	getMember	130
7.63.2.3	getMembers	130

7.63.2.4	getMembers	130
7.63.2.5	getType	130
7.63.2.6	membersPrint	130
7.63.2.7	setMember	131
7.63.2.8	setMembers	131
7.64	BObjMember Struct Reference	131
7.64.1	Member Data Documentation	131
7.64.1.1	dataOffset	131
7.64.1.2	name	131
7.64.1.3	size	131
7.64.1.4	type	131
7.64.1.5	typeComp	131
7.64.1.6	typeName	131
7.65	BPoll Class Reference	131
7.65.1	Detailed Description	132
7.65.2	Member Typedef Documentation	132
7.65.2.1	PollFd	132
7.65.3	Constructor & Destructor Documentation	132
7.65.3.1	BPoll	132
7.65.3.2	~BPoll	132
7.65.4	Member Function Documentation	132
7.65.4.1	append	132
7.65.4.2	clear	133
7.65.4.3	delFd	133
7.65.4.4	doPoll	133
7.65.4.5	doPollEvents	133
7.65.4.6	getPollFds	133
7.65.4.7	getPollFdsNum	133
7.65.4.8	nextFd	133
7.65.5	Member Data Documentation	133
7.65.5.1	ofds	133
7.65.5.2	ofdsNext	133
7.65.5.3	ofdsNum	133
7.66	BQueue< T > Class Template Reference	133
7.66.1	Detailed Description	134
7.66.2	Constructor & Destructor Documentation	134
7.66.2.1	BQueue	134
7.66.2.2	~BQueue	134
7.66.3	Member Function Documentation	134
7.66.3.1	clear	134

7.66.3.2	read	134
7.66.3.3	readAvailable	134
7.66.3.4	write	134
7.66.3.5	writeAvailable	135
7.66.4	Member Data Documentation	135
7.66.4.1	olock	135
7.66.4.2	onumber	135
7.66.4.3	osize	135
7.67	BRefData Class Reference	135
7.67.1	Detailed Description	135
7.67.2	Constructor & Destructor Documentation	136
7.67.2.1	BRefData	136
7.67.2.2	BRefData	136
7.67.2.3	BRefData	136
7.67.2.4	~BRefData	136
7.67.3	Member Function Documentation	136
7.67.3.1	addRef	136
7.67.3.2	copy	136
7.67.3.3	data	136
7.67.3.4	deleteRef	136
7.67.3.5	len	136
7.67.3.6	operator=	136
7.67.3.7	setLen	136
7.67.4	Member Data Documentation	136
7.67.4.1	odata	136
7.67.4.2	olen	136
7.67.4.3	orefCount	137
7.68	BRtc Class Reference	137
7.68.1	Detailed Description	137
7.68.2	Constructor & Destructor Documentation	137
7.68.2.1	BRtc	137
7.68.2.2	~BRtc	137
7.68.3	Member Function Documentation	137
7.68.3.1	init	137
7.68.3.2	wait	137
7.68.4	Member Data Documentation	138
7.68.4.1	ofd	138
7.68.4.2	orate	138
7.69	BRtcThreaded Class Reference	138
7.69.1	Detailed Description	138

7.69.2	Constructor & Destructor Documentation	139
7.69.2.1	BRtcThreaded	139
7.69.2.2	~BRtcThreaded	139
7.69.3	Member Function Documentation	139
7.69.3.1	function	139
7.69.3.2	init	139
7.69.3.3	wait	139
7.69.4	Member Data Documentation	139
7.69.4.1	ocond	139
7.69.4.2	orate	139
7.69.4.3	ortc	139
7.70	BRWLock Class Reference	139
7.70.1	Detailed Description	140
7.70.2	Constructor & Destructor Documentation	140
7.70.2.1	BRWLock	140
7.70.2.2	BRWLock	140
7.70.2.3	~BRWLock	140
7.70.3	Member Function Documentation	140
7.70.3.1	operator=	140
7.70.3.2	rdLock	140
7.70.3.3	tryRdLock	140
7.70.3.4	tryWrLock	140
7.70.3.5	unlock	140
7.70.3.6	wrLock	140
7.70.4	Member Data Documentation	140
7.70.4.1	olock	140
7.71	BSema Class Reference	141
7.71.1	Detailed Description	141
7.71.2	Constructor & Destructor Documentation	141
7.71.2.1	BSema	141
7.71.2.2	BSema	141
7.71.2.3	~BSema	141
7.71.3	Member Function Documentation	141
7.71.3.1	getValue	141
7.71.3.2	operator=	141
7.71.3.3	post	141
7.71.3.4	timedWait	142
7.71.3.5	tryWait	142
7.71.3.6	wait	142
7.71.4	Member Data Documentation	142

7.71.4.1	osema	142
7.72	BSemaphore Class Reference	142
7.72.1	Detailed Description	142
7.72.2	Constructor & Destructor Documentation	143
7.72.2.1	BSemaphore	143
7.72.2.2	BSemaphore	143
7.72.2.3	~BSemaphore	143
7.72.3	Member Function Documentation	143
7.72.3.1	getValue	143
7.72.3.2	operator=	143
7.72.3.3	set	143
7.72.3.4	wait	143
7.72.4	Member Data Documentation	143
7.72.4.1	osema	143
7.73	BSemaphoreCount Class Reference	143
7.73.1	Constructor & Destructor Documentation	144
7.73.1.1	BSemaphoreCount	144
7.73.1.2	BSemaphoreCount	144
7.73.1.3	~BSemaphoreCount	144
7.73.2	Member Function Documentation	144
7.73.2.1	add	144
7.73.2.2	operator=	144
7.73.2.3	setValue	144
7.73.2.4	take	144
7.73.2.5	value	144
7.73.2.6	wait	144
7.73.3	Member Data Documentation	144
7.73.3.1	olock	144
7.73.3.2	osema	144
7.73.3.3	ovalue	144
7.74	BSignal Class Reference	144
7.74.1	Member Enumeration Documentation	145
7.74.1.1	anonymous enum	145
7.74.2	Constructor & Destructor Documentation	145
7.74.2.1	BSignal	145
7.74.2.2	BSignal	145
7.74.2.3	~BSignal	145
7.74.3	Member Function Documentation	145
7.74.3.1	operator=	145
7.74.4	Member Data Documentation	145

7.74.4.1	data	145
7.74.4.2	id	145
7.74.4.3	nextId	145
7.74.4.4	numRepeat	145
7.74.4.5	numSamples	145
7.75	BSocket Class Reference	146
7.75.1	Member Enumeration Documentation	147
7.75.1.1	NType	147
7.75.1.2	Priority	147
7.75.2	Constructor & Destructor Documentation	147
7.75.2.1	BSocket	147
7.75.2.2	BSocket	147
7.75.2.3	BSocket	147
7.75.2.4	BSocket	147
7.75.2.5	~BSocket	147
7.75.3	Member Function Documentation	147
7.75.3.1	accept	147
7.75.3.2	accept	147
7.75.3.3	bind	147
7.75.3.4	close	147
7.75.3.5	connect	148
7.75.3.6	getAddress	148
7.75.3.7	getFd	148
7.75.3.8	getMTU	148
7.75.3.9	getSockOpt	148
7.75.3.10	init	148
7.75.3.11	init	148
7.75.3.12	listen	148
7.75.3.13	recv	148
7.75.3.14	recvFrom	148
7.75.3.15	recvFromWithTimeout	148
7.75.3.16	recvWithTimeout	148
7.75.3.17	send	148
7.75.3.18	sendTo	148
7.75.3.19	setBroadCast	148
7.75.3.20	setFd	148
7.75.3.21	setPriority	148
7.75.3.22	setReuseAddress	148
7.75.3.23	setSockOpt	148
7.75.3.24	shutdown	148

7.75.4	Member Data Documentation	148
7.75.4.1	osocket	148
7.76	BSocketAddress Class Reference	148
7.76.1	Detailed Description	149
7.76.2	Member Typedef Documentation	149
7.76.2.1	SockAddr	149
7.76.3	Constructor & Destructor Documentation	149
7.76.3.1	BSocketAddress	149
7.76.3.2	BSocketAddress	149
7.76.3.3	BSocketAddress	149
7.76.3.4	~BSocketAddress	149
7.76.4	Member Function Documentation	149
7.76.4.1	len	150
7.76.4.2	operator const SockAddr *	150
7.76.4.3	operator!=	150
7.76.4.4	operator=	150
7.76.4.5	operator==	150
7.76.4.6	raw	150
7.76.4.7	set	150
7.76.5	Member Data Documentation	150
7.76.5.1	oaddress	150
7.76.5.2	olen	150
7.77	BSocketAddressINET Class Reference	150
7.77.1	Detailed Description	151
7.77.2	Member Typedef Documentation	151
7.77.2.1	SockAddrIP	151
7.77.3	Member Function Documentation	151
7.77.3.1	address	151
7.77.3.2	getHostName	151
7.77.3.3	getIpAddresses	151
7.77.3.4	getIpAddressList	151
7.77.3.5	getIpAddressListAll	151
7.77.3.6	getString	151
7.77.3.7	port	152
7.77.3.8	set	152
7.77.3.9	set	152
7.77.3.10	set	152
7.77.3.11	setPort	152
7.78	BSpi Class Reference	152
7.78.1	Detailed Description	152

7.78.2	Member Enumeration Documentation	152
7.78.2.1	Mode	152
7.78.3	Constructor & Destructor Documentation	153
7.78.3.1	BSpi	153
7.78.4	Member Function Documentation	153
7.78.4.1	init	153
7.78.4.2	transact	153
7.78.5	Member Data Documentation	153
7.78.5.1	odev	153
7.78.5.2	odevName	153
7.79	BString Class Reference	153
7.79.1	Constructor & Destructor Documentation	156
7.79.1.1	BString	156
7.79.1.2	BString	156
7.79.1.3	BString	156
7.79.1.4	BString	156
7.79.1.5	BString	156
7.79.1.6	BString	156
7.79.1.7	BString	156
7.79.1.8	BString	156
7.79.1.9	BString	156
7.79.1.10	~BString	156
7.79.2	Member Function Documentation	156
7.79.2.1	add	156
7.79.2.2	append	157
7.79.2.3	base64Decode	157
7.79.2.4	base64Encode	157
7.79.2.5	basename	157
7.79.2.6	clear	157
7.79.2.7	compare	157
7.79.2.8	compareRegex	157
7.79.2.9	compareWild	157
7.79.2.10	compareWildExpression	157
7.79.2.11	convert	157
7.79.2.12	convert	157
7.79.2.13	convert	157
7.79.2.14	convert	157
7.79.2.15	convert	158
7.79.2.16	convertHex	158
7.79.2.17	convertHex	158

7.79.2.18 copy	158
7.79.2.19 csvDecode	158
7.79.2.20 csvEncode	158
7.79.2.21 del	158
7.79.2.22 dirname	158
7.79.2.23 extension	158
7.79.2.24 field	158
7.79.2.25 fields	158
7.79.2.26 find	158
7.79.2.27 find	158
7.79.2.28 findReverse	158
7.79.2.29 firstLine	158
7.79.2.30 fixedLen	159
7.79.2.31 get	159
7.79.2.32 get	159
7.79.2.33 getTokenList	159
7.79.2.34 getTokenList	159
7.79.2.35 hash	159
7.79.2.36 init	159
7.79.2.37 insert	159
7.79.2.38 inString	159
7.79.2.39 isSpace	159
7.79.2.40 justify	159
7.79.2.41 len	159
7.79.2.42 lowerFirst	159
7.79.2.43 operator const char *	159
7.79.2.44 operator!=	159
7.79.2.45 operator!=	159
7.79.2.46 operator+	159
7.79.2.47 operator+	159
7.79.2.48 operator+	159
7.79.2.49 operator+	160
7.79.2.50 operator+	160
7.79.2.51 operator+	160
7.79.2.52 operator+=	160
7.79.2.53 operator+=	160
7.79.2.54 operator<	160
7.79.2.55 operator<	160
7.79.2.56 operator<=	160
7.79.2.57 operator=	160

7.79.2.58 operator==	160
7.79.2.59 operator==	160
7.79.2.60 operator>	160
7.79.2.61 operator>	160
7.79.2.62 operator>=	160
7.79.2.63 operator[]	160
7.79.2.64 pad	160
7.79.2.65 printf	160
7.79.2.66 pullLine	160
7.79.2.67 pullSeparators	160
7.79.2.68 pullToken	160
7.79.2.69 pullWord	160
7.79.2.70 removeNL	161
7.79.2.71 removeSeparators	161
7.79.2.72 retDouble	161
7.79.2.73 retInt	161
7.79.2.74 retStr	161
7.79.2.75 retStrDup	161
7.79.2.76 retUInt	161
7.79.2.77 reverse	161
7.79.2.78 split	161
7.79.2.79 subString	161
7.79.2.80 toLower	161
7.79.2.81 toUpper	161
7.79.2.82 translateChar	162
7.79.2.83 truncate	162
7.79.3 Member Data Documentation	162
7.79.3.1 ostr	162
7.80 BStringLocked Class Reference	162
7.80.1 Constructor & Destructor Documentation	162
7.80.1.1 BStringLocked	162
7.80.1.2 BStringLocked	162
7.80.1.3 BStringLocked	162
7.80.2 Member Function Documentation	162
7.80.2.1 len	162
7.80.2.2 operator BString	163
7.80.2.3 operator+	163
7.80.2.4 operator=	163
7.80.3 Member Data Documentation	163
7.80.3.1 olock	163

7.80.3.2	ostr	163
7.81	BStringMutex Class Reference	163
7.81.1	Constructor & Destructor Documentation	163
7.81.1.1	BStringMutex	163
7.82	BTable Class Reference	163
7.82.1	Constructor & Destructor Documentation	164
7.82.1.1	BTable	164
7.82.1.2	~BTable	164
7.82.2	Member Function Documentation	164
7.82.2.1	addRow	164
7.82.2.2	calculateWidths	164
7.82.2.3	clear	164
7.82.2.4	print	164
7.82.2.5	printLine	164
7.82.2.6	setTitle	164
7.82.3	Member Data Documentation	164
7.82.3.1	ocolumnWidths	164
7.82.3.2	odata	164
7.82.3.3	otitle	164
7.83	BThread Class Reference	164
7.83.1	Constructor & Destructor Documentation	165
7.83.1.1	BThread	165
7.83.1.2	~BThread	165
7.83.2	Member Function Documentation	165
7.83.2.1	cancel	165
7.83.2.2	function	165
7.83.2.3	getThread	166
7.83.2.4	result	166
7.83.2.5	running	166
7.83.2.6	setInitPriority	166
7.83.2.7	setInitStackSize	166
7.83.2.8	setPriority	166
7.83.2.9	start	166
7.83.2.10	startFunc	166
7.83.2.11	waitForCompletion	166
7.83.3	Member Data Documentation	166
7.83.3.1	opolicy	166
7.83.3.2	opriority	166
7.83.3.3	oreult	166
7.83.3.4	orunning	166

7.83.3.5	ostackSize	166
7.83.3.6	othread	166
7.84	BTime Class Reference	166
7.84.1	Constructor & Destructor Documentation	167
7.84.1.1	BTime	167
7.84.2	Member Function Documentation	167
7.84.2.1	addSeconds	167
7.84.2.2	getDate	167
7.84.2.3	getSeconds	167
7.84.2.4	getString	167
7.84.2.5	getTime	168
7.84.2.6	isLeapYear	168
7.84.2.7	isSet	168
7.84.2.8	operator!=	168
7.84.2.9	operator+	168
7.84.2.10	operator+=	168
7.84.2.11	operator<	168
7.84.2.12	operator<=	168
7.84.2.13	operator==	168
7.84.2.14	operator>	168
7.84.2.15	operator>=	168
7.84.2.16	set	168
7.84.2.17	set	168
7.84.2.18	setString	168
7.84.2.19	setYearDay	168
7.84.3	Member Data Documentation	168
7.84.3.1	otime	168
7.85	BTimer Class Reference	169
7.85.1	Detailed Description	169
7.85.2	Constructor & Destructor Documentation	170
7.85.2.1	BTimer	170
7.85.2.2	~BTimer	170
7.85.3	Member Function Documentation	170
7.85.3.1	add	170
7.85.3.2	average	170
7.85.3.3	clear	170
7.85.3.4	getElapsedTime	170
7.85.3.5	getTime	170
7.85.3.6	peak	170
7.85.3.7	start	170

7.85.3.8	stop	170
7.85.4	Member Data Documentation	170
7.85.4.1	oaverage	170
7.85.4.2	oendTime	170
7.85.4.3	oclock	170
7.85.4.4	onum	170
7.85.4.5	opeak	170
7.85.4.6	ostartTime	171
7.86	BTimeStamp Class Reference	171
7.86.1	Constructor & Destructor Documentation	172
7.86.1.1	BTimeStamp	172
7.86.1.2	BTimeStamp	172
7.86.1.3	BTimeStamp	172
7.86.1.4	~BTimeStamp	173
7.86.2	Member Function Documentation	173
7.86.2.1	addMicroSeconds	173
7.86.2.2	addMilliSeconds	173
7.86.2.3	addSeconds	173
7.86.2.4	clear	173
7.86.2.5	compare	173
7.86.2.6	day	173
7.86.2.7	difference	173
7.86.2.8	getDate	173
7.86.2.9	getString	173
7.86.2.10	getStringFormatted	173
7.86.2.11	getStringNoMs	173
7.86.2.12	getYearMicroSeconds	173
7.86.2.13	getYearSeconds	173
7.86.2.14	hour	174
7.86.2.15	isLeap	174
7.86.2.16	isSet	174
7.86.2.17	microSecond	174
7.86.2.18	minute	174
7.86.2.19	month	174
7.86.2.20	operator BString	174
7.86.2.21	operator!=	174
7.86.2.22	operator<	174
7.86.2.23	operator<=	174
7.86.2.24	operator=	174
7.86.2.25	operator==	174

7.86.2.26 operator>	174
7.86.2.27 operator>=	174
7.86.2.28 second	174
7.86.2.29 set	174
7.86.2.30 set	174
7.86.2.31 set	174
7.86.2.32 setFirst	174
7.86.2.33 setLast	174
7.86.2.34 setNow	174
7.86.2.35 setString	175
7.86.2.36 setTime	175
7.86.2.37 setYDay	175
7.86.2.38 yday	175
7.86.2.39 year	175
7.86.3 Member Data Documentation	175
7.86.3.1 ohour	175
7.86.3.2 omicroSecond	175
7.86.3.3 ominute	175
7.86.3.4 osecond	175
7.86.3.5 ospare	175
7.86.3.6 oyday	175
7.86.3.7 oyear	175
7.87 BTimeStampMs Class Reference	175
7.87.1 Constructor & Destructor Documentation	177
7.87.1.1 BTimeStampMs	177
7.87.1.2 ~BTimeStampMs	177
7.87.2 Member Function Documentation	177
7.87.2.1 addMilliseconds	177
7.87.2.2 addSeconds	177
7.87.2.3 clear	177
7.87.2.4 compare	177
7.87.2.5 difference	177
7.87.2.6 getDate	177
7.87.2.7 getDurationString	177
7.87.2.8 getDurationStringNoMs	178
7.87.2.9 getString	178
7.87.2.10 getStringNoMs	178
7.87.2.11 getStringRaw	178
7.87.2.12 getYearMilliseconds	178
7.87.2.13 getYearSeconds	178

7.87.2.14	isLeap	178
7.87.2.15	operator<	178
7.87.2.16	operator<=	178
7.87.2.17	operator>	178
7.87.2.18	operator>=	178
7.87.2.19	setDurationString	178
7.87.2.20	setNow	178
7.87.2.21	setString	178
7.87.2.22	subMilliseconds	178
7.87.2.23	subSeconds	178
7.87.3	Member Data Documentation	179
7.87.3.1	hour	179
7.87.3.2	milliSecond	179
7.87.3.3	minute	179
7.87.3.4	sampleNumber	179
7.87.3.5	second	179
7.87.3.6	yday	179
7.87.3.7	year	179
7.88	BUrl Class Reference	179
7.88.1	Detailed Description	180
7.88.2	Constructor & Destructor Documentation	180
7.88.2.1	BUrl	180
7.88.2.2	~BUrl	180
7.88.3	Member Function Documentation	180
7.88.3.1	readString	180
7.88.3.2	writeData	180
7.88.4	Member Data Documentation	180
7.88.4.1	oinit	180
7.88.4.2	ores	180
7.89	Tms::ConfigInfo Class Reference	180
7.89.1	Detailed Description	181
7.89.2	Constructor & Destructor Documentation	181
7.89.2.1	ConfigInfo	181
7.89.3	Member Data Documentation	181
7.89.3.1	puReferences	181
7.90	Tms::CycleInformation Class Reference	181
7.90.1	Constructor & Destructor Documentation	181
7.90.1.1	CycleInformation	181
7.90.2	Member Data Documentation	181
7.90.2.1	cycleNumber	181

7.90.2.2	cycleType	181
7.90.2.3	periods	182
7.91	Tms::CycleInformationPeriod Class Reference	182
7.91.1	Detailed Description	182
7.91.2	Constructor & Destructor Documentation	182
7.91.2.1	CycleInformationPeriod	182
7.91.3	Member Data Documentation	182
7.91.3.1	bunchMask	182
7.91.3.2	cyclePeriod	183
7.91.3.3	endTime	183
7.91.3.4	harmonic	183
7.91.3.5	numBunches	183
7.91.3.6	numValues	183
7.91.3.7	startTime	183
7.92	Tms::CycleParam Class Reference	183
7.92.1	Detailed Description	184
7.92.2	Constructor & Destructor Documentation	184
7.92.2.1	CycleParam	184
7.92.3	Member Data Documentation	185
7.92.3.1	channel	185
7.92.3.2	cycleType	185
7.92.3.3	frefPhaseDelay	185
7.92.3.4	info	185
7.92.3.5	name	185
7.92.3.6	pllCycleStartFrequency	185
7.92.3.7	pllDdsMaximum	185
7.92.3.8	pllDdsMinimum	185
7.92.3.9	pllFrefGain	185
7.92.3.10	pllGain	185
7.92.3.11	pllInitialFrequency	185
7.92.3.12	pllInitialFrequencyDelay	186
7.92.3.13	ring	186
7.92.3.14	settings	186
7.92.3.15	stateDelay	186
7.92.3.16	stateTable	186
7.93	Tms::CycleParamDb Class Reference	186
7.93.1	Detailed Description	187
7.93.2	Constructor & Destructor Documentation	187
7.93.2.1	CycleParamDb	187
7.93.3	Member Function Documentation	187

7.93.3.1	deleteCycleParams	187
7.93.3.2	getCycleParams	187
7.93.3.3	getCycleTypes	187
7.93.3.4	getFileNames	187
7.93.3.5	readCycleParams	187
7.93.3.6	setCycleParams	187
7.93.3.7	writeCycleParams	187
7.93.4	Member Data Documentation	187
7.93.4.1	obaseDir	187
7.94	Tms::CycleParamEdit Class Reference	188
7.94.1	Detailed Description	188
7.94.2	Constructor & Destructor Documentation	189
7.94.2.1	CycleParamEdit	189
7.94.2.2	CycleParamEdit	189
7.94.3	Member Function Documentation	189
7.94.3.1	bunch	189
7.94.3.2	clear	189
7.94.3.3	generateState	189
7.94.3.4	getDefaultPickupPositions	189
7.94.3.5	getDefaultState	189
7.94.3.6	getStates	189
7.94.3.7	getString	189
7.94.3.8	readFromFile	189
7.94.3.9	setStates	189
7.94.3.10	setString	189
7.94.3.11	value	189
7.94.3.12	writeToFile	189
7.95	Tms::CycleParamItem Class Reference	190
7.95.1	Constructor & Destructor Documentation	190
7.95.1.1	CycleParamItem	190
7.95.2	Member Data Documentation	190
7.95.2.1	channel	190
7.95.2.2	cycleType	190
7.95.2.3	ring	190
7.96	Tms::CycleParamState Class Reference	190
7.96.1	Constructor & Destructor Documentation	191
7.96.1.1	CycleParamState	191
7.96.2	Member Function Documentation	191
7.96.2.1	clear	191
7.96.2.2	getString	192

7.96.2.3	setNext	192
7.96.2.4	setString	192
7.96.3	Member Data Documentation	192
7.96.3.1	blrPhase	192
7.96.3.2	blrWidth	192
7.96.3.3	bunchMask	192
7.96.3.4	gatePhase	192
7.96.3.5	gateWidth	192
7.96.3.6	lo1Harmonic	192
7.96.3.7	lo1Phase	192
7.96.3.8	lo2Harmonic	192
7.96.3.9	lo2Phase	192
7.96.3.10	mean1Mask	193
7.96.3.11	mean2Mask	193
7.96.3.12	num	193
7.96.3.13	period	193
7.96.3.14	state	193
7.97	Tms::CycleTypeInfo Class Reference	193
7.97.1	Constructor & Destructor Documentation	193
7.97.1.1	CycleTypeInfo	193
7.97.2	Member Data Documentation	193
7.97.2.1	cycleType	194
7.97.2.2	info	194
7.97.2.3	periods	194
7.98	Tms::CycleTypeInfoPeriod Class Reference	194
7.98.1	Detailed Description	194
7.98.2	Constructor & Destructor Documentation	194
7.98.2.1	CycleTypeInfoPeriod	194
7.98.3	Member Data Documentation	194
7.98.3.1	bunchMask	195
7.98.3.2	cyclePeriod	195
7.98.3.3	harmonic	195
7.98.3.4	numBunches	195
7.99	Tms::Data Class Reference	195
7.99.1	Detailed Description	195
7.99.2	Constructor & Destructor Documentation	196
7.99.2.1	Data	196
7.99.3	Member Data Documentation	196
7.99.3.1	dataType	196
7.99.3.2	dataValues	196

7.99.3.3 errors	196
7.99.3.4 numBunches	196
7.99.3.5 numChannels	196
7.99.3.6 numValues	196
7.100Tms::DataInfo Class Reference	196
7.100.1 Detailed Description	197
7.100.2 Constructor & Destructor Documentation	197
7.100.2.1 DataInfo	197
7.100.3 Member Data Documentation	197
7.100.3.1 argument	197
7.100.3.2 beyondPeriod	197
7.100.3.3 bunchNumber	197
7.100.3.4 channel	197
7.100.3.5 cycleNumber	198
7.100.3.6 cyclePeriod	198
7.100.3.7 function	198
7.100.3.8 numValues	198
7.100.3.9 orbitNumber	198
7.100.3.10startTime	198
7.101Tms::DataValue Class Reference	198
7.101.1 Detailed Description	199
7.101.2 Constructor & Destructor Documentation	199
7.101.2.1 DataValue	199
7.101.3 Member Data Documentation	199
7.101.3.1 deltaX	199
7.101.3.2 deltaY	199
7.101.3.3 sigma	199
7.101.3.4 time	199
7.102Tms::NameValue Class Reference	199
7.102.1 Constructor & Destructor Documentation	200
7.102.1.1 NameValue	200
7.102.2 Member Data Documentation	200
7.102.2.1 name	200
7.102.2.2 value	200
7.103BList< T >::Node Class Reference	200
7.103.1 Constructor & Destructor Documentation	200
7.103.1.1 Node	200
7.103.2 Member Data Documentation	200
7.103.2.1 item	200
7.104Tms::PuChannel Class Reference	201

7.104.1 Detailed Description	201
7.104.2 Constructor & Destructor Documentation	201
7.104.2.1 PuChannel	201
7.104.3 Member Data Documentation	201
7.104.3.1 moduleNum	201
7.104.3.2 pupeChan	201
7.104.3.3 pupeNum	201
7.105Tms::PuControl Class Reference	202
7.105.1 Detailed Description	203
7.105.2 Constructor & Destructor Documentation	203
7.105.2.1 PuControl	203
7.105.3 Member Function Documentation	203
7.105.3.1 captureDiagnostics	203
7.105.3.2 configure	203
7.105.3.3 getMasterPuChannel	203
7.105.3.4 getPupeConfig	203
7.105.3.5 getStatistics	203
7.105.3.6 getStatus	204
7.105.3.7 getVersion	204
7.105.3.8 init	204
7.105.3.9 setControlInfo	204
7.105.3.10setNextCycle	204
7.105.3.11setProcessPriority	204
7.105.3.12setPupeConfig	204
7.105.3.13setTestData	204
7.105.3.14setTestMode	204
7.105.3.15setTimingSignals	204
7.105.3.16test	205
7.106Tms::PupeConfig Class Reference	205
7.106.1 Constructor & Destructor Documentation	205
7.106.1.1 PupeConfig	205
7.106.2 Member Data Documentation	205
7.106.2.1 adcSysclkSync	205
7.106.2.2 disableBlr	205
7.106.2.3 doubleInjection	205
7.106.2.4 internalTimingMask	206
7.107Tms::PuProcess Class Reference	206
7.107.1 Detailed Description	206
7.107.2 Constructor & Destructor Documentation	207
7.107.2.1 PuProcess	207

7.107.3 Member Function Documentation	207
7.107.3.1 addEventServer	207
7.107.3.2 getCycleInformation	207
7.107.3.3 getData	207
7.107.3.4 getStatus	207
7.107.3.5 getVersion	207
7.107.3.6 requestData	207
7.108Tms::PuStateTable Class Reference	207
7.108.1 Detailed Description	208
7.108.2 Constructor & Destructor Documentation	208
7.108.2.1 PuStateTable	208
7.108.3 Member Data Documentation	208
7.108.3.1 bunchMask	208
7.108.3.2 harmonic	208
7.108.3.3 numBunches	208
7.108.3.4 period	208
7.108.3.5 phaseTable	208
7.108.3.6 state	209
7.109Tms::PuStatus Class Reference	209
7.109.1 Detailed Description	209
7.109.2 Constructor & Destructor Documentation	209
7.109.2.1 PuStatus	209
7.109.3 Member Data Documentation	209
7.109.3.1 error	209
7.109.3.2 running	209
7.110SigGen Class Reference	210
7.110.1 Constructor & Destructor Documentation	210
7.110.1.1 SigGen	210
7.110.1.2 ~SigGen	210
7.110.2 Member Function Documentation	210
7.110.2.1 config	210
7.110.2.2 generate	210
7.110.3 Member Data Documentation	210
7.110.3.1 osampleRate	210
7.110.3.2 ox	210
7.111SigGenBeam Class Reference	211
7.111.1 Constructor & Destructor Documentation	211
7.111.1.1 SigGenBeam	211
7.111.1.2 ~SigGenBeam	211
7.111.2 Member Function Documentation	211

7.111.2.1 config	211
7.111.2.2 generate	211
7.111.2.3 generateIntegrated	211
7.111.3 Member Data Documentation	211
7.111.3.1 oamplitude	211
7.111.3.2 obl	212
7.111.3.3 obunchSet	212
7.111.3.4 ofref	212
7.111.3.5 oharmonic	212
7.111.3.6 oreduce	212
7.112SigGenNoise Class Reference	212
7.112.1 Constructor & Destructor Documentation	212
7.112.1.1 SigGenNoise	212
7.112.1.2 ~SigGenNoise	212
7.112.2 Member Function Documentation	212
7.112.2.1 config	212
7.112.2.2 generate	212
7.112.3 Member Data Documentation	213
7.112.3.1 oamplitude	213
7.113SigGenPulse Class Reference	213
7.113.1 Constructor & Destructor Documentation	213
7.113.1.1 SigGenPulse	213
7.113.1.2 ~SigGenPulse	213
7.113.2 Member Function Documentation	213
7.113.2.1 config	213
7.113.2.2 generate	213
7.113.3 Member Data Documentation	214
7.113.3.1 oamplitude	214
7.113.3.2 ofreq	214
7.113.3.3 oonTime	214
7.113.3.4 ostartTime	214
7.114SigGenSine Class Reference	214
7.114.1 Constructor & Destructor Documentation	214
7.114.1.1 SigGenSine	214
7.114.1.2 ~SigGenSine	214
7.114.2 Member Function Documentation	214
7.114.2.1 config	214
7.114.2.2 generate	215
7.114.3 Member Data Documentation	215
7.114.3.1 oamplitude	215

7.114.3.2 ofreq	215
7.115SigGenSquare Class Reference	215
7.115.1 Constructor & Destructor Documentation	215
7.115.1.1 SigGenSquare	215
7.115.1.2 ~SigGenSquare	215
7.115.2 Member Function Documentation	215
7.115.2.1 config	215
7.115.2.2 generate	216
7.115.3 Member Data Documentation	216
7.115.3.1 oamplitude	216
7.115.3.2 ofreq	216
7.115.3.3 ooffset	216
7.116Tms::Simulation Class Reference	216
7.116.1 Constructor & Destructor Documentation	216
7.116.1.1 Simulation	216
7.116.2 Member Data Documentation	216
7.116.2.1 cycleType	216
7.116.2.2 data	217
7.116.2.3 doubleInjection	217
7.116.2.4 setNextCycle	217
7.116.2.5 timing	217
7.117Tms::TestCaptureInfo Class Reference	217
7.117.1 Detailed Description	218
7.117.2 Constructor & Destructor Documentation	218
7.117.2.1 TestCaptureInfo	218
7.117.3 Member Data Documentation	218
7.117.3.1 clock	218
7.117.3.2 postTriggerDelay	218
7.117.3.3 source	218
7.117.3.4 startTime	218
7.117.3.5 triggerAnd	218
7.117.3.6 triggerMask	218
7.117.3.7 triggerSourceData	218
7.117.3.8 triggerState	218
7.117.3.9 triggerStateEnable	219
7.117.3.10triggerStore	219
7.118Tms::TmsControl Class Reference	219
7.118.1 Detailed Description	220
7.118.2 Constructor & Destructor Documentation	220
7.118.2.1 TmsControl	220

7.118.3 Member Function Documentation	220
7.118.3.1 captureDiagnostics	221
7.118.3.2 configure	221
7.118.3.3 delControlInfo	221
7.118.3.4 getConfiguration	221
7.118.3.5 getControlInfo	221
7.118.3.6 getControlList	221
7.118.3.7 getPuChannel	222
7.118.3.8 getPupeConfig	222
7.118.3.9 getSimulation	222
7.118.3.10 getStatistics	222
7.118.3.11 getStatus	222
7.118.3.12 getVersion	222
7.118.3.13 nit	223
7.118.3.14 puServerStarted	223
7.118.3.15 setControlInfo	223
7.118.3.16 setNextCycle	223
7.118.3.17 setProcessPriority	223
7.118.3.18 setPupeConfig	224
7.118.3.19 setSimulation	224
7.118.3.20 setTestData	224
7.118.3.21 setTestMode	224
7.118.3.22 setTimingSignals	224
7.118.3.23 test	225
7.119 Tms::TmsEvent Class Reference	225
7.119.1 Detailed Description	226
7.119.2 Constructor & Destructor Documentation	226
7.119.2.1 TmsEvent	226
7.119.3 Member Function Documentation	226
7.119.3.1 cycleStartEvent	226
7.119.3.2 cycleStopEvent	226
7.119.3.3 dataEvent	226
7.119.3.4 errorEvent	226
7.120 Tms::TmsEventServerList Class Reference	226
7.120.1 Constructor & Destructor Documentation	227
7.120.1.1 TmsEventServerList	227
7.120.1.2 ~TmsEventServerList	227
7.120.2 Member Function Documentation	227
7.120.2.1 append	227
7.120.2.2 cycleStartEvent	227

7.120.2.3 cycleStopEvent	227
7.120.2.4 dataEvent	227
7.120.2.5 del	227
7.120.2.6 errorEvent	227
7.120.3 Member Data Documentation	227
7.120.3.1 oeventServers	227
7.120.3.2 olock	227
7.121 Tms::TmsPhase Union Reference	228
7.121.1 Detailed Description	228
7.121.2 Member Data Documentation	228
7.121.2.1 "@4	228
7.121.2.2 blr	228
7.121.2.3 gate	228
7.121.2.4 lo1	228
7.121.2.5 lo2	228
7.121.2.6 meanFilter1	228
7.121.2.7 meanFilter2	228
7.121.2.8 spare	228
7.121.2.9 value	228
7.122 Tms::TmsProcess Class Reference	228
7.122.1 Detailed Description	229
7.122.2 Constructor & Destructor Documentation	229
7.122.2.1 TmsProcess	229
7.122.3 Member Function Documentation	230
7.122.3.1 addEventServer	230
7.122.3.2 getCycleInfo	231
7.122.3.3 getCycleInformation	231
7.122.3.4 getCycleTypeInfoInformation	231
7.122.3.5 getData	231
7.122.3.6 getVersion	232
7.122.3.7 requestData	232
7.123 Tms::TmsState Union Reference	232
7.123.1 Detailed Description	233
7.123.2 Member Data Documentation	233
7.123.2.1 "@2	233
7.123.2.2 acquireData	233
7.123.2.3 bit6	233
7.123.2.4 bit7	233
7.123.2.5 calStart	233
7.123.2.6 calStop	233

7.123.2.7	cycleStop	233
7.123.2.8	delay	233
7.123.2.9	hchange	233
7.123.2.10	injection	233
7.123.2.11	pllFeedbackSelect	233
7.123.2.12	pllLO1FromAddress	233
7.123.2.13	pllLO2FromAddress	233
7.123.2.14	pllReference1	233
7.123.2.15	pllReference2	233
7.123.2.16	value	233
8	File Documentation	235
8.1	/src/cern/tms/beam/libBeam/BArray.h File Reference	235
8.1.1	Macro Definition Documentation	235
8.1.1.1	BArrayLoop	235
8.2	/src/cern/tms/beam/libBeam/BAtomic.h File Reference	235
8.2.1	Typedef Documentation	236
8.2.1.1	BAtomicInt32	236
8.2.1.2	BAtomicInt64	236
8.2.1.3	BAtomicUInt32	236
8.2.1.4	BAtomicUInt64	236
8.3	/src/cern/tms/beam/libBeam/BAtomicCount.h File Reference	236
8.4	/src/cern/tms/beam/libBeam/BBuffer.cpp File Reference	236
8.4.1	Variable Documentation	236
8.4.1.1	roundSize	236
8.5	/src/cern/tms/beam/libBeam/BBuffer.h File Reference	236
8.5.1	Macro Definition Documentation	237
8.5.1.1	BBigEndian	237
8.6	/src/cern/tms/beam/libBeam/BComms.cpp File Reference	237
8.7	/src/cern/tms/beam/libBeam/BComms.h File Reference	237
8.8	/src/cern/tms/beam/libBeam/BComplex.h File Reference	237
8.8.1	Typedef Documentation	237
8.8.1.1	BComplex	237
8.8.1.2	BComplex32	237
8.8.1.3	BComplex64	237
8.9	/src/cern/tms/beam/libBeam/BCond.cpp File Reference	238
8.10	/src/cern/tms/beam/libBeam/BCond.h File Reference	238
8.11	/src/cern/tms/beam/libBeam/BCondInt.cpp File Reference	238
8.11.1	Function Documentation	238
8.11.1.1	getTimeout	238

8.12	/src/cern/tms/beam/libBeam/BCondInt.h File Reference	238
8.13	/src/cern/tms/beam/libBeam/BConfig.cpp File Reference	239
8.14	/src/cern/tms/beam/libBeam/BConfig.h File Reference	239
8.15	/src/cern/tms/beam/libBeam/BCrc16.cpp File Reference	239
8.15.1	Function Documentation	239
8.15.1.1	bcrc16	239
8.15.2	Variable Documentation	239
8.15.2.1	table_crc_hi	239
8.15.2.2	table_crc_lo	240
8.16	/src/cern/tms/beam/libBeam/BCrc16.h File Reference	240
8.16.1	Function Documentation	240
8.16.1.1	bcrc16	240
8.17	/src/cern/tms/beam/libBeam/BDate.cpp File Reference	240
8.17.1	Function Documentation	241
8.17.1.1	fromBString	241
8.17.1.2	toBString	241
8.17.2	Variable Documentation	241
8.17.2.1	mon_yday	241
8.18	/src/cern/tms/beam/libBeam/BDate.h File Reference	241
8.18.1	Function Documentation	241
8.18.1.1	fromBString	241
8.18.1.2	toBString	241
8.19	/src/cern/tms/beam/libBeam/BDebug.cpp File Reference	242
8.19.1	Macro Definition Documentation	242
8.19.1.1	BTRACE_SIZE	242
8.19.2	Function Documentation	242
8.19.2.1	gettid	242
8.19.2.2	getTime	242
8.19.2.3	hd32	242
8.19.2.4	hd8	242
8.19.2.5	hd8a	242
8.19.2.6	hda32	243
8.19.2.7	hda8	243
8.19.2.8	setDebug	243
8.19.2.9	tprintf	243
8.19.3	Variable Documentation	243
8.19.3.1	bdebug	243
8.19.3.2	STRBUF_SIZE	243
8.20	/src/cern/tms/beam/libBeam/BDebug.h File Reference	243
8.20.1	Macro Definition Documentation	244

8.20.1.1	BDebug_STD	244
8.20.1.2	dprintf	244
8.20.1.3	eprintf	244
8.20.1.4	nprintf	244
8.20.1.5	wprintf	244
8.20.2	Function Documentation	244
8.20.2.1	gettid	244
8.20.2.2	getTime	244
8.20.2.3	hd32	244
8.20.2.4	hd8	244
8.20.2.5	hd8a	244
8.20.2.6	hda8	244
8.20.2.7	hds32	244
8.20.2.8	setDebug	244
8.20.2.9	tprintf	244
8.20.3	Variable Documentation	244
8.20.3.1	bdebug	244
8.21	/src/cern/tms/beam/libBeam/BDict.cpp File Reference	244
8.21.1	Function Documentation	245
8.21.1.1	bdictStringToString	245
8.21.1.2	fromBString	245
8.21.1.3	toBString	245
8.22	/src/cern/tms/beam/libBeam/BDict.h File Reference	245
8.22.1	Typedef Documentation	245
8.22.1.1	BDictString	245
8.22.2	Function Documentation	245
8.22.2.1	bdictStringToString	245
8.22.2.2	fromBString	245
8.22.2.3	toBString	245
8.23	/src/cern/tms/beam/libBeam/BDictMap.h File Reference	245
8.23.1	Typedef Documentation	246
8.23.1.1	BDictMapString	246
8.24	/src/cern/tms/beam/libBeam/BDir.cpp File Reference	246
8.24.1	Function Documentation	246
8.24.1.1	wild	246
8.24.2	Variable Documentation	246
8.24.2.1	wildString	246
8.25	/src/cern/tms/beam/libBeam/BDir.h File Reference	246
8.26	/src/cern/tms/beam/libBeam/BDuration.cpp File Reference	246
8.27	/src/cern/tms/beam/libBeam/BDuration.h File Reference	247

8.28 /src/cern/tms/beam/libBeam/BEndian.cpp File Reference	247
8.28.1 Function Documentation	247
8.28.1.1 bswap_copy	247
8.29 /src/cern/tms/beam/libBeam/BEndian.h File Reference	247
8.29.1 Macro Definition Documentation	248
8.29.1.1 be16toh	248
8.29.1.2 be32toh	248
8.29.1.3 be64toh	248
8.29.1.4 htobe16	248
8.29.1.5 htobe32	248
8.29.1.6 htobe64	248
8.29.1.7 htole16	248
8.29.1.8 htole32	248
8.29.1.9 htole64	248
8.29.1.10 le16toh	249
8.29.1.11 le32toh	249
8.29.1.12 le64toh	249
8.29.2 Function Documentation	249
8.29.2.1 betoh	249
8.29.2.2 betoh	249
8.29.2.3 betoh	249
8.29.2.4 betoh	249
8.29.2.5 betoh	249
8.29.2.6 betoh	249
8.29.2.7 betoh	249
8.29.2.8 betoh	249
8.29.2.9 bswap_copy	249
8.29.2.10 bswap_p16	249
8.29.2.11 bswap_p32	249
8.29.2.12 bswap_p64	249
8.29.2.13 bswap_p8	249
8.29.2.14 htobe	249
8.29.2.15 htobe	249
8.29.2.16 htobe	249
8.29.2.17 htobe	249
8.29.2.18 htobe	249
8.29.2.19 htobe	249
8.29.2.20 htobe	249
8.29.2.21 htobe	249
8.29.2.22 htobe	249

8.29.2.23 htole	249
8.29.2.24 htole	249
8.29.2.25 htole	250
8.29.2.26 htole	250
8.29.2.27 htole	250
8.29.2.28 htole	250
8.29.2.29 htole	250
8.29.2.30 letoh	250
8.29.2.31 letoh	250
8.29.2.32 letoh	250
8.29.2.33 letoh	250
8.29.2.34 letoh	250
8.29.2.35 letoh	250
8.29.2.36 letoh	250
8.29.2.37 letoh	250
8.30 /src/cern/tms/beam/libBeam/BEntry.cpp File Reference	250
8.31 /src/cern/tms/beam/libBeam/BEntry.h File Reference	250
8.32 /src/cern/tms/beam/libBeam/BError.cpp File Reference	251
8.33 /src/cern/tms/beam/libBeam/BError.h File Reference	251
8.33.1 Enumeration Type Documentation	251
8.33.1.1 BErrorNum	251
8.34 /src/cern/tms/beam/libBeam/BErrorTime.cpp File Reference	252
8.35 /src/cern/tms/beam/libBeam/BErrorTime.h File Reference	252
8.36 /src/cern/tms/beam/libBeam/BEvent.cpp File Reference	252
8.37 /src/cern/tms/beam/libBeam/BEvent.h File Reference	252
8.37.1 Typedef Documentation	253
8.37.1.1 BEventQueue	253
8.37.2 Enumeration Type Documentation	253
8.37.2.1 BEventType	253
8.38 /src/cern/tms/beam/libBeam/BEvent1.cpp File Reference	253
8.39 /src/cern/tms/beam/libBeam/BEvent1.h File Reference	253
8.39.1 Enumeration Type Documentation	254
8.39.1.1 BEvent1Type	254
8.40 /src/cern/tms/beam/libBeam/BFifo.h File Reference	254
8.41 /src/cern/tms/beam/libBeam/BFifo.inc File Reference	254
8.42 /src/cern/tms/beam/libBeam/BFifoCirc.cpp File Reference	254
8.42.1 Macro Definition Documentation	254
8.42.1.1 dprintf	254
8.43 /src/cern/tms/beam/libBeam/BFifoCirc.h File Reference	254
8.44 /src/cern/tms/beam/libBeam/BFifoCirc.inc File Reference	255

8.45	/src/cern/tms/beam/libBeam/BFile.cpp File Reference	255
8.45.1	Macro Definition Documentation	255
8.45.1.1	STRBUF	255
8.46	/src/cern/tms/beam/libBeam/BFile.h File Reference	255
8.47	/src/cern/tms/beam/libBeam/BFileCsv.cpp File Reference	255
8.48	/src/cern/tms/beam/libBeam/BFileCsv.h File Reference	256
8.49	/src/cern/tms/beam/libBeam/BFileData.cpp File Reference	256
8.50	/src/cern/tms/beam/libBeam/BFileData.h File Reference	256
8.51	/src/cern/tms/beam/libBeam/BList.h File Reference	256
8.51.1	Macro Definition Documentation	257
8.51.1.1	BListLoop	257
8.52	/src/cern/tms/beam/libBeam/BList_func.h File Reference	257
8.53	/src/cern/tms/beam/libBeam/BMutex.cpp File Reference	257
8.53.1	Macro Definition Documentation	257
8.53.1.1	MDEBUG	257
8.54	/src/cern/tms/beam/libBeam/BMutex.h File Reference	257
8.55	/src/cern/tms/beam/libBeam/BMysql.cpp File Reference	257
8.56	/src/cern/tms/beam/libBeam/BMysql.h File Reference	257
8.57	/src/cern/tms/beam/libBeam/BNameValue.h File Reference	258
8.58	/src/cern/tms/beam/libBeam/Boap.cpp File Reference	258
8.58.1	Macro Definition Documentation	258
8.58.1.1	APIVERSION_TEST	258
8.58.1.2	DEBUG	258
8.58.1.3	dprintf	259
8.58.1.4	IS_BIG_ENDIAN	259
8.58.2	Variable Documentation	259
8.58.2.1	boapPort	259
8.59	/src/cern/tms/beam/libBeam/Boap.h File Reference	259
8.59.1	Typedef Documentation	260
8.59.1.1	BoapFunc	260
8.59.1.2	BoapService	260
8.59.2	Enumeration Type Documentation	260
8.59.2.1	BoapPriority	260
8.59.2.2	BoapType	260
8.59.3	Variable Documentation	260
8.59.3.1	BoapMagic	260
8.60	/src/cern/tms/beam/libBeam/BoapMc.cpp File Reference	260
8.60.1	Macro Definition Documentation	261
8.60.1.1	DEBUG_LOCAL	261
8.60.1.2	DEBUG_LOCAL1	261

8.60.1.3	dl1printf	261
8.60.1.4	dlprintf	261
8.61	/src/cern/tms/beam/libBeam/BoapMc.h File Reference	261
8.61.1	Enumeration Type Documentation	261
8.61.1.1	BoapMcType	261
8.61.2	Function Documentation	262
8.61.2.1	__attribute__	262
8.61.3	Variable Documentation	262
8.61.3.1	__attribute__	262
8.61.3.2	addressFrom	262
8.61.3.3	addressTo	262
8.61.3.4	checksum	262
8.61.3.5	cmd	262
8.61.3.6	error	262
8.61.3.7	length	262
8.62	/src/cern/tms/beam/libBeam/BoapnsC.cpp File Reference	262
8.63	/src/cern/tms/beam/libBeam/BoapnsC.h File Reference	262
8.64	/src/cern/tms/beam/libBeam/BoapnsD.cpp File Reference	263
8.65	/src/cern/tms/beam/libBeam/BoapnsD.h File Reference	263
8.66	/src/cern/tms/beam/libBeam/BoapSimple.cc File Reference	263
8.66.1	Macro Definition Documentation	264
8.66.1.1	DEBUG	264
8.66.1.2	dprintf	264
8.66.2	Variable Documentation	264
8.66.2.1	roundSize	264
8.67	/src/cern/tms/beam/libBeam/BoapSimple.h File Reference	264
8.67.1	Typedef Documentation	265
8.67.1.1	BoapFunc	265
8.67.1.2	BoapService	265
8.67.1.3	Double	265
8.67.1.4	Int16	265
8.67.1.5	Int32	265
8.67.1.6	Int8	265
8.67.1.7	UInt16	265
8.67.1.8	UInt32	265
8.67.1.9	UInt8	265
8.67.2	Enumeration Type Documentation	265
8.67.2.1	BoapType	265
8.68	/src/cern/tms/beam/libBeam/BObj.cpp File Reference	265
8.69	/src/cern/tms/beam/libBeam/BObj.h File Reference	265

8.70 /src/cern/tms/beam/libBeam/BObjStringFormat.cpp File Reference	266
8.70.1 Function Documentation	267
8.70.1.1 toBDictStringFromJson	267
8.70.1.2 toBString	267
8.70.1.3 toBString	267
8.70.1.4 toBString	267
8.70.1.5 toBString	267
8.70.1.6 toBString	267
8.70.1.7 toBString	267
8.70.1.8 toBString	267
8.70.1.9 toBString	267
8.70.1.10 toBString	267
8.70.1.11 toBString	267
8.70.1.12 toBString	267
8.70.1.13 toBString	267
8.70.1.14 toBString	267
8.70.1.15 toBString	267
8.70.1.16 toBString	267
8.70.1.17 toBString	267
8.70.1.18 toBString	267
8.70.1.19 toBString	267
8.70.1.20 toBStringJson	267
8.70.1.21 toBStringJson	267
8.70.1.22 toBStringJson	267
8.70.1.23 toBStringJson	267
8.70.1.24 toBStringJson	267
8.70.1.25 toBStringJson	267
8.70.1.26 toBStringJson	267
8.70.1.27 toBStringJson	267
8.70.1.28 toBStringJson	268
8.70.1.29 toBStringJson	268
8.70.1.30 toBStringJson	268
8.70.1.31 toBStringJson	268
8.70.1.32 toBStringJson	268
8.70.1.33 toBStringJson	268
8.70.1.34 toBStringJson	268
8.70.1.35 toBStringJson	268
8.70.1.36 toBStringJson	268
8.70.1.37 toBStringJson	268
8.71 /src/cern/tms/beam/libBeam/BObjStringFormat.h File Reference	268

8.71.1	Function Documentation	269
8.71.1.1	base64_decode	269
8.71.1.2	base64_encode	269
8.71.1.3	toBDictStringFromJson	269
8.71.1.4	toBString	269
8.71.1.5	toBString	269
8.71.1.6	toBString	269
8.71.1.7	toBString	269
8.71.1.8	toBString	269
8.71.1.9	toBString	269
8.71.1.10	toBString	269
8.71.1.11	toBString	269
8.71.1.12	toBString	269
8.71.1.13	toBString	269
8.71.1.14	toBString	269
8.71.1.15	toBString	269
8.71.1.16	toBString	269
8.71.1.17	toBString	269
8.71.1.18	toBString	269
8.71.1.19	toBString	269
8.71.1.20	toBStringJson	269
8.71.1.21	toBStringJson	270
8.71.1.22	toBStringJson	270
8.71.1.23	toBStringJson	270
8.71.1.24	toBStringJson	270
8.71.1.25	toBStringJson	270
8.71.1.26	toBStringJson	270
8.71.1.27	toBStringJson	270
8.71.1.28	toBStringJson	270
8.71.1.29	toBStringJson	270
8.71.1.30	toBStringJson	270
8.71.1.31	toBStringJson	270
8.71.1.32	toBStringJson	270
8.71.1.33	toBStringJson	270
8.71.1.34	toBStringJson	270
8.71.1.35	toBStringJson	270
8.72	/src/cern/tms/beam/libBeam/BPoll.cpp File Reference	270
8.73	/src/cern/tms/beam/libBeam/BPoll.h File Reference	270
8.74	/src/cern/tms/beam/libBeam/BQueue.h File Reference	271
8.74.1	Typedef Documentation	271

8.74.1.1	BQueueInt	271
8.75	/src/cern/tms/beam/libBeam/BRefData.cpp File Reference	271
8.75.1	Macro Definition Documentation	271
8.75.1.1	CHUNK	271
8.76	/src/cern/tms/beam/libBeam/BRefData.h File Reference	271
8.77	/src/cern/tms/beam/libBeam/BRtc.cpp File Reference	272
8.78	/src/cern/tms/beam/libBeam/BRtc.h File Reference	272
8.79	/src/cern/tms/beam/libBeam/BRWLock.cpp File Reference	272
8.80	/src/cern/tms/beam/libBeam/BRWLock.h File Reference	272
8.81	/src/cern/tms/beam/libBeam/BSema.cpp File Reference	272
8.82	/src/cern/tms/beam/libBeam/BSema.h File Reference	273
8.83	/src/cern/tms/beam/libBeam/BSemaphore.cpp File Reference	273
8.84	/src/cern/tms/beam/libBeam/BSemaphore.h File Reference	273
8.85	/src/cern/tms/beam/libBeam/BSocket.cpp File Reference	273
8.85.1	Macro Definition Documentation	274
8.85.1.1	IP_MTU	274
8.86	/src/cern/tms/beam/libBeam/BSocket.h File Reference	274
8.87	/src/cern/tms/beam/libBeam/BSpi.cpp File Reference	274
8.88	/src/cern/tms/beam/libBeam/BSpi.h File Reference	274
8.89	/src/cern/tms/beam/libBeam/BString.cpp File Reference	275
8.89.1	Macro Definition Documentation	275
8.89.1.1	MINUS	275
8.89.1.2	STRIP	275
8.89.2	Function Documentation	276
8.89.2.1	barrayToString	276
8.89.2.2	blistToString	276
8.89.2.3	bstringListinList	276
8.89.2.4	bstringToArray	276
8.89.2.5	bstringToList	276
8.89.2.6	charToArray	276
8.89.2.7	charToList	276
8.89.2.8	fromBString	276
8.89.2.9	fromBString	276
8.89.2.10	fromBString	276
8.89.2.11	fromBString	276
8.89.2.12	fromBString	276
8.89.2.13	fromBString	276
8.89.2.14	gmatch	276
8.89.2.15	operator<<	276
8.89.2.16	operator>>	276

8.89.2.17	toBString	276
8.89.2.18	toBString	276
8.89.2.19	toBString	276
8.89.2.20	toBString	276
8.89.2.21	toBString	276
8.89.2.22	toBString	276
8.89.3	Variable Documentation	276
8.89.3.1	base64_decode_table	276
8.90	/src/cern/tms/beam/libBeam/BString.h File Reference	277
8.90.1	Function Documentation	277
8.90.1.1	fromBString	277
8.90.1.2	fromBString	277
8.90.1.3	fromBString	277
8.90.1.4	fromBString	277
8.90.1.5	fromBString	277
8.90.1.6	fromBString	277
8.90.1.7	operator<<	277
8.90.1.8	operator>>	277
8.90.1.9	toBString	278
8.90.1.10	toBString	278
8.90.1.11	toBString	278
8.90.1.12	toBString	278
8.90.1.13	toBString	278
8.90.1.14	toBString	278
8.91	/src/cern/tms/beam/libBeam/BStringLocked.h File Reference	278
8.92	/src/cern/tms/beam/libBeam/BTable.cpp File Reference	278
8.93	/src/cern/tms/beam/libBeam/BTable.h File Reference	278
8.94	/src/cern/tms/beam/libBeam/BThread.cpp File Reference	278
8.95	/src/cern/tms/beam/libBeam/BThread.h File Reference	279
8.96	/src/cern/tms/beam/libBeam/BTime.cpp File Reference	279
8.96.1	Function Documentation	279
8.96.1.1	yearDays	279
8.96.1.2	yearIsLeap	279
8.96.2	Variable Documentation	279
8.96.2.1	monDays	279
8.97	/src/cern/tms/beam/libBeam/BTime.h File Reference	279
8.98	/src/cern/tms/beam/libBeam/BTimer.cpp File Reference	280
8.99	/src/cern/tms/beam/libBeam/BTimer.h File Reference	280
8.100	/src/cern/tms/beam/libBeam/BTimeStamp.cpp File Reference	280
8.100.1	Function Documentation	280

8.100.1.1 fromBString	280
8.100.1.2 toBString	280
8.100.2 Variable Documentation	280
8.100.2.1 mon_yday	280
8.101/src/cern/tms/beam/libBeam/BTimeStamp.h File Reference	281
8.101.1 Function Documentation	281
8.101.1.1 fromBString	281
8.101.1.2 toBString	281
8.102/src/cern/tms/beam/libBeam/BTimeStampMs.cpp File Reference	281
8.102.1 Variable Documentation	281
8.102.1.1 mon_yday	281
8.103/src/cern/tms/beam/libBeam/BTimeStampMs.h File Reference	281
8.104/src/cern/tms/beam/libBeam/BTypes.h File Reference	282
8.104.1 Typedef Documentation	283
8.104.1.1 BArrayDouble	283
8.104.1.2 BArrayFloat	283
8.104.1.3 BChar	283
8.104.1.4 BDouble	283
8.104.1.5 BFloat	283
8.104.1.6 BFloat32	283
8.104.1.7 BFloat64	283
8.104.1.8 BInt	283
8.104.1.9 BInt16	283
8.104.1.10 BInt32	283
8.104.1.11 BInt64	283
8.104.1.12 BInt8	283
8.104.1.13 BBool	283
8.104.1.14 BSize	283
8.104.1.15 BTimeout	283
8.104.1.16 BUInt	283
8.104.1.17 BUInt16	283
8.104.1.18 BUInt32	283
8.104.1.19 BUInt64	283
8.104.1.20 BUInt8	283
8.104.2 Enumeration Type Documentation	284
8.104.2.1 BType	284
8.104.2.2 BTypeComp	284
8.104.3 Function Documentation	284
8.104.3.1 byteSwap16	284
8.104.3.2 byteSwap32	284

8.104.3.3 byteSwap64	284
8.104.3.4 byteSwap8	284
8.104.3.5 timeoutTicks	284
8.104.4 Variable Documentation	284
8.104.4.1 BTimeoutForever	284
8.105/src/cern/tms/beam/libBeam/BUrl.cpp File Reference	285
8.106/src/cern/tms/beam/libBeam/BUrl.h File Reference	285
8.107overview.dox File Reference	285
8.108SigGen.cpp File Reference	285
8.108.1 Macro Definition Documentation	285
8.108.1.1 DEBUG	285
8.108.1.2 dprintf	285
8.109SigGen.h File Reference	285
8.109.1 Typedef Documentation	286
8.109.1.1 BSignalList	286
8.109.1.2 Sample	286
8.110test1.cpp File Reference	286
8.110.1 Function Documentation	286
8.110.1.1 main	286
8.110.1.2 printCycleParams	286
8.111TmsC.cc File Reference	286
8.112TmsC.h File Reference	287
8.112.1 Detailed Description	287
8.113TmsCycleParam.cc File Reference	287
8.114TmsCycleParam.h File Reference	288
8.115TmsD.cc File Reference	288
8.116TmsD.h File Reference	288
8.117TmsEventServerList.cc File Reference	289
8.118TmsEventServerList.h File Reference	290
8.119tmsFunctions.dox File Reference	290
8.120TmsLib.cc File Reference	290
8.121TmsLib.h File Reference	290
8.122TmsS.cc File Reference	291
8.123TmsT.cc File Reference	291

Chapter 1

Main Page

Author

Dr Terry Barnaby

Version

2.0.0

Date

2013-02-01

1.1 Introduction

This document covers the BEAM LibTms software API for the CERN trajectory measurement system. This API provides the ability to control and receive data from the TMS System. The API is an object orientated API implemented in 'C++' with a number of object classes. The API operates over a network type interface using an RPC type mechanism.

The LibTms API makes use of the BEAM standard class library. The BEAM standard class library provides a small set of low level 'C++' classes for strings, lists and system interface functions. There is some brief information on the BEAM class library later on in this page.

1.2 Overview

Generally users of the system are only concerned with the top level, System Controller API. This is the API that control and data gathering clients use to control and gather data from the system. The System Controller API (Tms-Api) is implemented using a simple, object orientated, RPC mechanism. Two main objects, the [Tms::TmsControl](#) and [Tms::TmsProcess](#) objects, provide the full API.

The TmsApi has been developed using the BOAP (BEAM Object Access Protocol). This provides a simple but powerful Object Orientated RPC mechanism. The TmsApi is written in a high level interface definition language (IDL). The bidl tool generates the client and server side 'C++' interface and implementation files for the API. These are then provided as a set of 'C++' header files and a binary library file for the clients to link to. The BOAP system employs a simple BOAP name server process that provides a translation between object names and IPAddress/-Socket numbers. The BOAP name server runs on the System Controller. More information on the BOAP system can be found in the libBeam documentation.

There are two main Objects that are used by clients of the TMS API. They are the [Tms::TmsControl](#) and the [Tms::TmsProcess](#) objects. The [Tms::TmsControl](#) object is used for system configuration, testing and diagnostics. The

Tms::TmsProcess object is used for normal clients for Proton Synchrotron (PS) Cycle information configuration and data access. There is some example client code in the `tmsExamples` of the source code and displayed later on this page. These objects communicate through a network connection with the `TmsServer` process running on the TMS System Controller. The TMS System Controller operates as a multi-threaded process and can communicate with multiple clients simultaneously.

The TMS system takes most of its system timing signals from digital timing lines connected to the TMS rack hardware. The only timing information that external software needs to supply is the next cycle number and cycle type information. The cycle number is a 32bit unsigned number identifying the next Proton Synchrotron (PS) machine cycle. The cycle type is an ASCII string defining the type of BEAM present in the PS machine. The cycle type defines a set of state/phase tables to be loaded in order to measure the BEAM in the machine. The CERN client software needs to provide this information by calling the `setNextCycle()` function before the next PS cycle is initiated.

The TMS system keeps a library of state/phase tables indexed by the cycle type. These are loaded into the individual PUPE engines FPGA's during the `CYCLE_STOP` to `CYCLE_START` period. The API provides the **setControllInfo** and **delControllInfo** calls to maintain this database.

A client would generally use the **Tms::TmsProcess** object for its interface to the TMS system. It would use call `getData()` to fetch the required data from the system. There is also an event based data interface implemented using the `requestData()` call and `dataEvent()` event call.

Each of the TMS API calls return an error object. If there is an error, an appropriate error number will be given together with an ASCII string describing the error.

1.3 BEAM class library

The BEAM class library implements some basic low level classes and is used by the TMS API implementation itself. The main class functionality includes:

- **BString** - A simple string class
- **BList** - A templated list class
- **BArray** - A templated array class
- **BError** - An error return class containing an integer and string
- **BSocket** - A Network socket access class
- **BThread** - A thread implementation class
- **BPoll** - A file descriptor event polling class
- **BMutex** - A mutex lock
- **BRWLock** - A read/write lock
- **BSema** - A semaphore
- **BCondInt** - An integer condition class
- **BFile** - A simple file access class
- **BDir** - A simple directory access class
- **BEntry** - A name/value pair list class
- **BNameValue** - A name/value pair class
- **BRtc** - A realtime clock
- **BTimer** - A simple timer class
- **BUrl** - URL access class

1.4 Examples

There are some examples of client applications using the TmsApi in the **tmsExamples** directory of the source code. A simple Data Access client example is listed below:

```

/*****
 *      TmsDataClient.cpp      TMS API example code for a Data Client
 *                          T.Barnaby,      BEAM Ltd,      2007-02-07
 *****/
 *
 *      This is a very basic example of using the TmsApi from a clients perspective.
 *      It is designed to give an overview of using the API.
 */
#include <iostream>
#include <stdio.h>
#include <TmsD.h>
#include <TmsC.h>

using namespace Tms;
using namespace std;

// Function to reads some data
BError tmsTest(TmsProcess& tmsProcess){
    BError      err;
    DataInfo    dataInfo;
    Data        data;
    BUInt32     cn = 0;
    BString     ct;

    // Find out the current cycle number and type
    if(err = tmsProcess.getCycleInfo(cn, ct)){
        return err.set(1, BString("Error: Getting Cycle Number: ") + err.
getString());
    }

    printf("Getting data for cycles starting at cycle: %u\n", cn);

    for(;; cn++){
        // Setup dataInfo
        printf("GetData: Cycle Number: %u\n", cn);
        dataInfo.cycleNumber = cn;
        dataInfo.channel = 1;
        dataInfo.cyclePeriod = CyclePeriodEvent0;
        dataInfo.startTime = 0;
        dataInfo.orbitNumber = 0;
        dataInfo.bunchNumber = 0;
        dataInfo.function = DataFunctionRaw;
        dataInfo.argument = 0;
        dataInfo.numValues = 1024;
        dataInfo.beyondPeriod = 1;

        if(err = tmsProcess.getData(dataInfo, data)){
            return err.set(1, BString("Error: Getting Data: ") + err.
getString());
        }
        printf("Data: NumValues: %d\n", data.numValues);
    }

    return err;
}

int main(int argc, char** argv){
    BError      err;
    BString     host = "localhost";
    TmsProcess  tmsProcess;

    if(argc == 2)
        host = argv[1];

    // Connect to the Process service
    if(err = tmsProcess.connectService(BString("/") + host + "/tmsProcess1"){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }

    // Run a normal data gathering cycle as a normal client would.
    if(err = tmsTest(tmsProcess)){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }

    return 0;
}

```

A simple Control client example is listed below:

```

/*****
 *      TmsControlClient1.cpp  TMS API example code
 *                          T.Barnaby,      BEAM Ltd,      2007-02-07
 *****/
 *
 *      This is a very basic example of using the TmsApi from a clients perspective.
 *      It is designed to give an overview of using the API.
 */
#include <iostream>
#include <stdio.h>
#include <TmsD.h>
#include <TmsC.h>

using namespace Tms;
using namespace std;

const BUInt32    tmsStateNum = 16;
const BUInt32    tmsPickupNum = 40;

// Initialise and test the TMS system
BError tmsInit(TmsControl& tmsControl){
    BError          err;
    ConfigInfo      configInfo;
    BIter           i;
    BList<BError>   errorList;
    BList<NameValue> nvList;
    BString         version;

    // Get Version
    if(err = tmsControl.getVersion(version)){
        return err.set(1, BString("Error: initialising TMS: ") + err.
getString());
    }
    cout << "Version: " << version << "\n";

    // Initialise TMS system
    if(err = tmsControl.init()){
        return err.set(1, BString("Error: initialising TMS: ") + err.
getString());
    }

    // Test TMS system
    if(err = tmsControl.test(errorList)){
        return err.set(1, BString("Error: testing TMS: ") + err.
getString());
    }

    for(errorList.start(i); !errorList.isEnd(i); errorList.next(i)){
        cout << "Warning: " << errorList[i].getString() << "\n";
    }

    // Get Status of TMS system
    if(err = tmsControl.getStatus(nvList)){
        return err.set(1, BString("Error: getting status: ") + err.
getString());
    }

    for(nvList.start(i); !nvList.isEnd(i); nvList.next(i)){
        cout << nvList[i].name << ":\t" << nvList[i].value << "\n";
    }

    return err;
}

int main(int argc, char** argv){
    BError          err;
    BString         host = "localhost";
    TmsControl      tmsControl;
    TmsProcess      tmsProcess;

    if(argc == 2)
        host = argv[1];

    // Connect to the Control service
    if(err = tmsControl.connectService(BString("//") + host + "/tmsControl1")){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }

    // Connect to the Process service
    if(err = tmsProcess.connectService(BString("//") + host + "/tmsProcess1")){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }
}

```



```

// Initialise and test the TMS system. Normally carried out by a configuration
// and test client program.
if(err = tmsInit(tmsControl)){
    cerr << "Error: " << err.getString() << "\n";
    return 1;
}

return 0;
}

```

A simple Control client to set the next cycle information example is listed below:

```

/*****
 *   TmsControlClient2.cpp   TMS API example code
 *   T.Barnaby,           BEAM Ltd,           2007-02-07
 *****/
 *
 *   This is a very basic example of using the TmsApi to set the
 *   TMS's cycleNumber and cycleType.
 *   It is designed to give an overview of using the API.
 */
#include <iostream>
#include <stdio.h>
#include <unistd.h>
#include <TmsD.h>
#include <TmsC.h>

using namespace Tms;
using namespace std;

// Loop sending next cycle information
BError tmsControlLoop(TmsControl& tmsControl){
    BError          err;
    BUInt32         cn = 0;
    BString         ct = "Beam3";

    while(1){
        // Wait for next cycle information
        usleep(1200000);

        // Set next cycle information
        cn = cn + 1;
        ct = "Beam3";

        printf("SendNextCycle\n");
        // Send the next cycle information to the TMS server
        if(err = tmsControl.setNextCycle(cn, ct)){
            cerr << "Error: " << err.getString() << "\n";
        }
    }

    return err;
}

int main(int argc, char** argv){
    BError          err;
    BString         host = "localhost";
    TmsControl      tmsControl;

    if(argc == 2)
        host = argv[1];

    // Connect to the Control service
    if(err = tmsControl.connectService(BString("/") + host + "/tmsControl1")){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }

    // Set the network priority high
    if(err = tmsControl.setPriority(BSocket::PriorityHigh)){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }

    // Set the TmsServer thread priority high
    if(err = tmsControl.setProcessPriority(PriorityHigh)){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }

    if(err = tmsControlLoop(tmsControl)){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }
}

```

```
    return 0;  
}
```

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Boapns	21
Tms	21

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

BAtomic< Type >	28
BAtomicCount	29
BBuffer	30
BBufferStore	32
BoapPacket	117
BComms	35
BCond	37
BCondBool	37
BCondInt	39
BCondResource	41
BCondValue	42
BCondWrap	45
BDate	48
BDebugBacktrace	51
BDictItem< Type >	54
BDuration	58
BEntry	60
BError	66
BEvent1Error	73
BErrorTime	69
BEvent	71
BEvent1	72
BEvent1Error	73
BEvent1Int	73
BEvent1Pipe	75
BEventPipe	76
BFifo< Type >	77
BFifo< BoapMcPacket >	77
BFifoCirc< Type >	81
BFifoCircPos	84
BFile	85
BFileCsv	89
BIter	91
BList< T >	91
BQueue< T >	133
BList< BArray< BString > >	91

BList< BDictItem< Type > >	91
BDict< Type >	52
BConfig	47
BList< BEntry >	91
BEntryList	64
BEntryFile	62
BList< BNameValue< T > >	91
BNameValueList< T >	102
BList< BoapFuncEntry >	91
BList< BoapMcPacket >	91
BQueue< BoapMcPacket >	133
BList< BoapServerConnection * >	91
BList< BoapServiceEntry >	91
BList< BString >	91
BList< BStringList >	91
BFileData	90
BList< struct dirent * >	91
BDir	56
BList< Tms::CycleInformationPeriod >	91
BList< Tms::CycleTypeInfoInformationPeriod >	91
BList< Tms::TmsEvent * >	91
BMutex	97
BStringMutex	163
BMutexLock	99
BMySQL	100
BNameValue< T >	101
BNode	103
BList< T >::Node	200
Boapns::BoapEntry	107
BoapFuncEntry	107
BoapMcClientObject	108
BoapMcComms	110
BoapMcPacket	114
BoapMcPacketHead	114
BoapMcServiceObject	115
BoapMcSignalObject	116
BoapPacketHead	120
BoapServiceEntry	126
BoapServiceObject	127
BObj	130
BObjMember	131
BPoll	131
BRefData	135
BRtc	137
BRWLock	139
BSema	141
BSemaphore	142
BSemaphoreCount	143
BSignal	144
BSocket	146
BoapClientObject	103
Boapns::Boapns	116
Tms::PuControl	202
Tms::PuProcess	206
Tms::TmsControl	219
Tms::TmsEvent	225

Tms::TmsProcess	228
BoapClientObject	103
BoapSignalObject	129
BoapSignalObject	129
BSocketAddress	148
BSocketAddressINET	150
BSpi	152
BString	153
BStringLocked	162
BTable	163
BThread	164
BoapServer	121
BoapServerConnection	125
BRtcThreaded	138
BTime	166
BTimer	169
BTimeStamp	171
BTimeStampMs	175
BUrl	179
Tms::ConfigInfo	180
Tms::CycleInformation	181
Tms::CycleInformationPeriod	182
Tms::CycleParam	183
Tms::CycleParamEdit	188
Tms::CycleParamDb	186
Tms::CycleParamItem	190
Tms::CycleParamState	190
Tms::CycleTypeInfoInformation	193
Tms::CycleTypeInfoInformationPeriod	194
Tms::Data	195
Tms::DataInfo	196
Tms::DataValue	198
map	
BDictMap< Value >	55
Tms::NameValue	199
Tms::PuChannel	201
Tms::PupeConfig	205
Tms::PuStateTable	207
Tms::PuStatus	209
SigGen	210
SigGenBeam	211
SigGenNoise	212
SigGenPulse	213
SigGenSine	214
SigGenSquare	215
Tms::Simulation	216
Tms::TestCaptureInfo	217
Tms::TmsEventServerList	226
Tms::TmsPhase	228
Tms::TmsState	232
vector	
BArray< T >	27
BArray< BError >	27
BArray< BInt32 >	27
BArray< BList< BIter > >	27
BArray< BString >	27
BArray< BUInt8 >	27

BArray< int >	27
BArray< Tms::DataValue >	27
BArray< Tms::PuChannel >	27
BArray< Tms::PuStateTable >	27

Chapter 4

Class Index

4.1 Class List

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

BArray< T >	27
BAtomic< Type >	
BAtomic class	28
BAtomicCount	
BAtomicCount class	29
BBuffer	30
BBufferStore	32
BComms	35
BCond	37
BCondBool	
Thread conditional boolean	37
BCondInt	
Thread conditional value	39
BCondResource	
Resource lock	41
BCondValue	
Thread conditional value	42
BCondWrap	45
BConfig	
This class implements the configuration file access	47
BDate	48
BDebugBacktrace	51
BDict< Type >	52
BDictItem< Type >	
Template based Dictionary class	54
BDictMap< Value >	55
BDir	
File system directory class	56
BDuration	58
BEntry	
Manipulate a name value pair	60
BEntryFile	
File of Entries	62
BEntryList	
List of Entries. Where an entry is a name value pair	64
BError	
Error return class	66

BErrorTime	Error return class	69
BEvent		71
BEvent1	This class provides a base class for all event objects that can be sent over the events interface	72
BEvent1Error		73
BEvent1Int	This class provides an interface for sending simple integer events via a file descriptor. This allows threads to send events that can be picked up by the poll system call	73
BEvent1Pipe	This class provides a base interface for sending events via a pipe. This allows threads to send events that can be picked up by the poll system call	75
BEventPipe	This class provides an interface for sending simple integer events via a pipe file descriptor	76
BFifo< Type >		77
BFifoCirc< Type >	This class implements a thread safe FIFO buffer	81
BFifoCircPos	This class implements a pointer into the Fifo's circular buffer	84
BFile	File operations class	85
BFileCsv		89
BFileData		90
BIter	Iterator for BList	91
BList< T >	Template based list class	91
BMutex	Mutex class	97
BMutexLock		99
BMySQL		100
BNameValue< T >		101
BNameValueList< T >		102
BNode		103
BoapClientObject		103
Boapns::BoapEntry		107
BoapFuncEntry		107
BoapMcClientObject		108
BoapMcComms		110
BoapMcPacket		114
BoapMcPacketHead		114
BoapMcServiceObject		115
BoapMcSignalObject		116
Boapns::Boapns		116
BoapPacket		117
BoapPacketHead		120
BoapServer		121
BoapServerConnection		125
BoapServiceEntry		126
BoapServiceObject		127
BoapSignalObject		129
BObj		130
BObjMember		131
BPoll	This class provides an interface for polling a number of file descriptors. It uses round robin polling	131
BQueue< T >	Queue class	133
BRefData		135

BRtc	Realtime clock	137
BRtcThreaded	Threaded real time clock	138
BRWLock	Thread read-write locks	139
BSema	Sempahore class	141
BSemaphore	Semaphore class	142
BSemaphoreCount		143
BSignal		144
BSocket		146
BSocketAddress	Socket Address	148
BSocketAddressINET	IP aware socket address	150
BSpi	BSpi class	152
BString		153
BStringLocked		162
BStringMutex		163
BTable		163
BThread		164
BTime		166
BTimer	Stopwatch style timer	169
BTimeStamp		171
BTimeStampMs		175
BUrl	Basic access to a Url	179
Tms::ConfigInfo	This class describes the configuration of the TMS	180
Tms::CycleInformation		181
Tms::CycleInformationPeriod	Cycle information	182
Tms::CycleParam	This class defines the parameters for a PS processing cycle	183
Tms::CycleParamDb	Internal CycleParameter management class	186
Tms::CycleParamEdit	Cycle Parameter management class	188
Tms::CycleParamItem		190
Tms::CycleParamState		190
Tms::CycleTypeInfo		193
Tms::CycleTypeInfoPeriod	Cycle Type information	194
Tms::Data	This class stores the raw data	195
Tms::DataInfo	This class defines the data to be acquired and/or fetched	196
Tms::DataValue	This is the definition of a single data value	198
Tms::NameValue		199
BList< T >::Node		200
Tms::PuChannel	This class stores a Physical Pick-Up channel id	201

Tms::PuControl	
This class defines the parameters for a test data capture	202
Tms::PupeConfig	205
Tms::PuProcess	
This interface provides functions to configure and capture data from individual pick-up	206
Tms::PuStateTable	
This class defines the Pick-Up state table	207
Tms::PuStatus	
This class stores the status of an individual Pick-Up	209
SigGen	210
SigGenBeam	211
SigGenNoise	212
SigGenPulse	213
SigGenSine	214
SigGenSquare	215
Tms::Simulation	216
Tms::TestCaptureInfo	
This class defines the parameters for a test data capture	217
Tms::TmsControl	
This interface provides functions to control, test and get statistics from the TMS as a whole	219
Tms::TmsEvent	
This interface provides functions for events to be sent to clients from the TMS as a whole	225
Tms::TmsEventServerList	226
Tms::TmsPhase	
The Tms Phase Table Entry	228
Tms::TmsProcess	
This interface provides functions to capture data from the TMS as a whole	228
Tms::TmsState	
The Tms State entry	232

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

/src/cern/tms/beam/libBeam/BArray.h	235
/src/cern/tms/beam/libBeam/BAtomic.h	235
/src/cern/tms/beam/libBeam/BAtomicCount.h	236
/src/cern/tms/beam/libBeam/BBuffer.cpp	236
/src/cern/tms/beam/libBeam/BBuffer.h	236
/src/cern/tms/beam/libBeam/BComms.cpp	237
/src/cern/tms/beam/libBeam/BComms.h	237
/src/cern/tms/beam/libBeam/BComplex.h	237
/src/cern/tms/beam/libBeam/BCond.cpp	238
/src/cern/tms/beam/libBeam/BCond.h	238
/src/cern/tms/beam/libBeam/BCondInt.cpp	238
/src/cern/tms/beam/libBeam/BCondInt.h	238
/src/cern/tms/beam/libBeam/BConfig.cpp	239
/src/cern/tms/beam/libBeam/BConfig.h	239
/src/cern/tms/beam/libBeam/BCrc16.cpp	239
/src/cern/tms/beam/libBeam/BCrc16.h	240
/src/cern/tms/beam/libBeam/BDate.cpp	240
/src/cern/tms/beam/libBeam/BDate.h	241
/src/cern/tms/beam/libBeam/BDebug.cpp	242
/src/cern/tms/beam/libBeam/BDebug.h	243
/src/cern/tms/beam/libBeam/BDict.cpp	244
/src/cern/tms/beam/libBeam/BDict.h	245
/src/cern/tms/beam/libBeam/BDictMap.h	245
/src/cern/tms/beam/libBeam/BDir.cpp	246
/src/cern/tms/beam/libBeam/BDir.h	246
/src/cern/tms/beam/libBeam/BDuration.cpp	246
/src/cern/tms/beam/libBeam/BDuration.h	247
/src/cern/tms/beam/libBeam/BEndian.cpp	247
/src/cern/tms/beam/libBeam/BEndian.h	247
/src/cern/tms/beam/libBeam/BEntry.cpp	250
/src/cern/tms/beam/libBeam/BEntry.h	250
/src/cern/tms/beam/libBeam/BError.cpp	251
/src/cern/tms/beam/libBeam/BError.h	251
/src/cern/tms/beam/libBeam/BErrorTime.cpp	252
/src/cern/tms/beam/libBeam/BErrorTime.h	252
/src/cern/tms/beam/libBeam/BEvent.cpp	252
/src/cern/tms/beam/libBeam/BEvent.h	252
/src/cern/tms/beam/libBeam/BEvent1.cpp	253

/src/cern/tms/beam/libBeam/BEvent1.h	253
/src/cern/tms/beam/libBeam/BFifo.h	254
/src/cern/tms/beam/libBeam/BFifo.inc	254
/src/cern/tms/beam/libBeam/BFifoCirc.cpp	254
/src/cern/tms/beam/libBeam/BFifoCirc.h	254
/src/cern/tms/beam/libBeam/BFifoCirc.inc	255
/src/cern/tms/beam/libBeam/BFile.cpp	255
/src/cern/tms/beam/libBeam/BFile.h	255
/src/cern/tms/beam/libBeam/BFileCsv.cpp	255
/src/cern/tms/beam/libBeam/BFileCsv.h	256
/src/cern/tms/beam/libBeam/BFileData.cpp	256
/src/cern/tms/beam/libBeam/BFileData.h	256
/src/cern/tms/beam/libBeam/BList.h	256
/src/cern/tms/beam/libBeam/BList_func.h	257
/src/cern/tms/beam/libBeam/BMutex.cpp	257
/src/cern/tms/beam/libBeam/BMutex.h	257
/src/cern/tms/beam/libBeam/BMysql.cpp	257
/src/cern/tms/beam/libBeam/BMysql.h	257
/src/cern/tms/beam/libBeam/BNameValue.h	258
/src/cern/tms/beam/libBeam/Boap.cpp	258
/src/cern/tms/beam/libBeam/Boap.h	259
/src/cern/tms/beam/libBeam/BoapMc.cpp	260
/src/cern/tms/beam/libBeam/BoapMc.h	261
/src/cern/tms/beam/libBeam/BoapnsC.cpp	262
/src/cern/tms/beam/libBeam/BoapnsC.h	262
/src/cern/tms/beam/libBeam/BoapnsD.cpp	263
/src/cern/tms/beam/libBeam/BoapnsD.h	263
/src/cern/tms/beam/libBeam/BoapSimple.cc	263
/src/cern/tms/beam/libBeam/BoapSimple.h	264
/src/cern/tms/beam/libBeam/BObj.cpp	265
/src/cern/tms/beam/libBeam/BObj.h	265
/src/cern/tms/beam/libBeam/BObjStringFormat.cpp	266
/src/cern/tms/beam/libBeam/BObjStringFormat.h	268
/src/cern/tms/beam/libBeam/BPoll.cpp	270
/src/cern/tms/beam/libBeam/BPoll.h	270
/src/cern/tms/beam/libBeam/BQueue.h	271
/src/cern/tms/beam/libBeam/BRefData.cpp	271
/src/cern/tms/beam/libBeam/BRefData.h	271
/src/cern/tms/beam/libBeam/BRtc.cpp	272
/src/cern/tms/beam/libBeam/BRtc.h	272
/src/cern/tms/beam/libBeam/BRWLock.cpp	272
/src/cern/tms/beam/libBeam/BRWLock.h	272
/src/cern/tms/beam/libBeam/BSema.cpp	272
/src/cern/tms/beam/libBeam/BSema.h	273
/src/cern/tms/beam/libBeam/BSemaphore.cpp	273
/src/cern/tms/beam/libBeam/BSemaphore.h	273
/src/cern/tms/beam/libBeam/BSocket.cpp	273
/src/cern/tms/beam/libBeam/BSocket.h	274
/src/cern/tms/beam/libBeam/BSpi.cpp	274
/src/cern/tms/beam/libBeam/BSpi.h	274
/src/cern/tms/beam/libBeam/BString.cpp	275
/src/cern/tms/beam/libBeam/BString.h	277
/src/cern/tms/beam/libBeam/BStringLocked.h	278
/src/cern/tms/beam/libBeam/BTable.cpp	278
/src/cern/tms/beam/libBeam/BTable.h	278
/src/cern/tms/beam/libBeam/BThread.cpp	278
/src/cern/tms/beam/libBeam/BThread.h	279
/src/cern/tms/beam/libBeam/BTime.cpp	279

/src/cern/tms/beam/libBeam/BTime.h	279
/src/cern/tms/beam/libBeam/BTimer.cpp	280
/src/cern/tms/beam/libBeam/BTimer.h	280
/src/cern/tms/beam/libBeam/BTimeStamp.cpp	280
/src/cern/tms/beam/libBeam/BTimeStamp.h	281
/src/cern/tms/beam/libBeam/BTimeStampMs.cpp	281
/src/cern/tms/beam/libBeam/BTimeStampMs.h	281
/src/cern/tms/beam/libBeam/BTypes.h	282
/src/cern/tms/beam/libBeam/BUrl.cpp	285
/src/cern/tms/beam/libBeam/BUrl.h	285
SigGen.cpp	285
SigGen.h	285
test1.cpp	286
TmsC.cc	286
TmsC.h	
This file contains the TmsAPI class definitions	287
TmsCycleParam.cc	287
TmsCycleParam.h	288
TmsD.cc	288
TmsD.h	288
TmsEventServerList.cc	289
TmsEventServerList.h	290
TmsLib.cc	290
TmsLib.h	290
TmsS.cc	291
TmsT.cc	291

Chapter 6

Namespace Documentation

6.1 Boapns Namespace Reference

Classes

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

Variables

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

6.1.1 Variable Documentation

- 6.1.1.1 const [BUInt32](#) `Boapns::apiVersion` = 0

6.2 Tms Namespace Reference

Classes

- class [PuControl](#)
This class defines the parameters for a test data capture.
- class [PuProcess](#)
This interface provides functions to configure and capture data from individual pick-up.
- class [TmsControl](#)
This interface provides functions to control, test and get statistics from the TMS as a whole.
- class [TmsProcess](#)
This interface provides functions to capture data from the TMS as a whole.
- class [TmsEvent](#)
This interface provides functions for events to be sent to clients from the TMS as a whole.
- class [CycleParamState](#)
- class [CycleParamEdit](#)
Cycle Parameter management class.
- class [NameValue](#)
- class [PuChannel](#)
This class stores a Physical Pick-Up channel id.
- class [PuStatus](#)

- This class stores the status of an individual Pick-Up.*

 - class [ConfigInfo](#)
 - This class describes the configuration of the TMS.*
 - class [DataInfo](#)
 - This class defines the data to be acquired and/or fetched.*
 - class [DataValue](#)
 - This is the definition of a single data value.*
 - class [Data](#)
 - This class stores the raw data.*
 - class [PuStateTable](#)
 - This class defines the Pick-Up state table.*
 - class [CycleParam](#)
 - This class defines the parameters for a PS processing cycle.*
 - class [CycleParamItem](#)
 - class [TestCaptureInfo](#)
 - This class defines the parameters for a test data capture.*
 - class [PupeConfig](#)
 - class [CycleInformationPeriod](#)
 - Cycle information.*
 - class [CycleInformation](#)
 - class [CycleTypeInfoPeriod](#)
 - Cycle Type information.*
 - class [CycleTypeInfo](#)
 - class [Simulation](#)
 - class [TmsEventServerList](#)
 - union [TmsState](#)
 - The Tms State entry.*
 - union [TmsPhase](#)
 - The Tms Phase Table Entry.*
 - class [CycleParamDb](#)
 - Internal CycleParameter management class.*

Enumerations

- enum [CyclePeriod](#) {
[CyclePeriodAll](#), [CyclePeriodCalibration](#), [CyclePeriodEvent0](#), [CyclePeriodEvent1](#),
[CyclePeriodEvent2](#), [CyclePeriodEvent3](#), [CyclePeriodEvent4](#), [CyclePeriodEvent5](#),
[CyclePeriodEvent6](#), [CyclePeriodEvent7](#), [CyclePeriodEvent8](#), [CyclePeriodEvent9](#) }
- enum [DataType](#) { [DataTypeRaw](#) }
- enum [DataFunction](#) {
[DataFunctionRaw](#), [DataFunctionMean](#), [DataFunctionMeanAll](#), [DataFunctionMean0](#),
[DataFunctionMean1](#) }
- enum [TestOutput](#) { [TestOutputFrefLocal](#), [TestOutputPIIL1](#), [TestOutputPIIL2](#) }
- enum [Priority](#) { [PriorityLow](#), [PriorityNormal](#), [PriorityHigh](#) }
- enum [TimingSig](#) {
[TimingSigClock](#) = 0x01, [TimingSigCycleStart](#) = 0x02, [TimingSigCycleStop](#) = 0x04, [TimingSigCalStart](#) = 0x08,
[TimingSigCalStop](#) = 0x10, [TimingSigInjection](#) = 0x20, [TimingSigHChange](#) = 0x40, [TimingSigFRef](#) = 0x80 }
 - The timing signal bits.*
- enum [CaptureClock](#) {
[ClkAdcDiv_1](#) = 0x00, [ClkAdcDiv_2](#) = 0x01, [ClkAdcDiv_5](#) = 0x02, [ClkAdcDiv_10](#) = 0x03,
[ClkAdcDiv_20](#) = 0x04, [ClkAdcDiv_50](#) = 0x05, [ClkAdcDiv_100](#) = 0x06, [ClkAdcDiv_200](#) = 0x07,
[ClkAdcDiv_500](#) = 0x08, [ClkAdcDiv_1000](#) = 0x09, [ClkAdcDiv_2000](#) = 0x0A, [ClkAdcDiv_5000](#) = 0x0B,
[ClkAdcDiv_10000](#) = 0x0C, [ClkAdcDiv_20000](#) = 0x0D, [ClkAdcDiv_50000](#) = 0x0E, [ClkAdcDiv_100000](#) = 0x0F,
[ClkMs](#) = 0x10, [ClkFref](#) = 0x11 }
 - The Diagnostics Capture Clock settings.*

Variables

- const BUInt32 `apiVersion` = 0
- const unsigned int `tmsNumPickups` = 40
The default number of pickups.
- const unsigned int `tmsPhaseTableSize` = 512
The size of the Phase Table.

6.2.1 Enumeration Type Documentation

6.2.1.1 enum Tms::CaptureClock

The Diagnostics Capture Clock settings.

Enumerator

ClkAdcDiv_1 ADC Clock.
ClkAdcDiv_2 ADC Clock divided by 2.
ClkAdcDiv_5 ADC Clock divided by 5.
ClkAdcDiv_10 ADC Clock divided by 10.
ClkAdcDiv_20 ADC Clock divided by 20.
ClkAdcDiv_50 ADC Clock divided by 50.
ClkAdcDiv_100 ADC Clock divided by 100.
ClkAdcDiv_200 ADC Clock divided by 200.
ClkAdcDiv_500 ADC Clock divided by 500.
ClkAdcDiv_1000 ADC Clock divided by 1000.
ClkAdcDiv_2000 ADC Clock divided by 2000.
ClkAdcDiv_5000 ADC Clock divided by 5000.
ClkAdcDiv_10000 ADC Clock divided by 10000.
ClkAdcDiv_20000 ADC Clock divided by 20000.
ClkAdcDiv_50000 ADC Clock divided by 50000.
ClkAdcDiv_100000 ADC Clock divided by 100000.
ClkMs Millisecond Clock.
ClkFref FREF.

6.2.1.2 enum Tms::CyclePeriod

Enumerator

CyclePeriodAll
CyclePeriodCalibration
CyclePeriodEvent0
CyclePeriodEvent1
CyclePeriodEvent2
CyclePeriodEvent3
CyclePeriodEvent4
CyclePeriodEvent5
CyclePeriodEvent6
CyclePeriodEvent7
CyclePeriodEvent8
CyclePeriodEvent9

6.2.1.3 enum Tms::DataFunction

Enumerator

DataFunctionRaw
DataFunctionMean
DataFunctionMeanAll
DataFunctionMean0
DataFunctionMean1

6.2.1.4 enum Tms::DataType

Enumerator

DataTypeRaw

6.2.1.5 enum Tms::Priority

Enumerator

PriorityLow
PriorityNormal
PriorityHigh

6.2.1.6 enum Tms::TestOutput

Enumerator

TestOutputFrefLocal
TestOutputPIIL1
TestOutputPIIL2

6.2.1.7 enum Tms::TimingSig

The timing signal bits.

Enumerator

TimingSigClock 10MHz System Clock
TimingSigCycleStart CYCLE_START event.
TimingSigCycleStop CYCLE_STOP event.
TimingSigCalStart CAL_START event.
TimingSigCalStop CAL_STOP event.
TimingSigInjection INJECTION event.
TimingSigHChange HCHANGE event.
TimingSigFRef FREF signal.

6.2.2 Variable Documentation

6.2.2.1 `const BUInt32 Tms::apiVersion = 0`

6.2.2.2 `const unsigned int Tms::tmsNumPickups = 40`

The default number of pick ups.

6.2.2.3 `const unsigned int Tms::tmsPhaseTableSize = 512`

The size of the Phase Table.

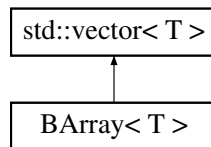
Chapter 7

Class Documentation

7.1 BArray< T > Class Template Reference

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

Inheritance diagram for BArray< T >:



Public Types

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

Public Member Functions

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

7.1.1 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.

7.1.2 Member Typedef Documentation

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

Prototype for sorting function.

7.1.3 Constructor & Destructor Documentation

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

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

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

7.1.4 Member Function Documentation

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

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

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

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

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

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

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

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

- [/src/cern/tms/beam/libBeam/BArray.h](#)

7.2 BAtomic< Type > Class Template Reference

[BAtomic](#) class.

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

Public Member Functions

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

Private Attributes

- Type [ovalue](#)

7.2.1 Detailed Description

```
template<class Type>class BAtomic< Type >
```

[BAtomic](#) class.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 `template<class Type > BAtomic< Type >::BAtomic (Type value = 0) [inline]`

7.2.3 Member Function Documentation

7.2.3.1 `template<class Type > Type BAtomic< Type >::add (long value) [inline]`

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

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

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

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

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

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

7.2.4 Member Data Documentation

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

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

- `/src/cern/tms/beam/libBeam/BAtomic.h`

7.3 BAtomicCount Class Reference

[BAtomicCount](#) class.

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

Public Member Functions

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

Private Attributes

- `_Atomic_word` [ovalue](#)

7.3.1 Detailed Description

[BAtomicCount](#) class.

7.3.2 Constructor & Destructor Documentation

7.3.2.1 [BAtomicCount::BAtomicCount \(long value = 0 \)](#) [inline]

7.3.3 Member Function Documentation

7.3.3.1 [long BAtomicCount::add \(long value \)](#) [inline]

7.3.3.2 [long BAtomicCount::getValue \(\) const](#) [inline]

7.3.3.3 [BAtomicCount::operator long \(\) const](#) [inline]

7.3.3.4 [long BAtomicCount::operator++ \(int \)](#) [inline]

7.3.3.5 [long BAtomicCount::operator++ \(\)](#) [inline]

7.3.3.6 [long BAtomicCount::operator-- \(int \)](#) [inline]

7.3.3.7 [long BAtomicCount::operator-- \(\)](#) [inline]

7.3.4 Member Data Documentation

7.3.4.1 [_Atomic_word BAtomicCount::ovalue](#) [mutable],[private]

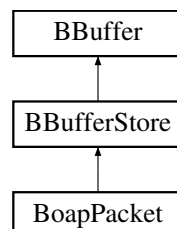
The documentation for this class was generated from the following file:

- [/src/cern/tms/beam/libBeam/BAtomicCount.h](#)

7.4 BBuffer Class Reference

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

Inheritance diagram for BBuffer:



Public Member Functions

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

Sets the bufer size.

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

Sets buffer data resized to contain the data.

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

Writes data into buffer from offset pos.

- char * [data](#) ()

The data.

- [BUInt32 size](#) ()

Size of the buffer in bytes.

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

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

Protected Attributes

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

7.4.1 Constructor & Destructor Documentation

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

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

7.4.1.2 BBuffer::~~BBuffer ()

7.4.2 Member Function Documentation

7.4.2.1 char * BBuffer::data ()

The data.

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

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

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

Sets buffer data resized to contain the data.

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

Sets the bufer size.

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

Size of the buffer in bytes.

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

Writes data into buffer from offset pos.

7.4.3 Member Data Documentation

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

7.4.3.2 `BUInt32 BBuffer::odataSize` [protected]

7.4.3.3 `BUInt32 BBuffer::osize` [protected]

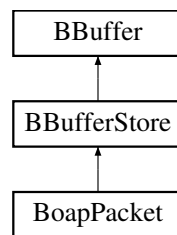
The documentation for this class was generated from the following files:

- [/src/cern/tms/beam/libBeam/BBuffer.h](#)
- [/src/cern/tms/beam/libBeam/BBuffer.cpp](#)

7.5 BBufferStore Class Reference

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

Inheritance diagram for BBufferStore:



Public Member Functions

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

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

Protected Attributes

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

7.5.1 Constructor & Destructor Documentation

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

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

7.5.2 Member Function Documentation

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

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

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

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

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

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

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

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

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

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

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

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

- 7.5.2.13 `int BBufferStore::pop (BString & v)`
- 7.5.2.14 `int BBufferStore::pop (BError & v)`
- 7.5.2.15 `int BBufferStore::pop (BTimeStamp & v)`
- 7.5.2.16 `int BBufferStore::pop (BComplex & v)`
- 7.5.2.17 `int BBufferStore::pop (BUInt32 nBytes, void * data, const char * swapType = "1")`
- 7.5.2.18 `int BBufferStore::push (Bint8 v)`
- 7.5.2.19 `int BBufferStore::push (BUInt8 v)`
- 7.5.2.20 `int BBufferStore::push (Bint16 v)`
- 7.5.2.21 `int BBufferStore::push (BUInt16 v)`
- 7.5.2.22 `int BBufferStore::push (Bint32 v)`
- 7.5.2.23 `int BBufferStore::push (BUInt32 v)`
- 7.5.2.24 `int BBufferStore::push (Bint64 v)`
- 7.5.2.25 `int BBufferStore::push (BUInt64 v)`
- 7.5.2.26 `int BBufferStore::push (BFloat32 v)`
- 7.5.2.27 `int BBufferStore::push (BFloat64 v)`
- 7.5.2.28 `int BBufferStore::push (const BString & v)`
- 7.5.2.29 `int BBufferStore::push (const BError & v)`
- 7.5.2.30 `int BBufferStore::push (const BTimeStamp & v)`
- 7.5.2.31 `int BBufferStore::push (const BComplex & v)`
- 7.5.2.32 `int BBufferStore::push (BUInt32 nBytes, const void * data, const char * swapType = "1")`
- 7.5.2.33 `void BBufferStore::setHexString (BString s)`
- 7.5.2.34 `void BBufferStore::setPos (BUInt32 pos)`

7.5.3 Member Data Documentation

- 7.5.3.1 `BUInt32 BBufferStore::opos` [protected]
- 7.5.3.2 `int BBufferStore::oswapBytes` [protected]

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

- [/src/cern/tms/beam/libBeam/BBuffer.h](#)
- [/src/cern/tms/beam/libBeam/BBuffer.cpp](#)

7.6 BComms Class Reference

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

Public Types

- enum `Wait` { `WaitNone` = 0x00, `WaitRead` = 0x01, `WaitWrite` = 0x02, `WaitError` = 0x04 }

Public Member Functions

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

Protected Attributes

- `Bool` `opacketMode`
- `BInt32` `otimeout`
- `BEventQueue *` `oeventQueue`
- `BInt32` `oevent`
- `BUInt` `oeventNum`

7.6.1 Member Enumeration Documentation

7.6.1.1 enum `BComms::Wait`

Enumerator

WaitNone

WaitRead

WaitWrite

WaitError

7.6.2 Constructor & Destructor Documentation

7.6.2.1 `BComms::BComms ()`

7.6.2.2 `BComms::~~BComms ()` [virtual]

7.6.3 Member Function Documentation

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

7.6.3.2 `BError BComms::init ()` [virtual]

7.6.3.3 `Bool BComms::packetMode ()` [virtual]

Device is in packet mode.

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

7.6.3.5 `BUInt BComms::readAvailable ()` [virtual]

7.6.3.6 `BError BComms::setPacketMode (Bool packetMode)` [virtual]

Set packet mode.

7.6.3.7 `BError BComms::setTimeout (BInt timeoutMs)` [virtual]

Set communication timeout.

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

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

7.6.3.10 `BUInt BComms::writeAvailable ()` [virtual]

7.6.4 Member Data Documentation

7.6.4.1 `BInt32 BComms::oevent` [protected]

7.6.4.2 `BUInt BComms::oeventNum` [protected]

7.6.4.3 `BEventQueue* BComms::oeventQueue` [protected]

7.6.4.4 `Bool BComms::opacketMode` [protected]

7.6.4.5 `BInt32 BComms::otimeout` [protected]

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

- [/src/cern/tms/beam/libBeam/BComms.h](#)
- [/src/cern/tms/beam/libBeam/BComms.cpp](#)

7.7 BCond Class Reference

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

Public Member Functions

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

Private Attributes

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

7.7.1 Constructor & Destructor Documentation

7.7.1.1 BCond::BCond ()

Thread conditional variable.

7.7.1.2 BCond::~~BCond ()

7.7.2 Member Function Documentation

7.7.2.1 int BCond::signal ()

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

7.7.2.3 int BCond::wait ()

7.7.3 Member Data Documentation

7.7.3.1 [pthread_cond_t BCond::ocond](#) [private]

7.7.3.2 [pthread_mutex_t BCond::omutex](#) [private]

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

- [/src/cern/tms/beam/libBeam/BCond.h](#)
- [/src/cern/tms/beam/libBeam/BCond.cpp](#)

7.8 BCondBool Class Reference

Thread conditional boolean.

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

Public Member Functions

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

Private Attributes

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

7.8.1 Detailed Description

Thread conditional boolean.

7.8.2 Constructor & Destructor Documentation

7.8.2.1 [BCondBool::BCondBool](#) ()

7.8.2.2 [BCondBool::~~BCondBool](#) ()

7.8.3 Member Function Documentation

7.8.3.1 [int BCondBool::clear](#) ()

Clear Value.

7.8.3.2 [BCondBool::operator int](#) () `[inline]`

7.8.3.3 [int BCondBool::set](#) ()

Set value. Wakes waiting.

7.8.3.4 [int BCondBool::timedWait](#) (int *timeOutUs*)

Wait until set, with timeout.

7.8.3.5 [int BCondBool::value](#) ()

Current value.

7.8.3.6 int BCondBool::wait ()

Wait until value is true.

7.8.4 Member Data Documentation

7.8.4.1 pthread_cond_t BCondBool::ocond [private]

7.8.4.2 pthread_mutex_t BCondBool::omutex [private]

7.8.4.3 int BCondBool::ovalue [private]

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

- /src/cern/tms/beam/libBeam/BCondInt.h
- /src/cern/tms/beam/libBeam/BCondInt.cpp

7.9 BCondInt Class Reference

Thread conditional value.

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

Public Member Functions

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

Private Attributes

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

7.9.1 Detailed Description

Thread conditional value.

7.9.2 Constructor & Destructor Documentation

7.9.2.1 BCondInt::BCondInt ()

7.9.2.2 BCondInt::~~BCondInt ()

7.9.3 Member Function Documentation

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

Decrement. Wakes waiting.

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

Increment. Wakes waiting.

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

Increment value. Wakes waiting.

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

Add to value. Wakes waiting.

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

Decrement value. Wakes waiting.

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

Subtract from value. Wakes waiting.

7.9.3.7 void BCondInt::setValue (BInt value)

Set the value. Wakes waiting.

7.9.3.8 BInt BCondInt::value () const

Current value.

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

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

7.9.3.10 **Bool** BCondInt::waitLessThanOrEqualTo (**BInt** *v*, **Bool** *increment* = 0, **BTimeout** *timeoutUs* = **BTimeoutForever**)

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

7.9.3.11 **Bool** BCondInt::waitMoreThanOrEqualTo (**BInt** *v*, **Bool** *decrement* = 0, **BTimeout** *timeoutUs* = **BTimeoutForever**)

Wait until value is at least the value given.

7.9.4 Member Data Documentation

7.9.4.1 `pthread_cond_t` BCondInt::ocond [private]

7.9.4.2 `pthread_mutex_t` BCondInt::omutex [private]

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

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

- /src/cern/tms/beam/libBeam/BCondInt.h
- /src/cern/tms/beam/libBeam/BCondInt.cpp

7.10 BCondResource Class Reference

Resource lock.

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

Public Member Functions

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

Private Attributes

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

7.10.1 Detailed Description

Resource lock.

7.10.2 Constructor & Destructor Documentation

7.10.2.1 `BCondResource::BCondResource ()`

7.10.2.2 `BCondResource::~~BCondResource ()`

7.10.3 Member Function Documentation

7.10.3.1 `int BCondResource::end ()`

Finish using the resource.

7.10.3.2 `int BCondResource::inUse ()`

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

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

7.10.3.4 `int BCondResource::locked ()`

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

Start using the resource.

7.10.3.6 `int BCondResource::unlock ()`

Unlock the resource.

7.10.4 Member Data Documentation

7.10.4.1 `pthread_cond_t BCondResource::ocond [private]`

7.10.4.2 `int BCondResource::olock [private]`

7.10.4.3 `pthread_mutex_t BCondResource::omutex [private]`

7.10.4.4 `int BCondResource::ouse [private]`

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

- [/src/cern/tms/beam/libBeam/BCondInt.h](#)
- [/src/cern/tms/beam/libBeam/BCondInt.cpp](#)

7.11 BCondValue Class Reference

Thread conditional value.

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

Public Member Functions

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

Private Attributes

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

7.11.1 Detailed Description

Thread conditional value.

7.11.2 Constructor & Destructor Documentation

7.11.2.1 [BCondValue::BCondValue](#) ()

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

7.11.3 Member Function Documentation

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

Decrement. Wakes waiting.

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

Increment. Wakes waiting.

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

Increment value. Wakes waiting.

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

Add to value. Wakes waiting.

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

Decrement value. Wakes waiting.

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

Subtract from value. Wakes waiting.

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

Set the value. Wakes waiting.

7.11.3.8 `int BCondValue::value()`

Current value.

7.11.3.9 `int BCondValue::waitLessThan(int v, int timeoutUs = 0)`

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

7.11.3.10 `int BCondValue::waitLessThanOrEqualTo(int v, int increment = 0, int timeoutUs = 0)`

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

7.11.3.11 `int BCondValue::waitMoreThanOrEqualTo(int v, int decrement = 0, int timeoutUs = 0)`

Wait until value is at least the value given.

7.11.4 Member Data Documentation

7.11.4.1 `pthread_cond_t BCondValue::ocond [private]`

7.11.4.2 `pthread_mutex_t BCondValue::omutex [private]`

7.11.4.3 `int BCondValue::ovalue [private]`

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

- [/src/cern/tms/beam/libBeam/BCondInt.h](#)
- [/src/cern/tms/beam/libBeam/BCondInt.cpp](#)

7.12 BCondWrap Class Reference

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

Public Member Functions

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

Private Member Functions

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

Private Attributes

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

7.12.1 Constructor & Destructor Documentation

7.12.1.1 [BCondWrap::BCondWrap](#) ()

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

7.12.2 Member Function Documentation

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

Decrement. Wakes waiting.

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

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

Increment. Wakes waiting.

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

Increment value. Wakes waiting.

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

Add to value. Wakes waiting.

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

Decrement value. Wakes waiting.

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

Subtract from value. Wakes waiting.

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

Set the value. Wakes waiting.

7.12.2.9 `uint32_t BCondWrap::value ()`

Current value.

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

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

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

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

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

Wait until value is at least the value given.

7.12.3 Member Data Documentation

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

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

7.12.3.3 uint32_t BCondWrap::ovalue [private]

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

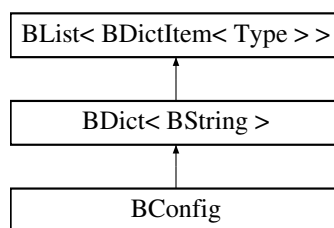
- [/src/cern/tms/beam/libBeam/BCondInt.h](#)
- [/src/cern/tms/beam/libBeam/BCondInt.cpp](#)

7.13 BConfig Class Reference

This class implements the configuration file access.

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

Inheritance diagram for BConfig:



Public Member Functions

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

Private Attributes

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

Additional Inherited Members

7.13.1 Detailed Description

This class implements the configuration file access.

7.13.2 Member Function Documentation

7.13.2.1 [void BConfig::close](#) ()

7.13.2.2 [BString BConfig::fileName](#) ()

7.13.2.3 [BString BConfig::findValue](#) ([BString name](#))

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

7.13.2.5 **BError** **BConfig::read** ()

7.13.2.6 **BError** **BConfig::write** ()

7.13.3 Member Data Documentation

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

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

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

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

- [/src/cern/tms/beam/libBeam/BConfig.h](#)
- [/src/cern/tms/beam/libBeam/BConfig.cpp](#)

7.14 BDate Class Reference

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

Public Member Functions

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

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

Static Public Member Functions

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

Public Attributes

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

7.14.1 Constructor & Destructor Documentation

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

7.14.1.2 `BDate::BDate` (`BString` str)

7.14.1.3 `BDate::~~BDate` ()

7.14.2 Member Function Documentation

7.14.2.1 void `BDate::clear` ()

Clear the date/time.

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

Compare two dates.

7.14.2.3 int `BDate::day` ()

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

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

7.14.2.6 `BString` `BDate::getString` ()

Get the time as an ISO date/time string.

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

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

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

7.14.2.9 int BDate::isSet () [inline]

7.14.2.10 int BDate::month ()

7.14.2.11 BDate::operator BString () [inline]

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

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

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

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

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

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

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

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

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

7.14.2.20 void BDate::setFirst ()

Set the first date available.

7.14.2.21 void BDate::setLast ()

Set the last date available.

7.14.2.22 void BDate::setNow ()

Set the timeStamp to now.

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

Set the time from an ISO date/time.

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

7.14.2.25 int BDate::yday ()

7.14.2.26 int BDate::year ()

7.14.3 Member Data Documentation

7.14.3.1 uint16_t BDate::oyday

Day in year (0 .. 365)

7.14.3.2 uint16_t BDate::oyear

Year (0 .. 65535)

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

- [/src/cern/tms/beam/libBeam/BDate.h](#)
- [/src/cern/tms/beam/libBeam/BDate.cpp](#)

7.15 BDebugBacktrace Class Reference

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

Public Member Functions

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

7.15.1 Constructor & Destructor Documentation

7.15.1.1 [BDebugBacktrace::BDebugBacktrace](#) ()

7.15.1.2 [BDebugBacktrace::~~BDebugBacktrace](#) ()

7.15.2 Member Function Documentation

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

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

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

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

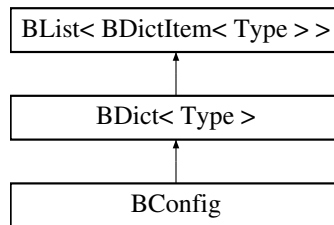
The documentation for this class was generated from the following files:

- [/src/cern/tms/beam/libBeam/BDebug.h](#)
- [/src/cern/tms/beam/libBeam/BDebug.cpp](#)

7.16 `BDict< Type >` Class Template Reference

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

Inheritance diagram for `BDict< Type >`:



Public Types

- typedef [Blter](#) iterator

Public Member Functions

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

Private Member Functions

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

Private Attributes

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

Additional Inherited Members

7.16.1 Member Typedef Documentation

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

7.16.2 Constructor & Destructor Documentation

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

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

7.16.3 Member Function Documentation

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

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

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

Clear the list.

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

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

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

Delete specified item.

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

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

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

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

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

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

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

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

Insert item before item.

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

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

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

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

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

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

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

7.16.4 Member Data Documentation

7.16.4.1 `template<class Type > BArray<BList<BIter> > BDict< Type >::ohashLists [private]`

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

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

- [/src/cern/tms/beam/libBeam/BDict.h](#)

7.17 BDictItem< Type > Class Template Reference

Template based Dictionary class.

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

Public Member Functions

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

Public Attributes

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

7.17.1 Detailed Description

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

Template based Dictionary class.

7.17.2 Constructor & Destructor Documentation

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

7.17.3 Member Data Documentation

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

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

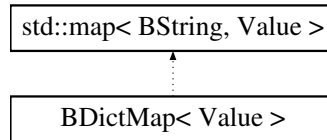
The documentation for this class was generated from the following file:

- [/src/cern/tms/beam/libBeam/BDict.h](#)

7.18 BDictMap< Value > Class Template Reference

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

Inheritance diagram for BDictMap< Value >:



Public Types

- typedef `BDictMap< Value >::iterator` `iterator`

Public Member Functions

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

7.18.1 Detailed Description

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

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

7.18.2 Member Typedef Documentation

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

7.18.3 Member Function Documentation

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

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

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

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

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

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

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

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

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

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

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

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

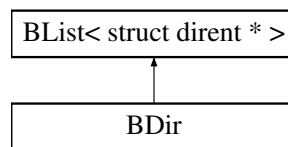
- [/src/cern/tms/beam/libBeam/BDictMap.h](#)

7.19 BDir Class Reference

File system directory class.

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

Inheritance diagram for BDir:



Public Member Functions

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

Private Attributes

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

Additional Inherited Members

7.19.1 Detailed Description

File system directory class.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 `BDir::BDir ()`

7.19.2.2 `BDir::BDir (BString name)`

7.19.2.3 `BDir::~~BDir ()`

7.19.3 Member Function Documentation

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

Clears list.

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

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

Get filename.

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

Get file stats.

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

Get file stats 64.

7.19.3.5 `BError BDir::error ()`

Current value of error.

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

Reads named directory.

7.19.3.7 `BError BDir::read ()`

read/re-reads directory

7.19.3.8 void BDir::setSort (int on)

Set alpha sort on/off.

7.19.3.9 void BDir::setWild (BString wild)

Set wildcard filter string used on read.

7.19.4 Member Data Documentation

7.19.4.1 BString BDir::odirname [private]

7.19.4.2 BError BDir::oerror [private]

7.19.4.3 int BDir::osort [private]

7.19.4.4 BString BDir::owild [private]

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

- [/src/cern/tms/beam/libBeam/BDir.h](#)
- [/src/cern/tms/beam/libBeam/BDir.cpp](#)

7.20 BDuration Class Reference

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

Public Member Functions

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

Private Attributes

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

7.20.1 Constructor & Destructor Documentation

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

7.20.1.2 `BDuration::BDuration (BString str)`

7.20.1.3 `BDuration::~~BDuration ()`

7.20.2 Member Function Documentation

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

Add the given number of micro seconds.

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

Add the given number of milli seconds.

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

Add the given number of seconds.

7.20.2.4 `void BDuration::clear ()`

Clear the duration.

7.20.2.5 `uint64_t BDuration::getMicroSeconds ()`

Get number of micro seconds.

7.20.2.6 `uint32_t BDuration::getSeconds ()`

Get number of seconds.

7.20.2.7 `BString BDuration::getString ()`

Get the time as an ISO date/time string.

7.20.2.8 `int BDuration::hour ()`

7.20.2.9 `int BDuration::microSecond ()`

7.20.2.10 `int BDuration::minute ()`

7.20.2.11 `int BDuration::second ()`

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

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

Set the time from an ISO date/time.

7.20.3 Member Data Documentation

7.20.3.1 `uint8_t BDuration::ohour [private]`

Hour (0 .. 23)

7.20.3.2 `uint32_t BDuration::omicroSecond [private]`

MicroSecond (0 .. 999999)

7.20.3.3 `uint8_t BDuration::ominute [private]`

Minute (0 .. 59)

7.20.3.4 `uint8_t BDuration::osecond [private]`

Second (0 .. 59)

7.20.3.5 `uint8_t BDuration::ospare [private]`

Padding.

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

- [/src/cern/tms/beam/libBeam/BDuration.h](#)
- [/src/cern/tms/beam/libBeam/BDuration.cpp](#)

7.21 BEntry Class Reference

Manipulate a name value pair.

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

Public Member Functions

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

Set name and value.

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

Private Attributes

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

7.21.1 Detailed Description

Manipulate a name value pair.

7.21.2 Constructor & Destructor Documentation

7.21.2.1 BEntry::BEntry ()

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

Set name and value.

7.21.2.3 BEntry::BEntry (BString line)

Set name and value from white space delimited string.

7.21.3 Member Function Documentation

7.21.3.1 BString BEntry::getName ()

Get the name.

7.21.3.2 BString BEntry::getValue ()

Get the value.

7.21.3.3 BString BEntry::line ()

Return name and value as padded single string.

7.21.3.4 void BEntry::print ()

Print name and value.

7.21.3.5 void BEntry::setLine (BString line)

Set name and value from white space delimited string.

7.21.3.6 void BEntry::setName (BString name)

Set the name.

7.21.3.7 void BEntry::setValue (BString value)

Set the value.

7.21.4 Member Data Documentation

7.21.4.1 BString BEntry::oname [private]

7.21.4.2 BString BEntry::ovalue [private]

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

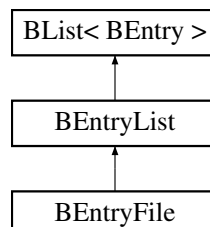
- [/src/cern/tms/beam/libBeam/BEntry.h](#)
- [/src/cern/tms/beam/libBeam/BEntry.cpp](#)

7.22 BEntryFile Class Reference

File of Entries.

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

Inheritance diagram for BEntryFile:



Public Member Functions

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

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

Private Attributes

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

Additional Inherited Members

7.22.1 Detailed Description

File of Entries.

7.22.2 Constructor & Destructor Documentation

7.22.2.1 [BEntryFile::BEntryFile \(\)](#)

7.22.2.2 [BEntryFile::BEntryFile \(BString filename \)](#)

Opens entryfile.

7.22.2.3 [BEntryFile::~~BEntryFile \(\)](#)

7.22.3 Member Function Documentation

7.22.3.1 [void BEntryFile::clear \(\)](#) `[virtual]`

Clears current list.

Reimplemented from [BEntryList](#).

7.22.3.2 [BString BEntryFile::filename \(\)](#)

Returns the filename.

7.22.3.3 [int BEntryFile::open \(BString filename \)](#)

Opens entryfile.

7.22.3.4 int BEntryFile::read ()

Reads entry file and builds list.

7.22.3.5 int BEntryFile::write ()

Writes list to entryfile.

7.22.3.6 int BEntryFile::writeList (BEntryList & l)

Writes specified list to file.

7.22.4 Member Data Documentation

7.22.4.1 BString BEntryFile::ocomments [private]

7.22.4.2 BString BEntryFile::ofilename [private]

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

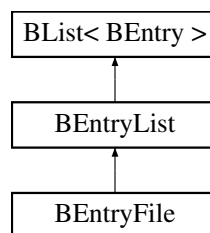
- [/src/cern/tms/beam/libBeam/BEntry.h](#)
- [/src/cern/tms/beam/libBeam/BEntry.cpp](#)

7.23 BEntryList Class Reference

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

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

Inheritance diagram for BEntryList:



Public Member Functions

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

- Raw setting of value without looking up existing entry.*

 - void `deleteEntry` (`BString` name)

Deletes the entry.
- void `print` ()

Print list.
- `BString` `getString` ()

Return list as string. Each Entry padded and on a new line.
- void `insert` (`BIter` &i, const `BEntry` &item)

Insert item before item.
- void `del` (`BIter` &i)

Delete specified item.
- void `clear` ()

Clear the list.
- `BEntryList` & `operator=` (const `BEntryList` &l)

Private Attributes

- `BIter` `olastPos`

Additional Inherited Members

7.23.1 Detailed Description

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

7.23.2 Constructor & Destructor Documentation

7.23.2.1 `BEntryList::BEntryList` ()

7.23.3 Member Function Documentation

7.23.3.1 `void BEntryList::clear` () `[virtual]`

Clear the list.

Reimplemented from `BList< BEntry >`.

Reimplemented in `BEntryFile`.

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

Delete specified item.

Reimplemented from `BList< BEntry >`.

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

Deletes the entry.

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

Returns entry if name is found otherwise NULL.

7.23.3.5 BString BEntryList::findValue (BString name)

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

7.23.3.6 BString BEntryList::getString ()

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

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

Insert item before item.

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

7.23.3.8 int BEntryList::isSet (BString name)

1 if name is in list and value is set

7.23.3.9 BEntryList & BEntryList::operator= (const BEntryList & l)

7.23.3.10 void BEntryList::print ()

Print list.

7.23.3.11 int BEntryList::setValue (BString name, BString value)

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

7.23.3.12 int BEntryList::setValueRaw (BString name, BString value)

Raw setting of value without looking up existing entry.

7.23.4 Member Data Documentation

7.23.4.1 BIter BEntryList::olastPos [private]

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

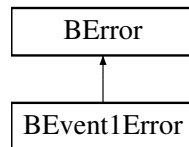
- [/src/cern/tms/beam/libBeam/BEntry.h](#)
- [/src/cern/tms/beam/libBeam/BEntry.cpp](#)

7.24 BError Class Reference

Error return class.

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

Inheritance diagram for BError:



Public Member Functions

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

Private Attributes

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

7.24.1 Detailed Description

Error return class.

7.24.2 Constructor & Destructor Documentation

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

Create object.

7.24.2.2 `BError::BError (BString errStr)`

Create with error set and error string.

7.24.3 Member Function Documentation

7.24.3.1 `BError & BError::clear ()`

Clear the error.

7.24.3.2 `BError BError::copy ()`

Return an independant copy.

7.24.3.3 `int BError::getErrorNo () const`

Get The error number.

7.24.3.4 `int BError::getNumber () const`

Get The error number.

7.24.3.5 `BString BError::getString () const`

Get error message.

7.24.3.6 `int BError::num () const`

Get The error number.

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

Return error number.

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

Set error number and message.

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

Set error type ERROR with optional message.

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

Return a char* string.

7.24.4 Member Data Documentation

7.24.4.1 `int BError::oerrNo` [private]

7.24.4.2 `BString BError::oerrStr` [private]

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

- [/src/cern/tms/beam/libBeam/BError.h](#)
- [/src/cern/tms/beam/libBeam/BError.cpp](#)

7.25 BErrorTime Class Reference

Error return class.

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

Public Types

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

Public Member Functions

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

Private Attributes

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

7.25.1 Detailed Description

Error return class.

7.25.2 Member Enumeration Documentation

7.25.2.1 enum BErrorTime::Type

Enumerator

None

Error

7.25.3 Constructor & Destructor Documentation

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

Create object.

7.25.4 Member Function Documentation

7.25.4.1 BErrorTime & BErrorTime::clear ()

Clear the error.

7.25.4.2 BErrorTime BErrorTime::copy ()

Return an independant copy.

7.25.4.3 int BErrorTime::getErrorNo () const

Get The error number.

7.25.4.4 BString BErrorTime::getString () const

Get error message.

7.25.4.5 BTimeStamp BErrorTime::getTime () const

Get time.

7.25.4.6 BErrorTime::operator int () const

Return error number.

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

Set error number and message.

7.25.5 Member Data Documentation

7.25.5.1 int BErrorTime::oerrNo [private]

7.25.5.2 BString BErrorTime::oerrStr [private]

7.25.5.3 BTimeStamp BErrorTime::oerrTime [private]

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

- [/src/cern/tms/beam/libBeam/BErrorTime.h](#)
- [/src/cern/tms/beam/libBeam/BErrorTime.cpp](#)

7.26 BEvent Class Reference

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

Public Member Functions

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

Private Attributes

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

7.26.1 Constructor & Destructor Documentation

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

7.26.2 Member Function Documentation

7.26.2.1 [BUInt32 BEvent::arg \(\)](#)

7.26.2.2 [BUInt32 BEvent::type \(\)](#)

7.26.3 Member Data Documentation

7.26.3.1 [BUInt32 BEvent::oarg \[private\]](#)

The events argument.

7.26.3.2 [BUInt32 BEvent::otype \[private\]](#)

The events type.

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

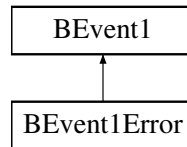
- [/src/cern/tms/beam/libBeam/BEvent.h](#)
- [/src/cern/tms/beam/libBeam/BEvent.cpp](#)

7.27 BEvent1 Class Reference

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

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

Inheritance diagram for BEvent1:



Public Member Functions

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

Private Attributes

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

7.27.1 Detailed Description

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

7.27.2 Constructor & Destructor Documentation

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

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

7.27.3 Member Function Documentation

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

Reimplemented in [BEvent1Error](#).

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

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

Reimplemented in [BEvent1Error](#).

7.27.4 Member Data Documentation

7.27.4.1 uint32_t BEvent1::otype [private]

The event type.

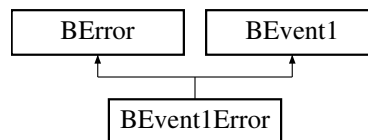
The documentation for this class was generated from the following files:

- [/src/cern/tms/beam/libBeam/BEvent1.h](#)
- [/src/cern/tms/beam/libBeam/BEvent1.cpp](#)

7.28 BEvent1Error Class Reference

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

Inheritance diagram for BEvent1Error:



Public Member Functions

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

7.28.1 Constructor & Destructor Documentation

7.28.1.1 [BEvent1Error::BEvent1Error](#) (int *errNo* = [ErrorOk](#), [BString](#) *errStr* = " ")

7.28.2 Member Function Documentation

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

Reimplemented from [BEvent1](#).

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

Reimplemented from [BEvent1](#).

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

- [/src/cern/tms/beam/libBeam/BEvent1.h](#)
- [/src/cern/tms/beam/libBeam/BEvent1.cpp](#)

7.29 BEvent1Int Class Reference

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

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

Public Member Functions

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

Private Attributes

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

7.29.1 Detailed Description

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

7.29.2 Constructor & Destructor Documentation

7.29.2.1 [BEvent1Int::BEvent1Int](#) ()

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

7.29.3 Member Function Documentation

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

Clear events pending.

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

Receive the event.

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

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

Send an event.

7.29.4 Member Data Documentation

7.29.4.1 int [BEvent1Int::ofds](#)[2] [*private*]

File descriptors for pipe.

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

- [/src/cern/tms/beam/libBeam/BEvent1.h](#)
- [/src/cern/tms/beam/libBeam/BEvent1.cpp](#)

7.30 BEvent1Pipe Class Reference

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

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

Public Member Functions

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

Private Attributes

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

7.30.1 Detailed Description

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

7.30.2 Constructor & Destructor Documentation

7.30.2.1 [BEvent1Pipe::BEvent1Pipe](#) ()

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

7.30.3 Member Function Documentation

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

Clear events pending.

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

Receive the event.

7.30.3.3 int BEvent1Pipe::getReceiveFd ()

returns the receive file descriptor for the poll system call

7.30.3.4 BError BEvent1Pipe::sendEvent (BEvent1 * event)

Send an event.

7.30.4 Member Data Documentation

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

File descriptors for pipe.

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

- [/src/cern/tms/beam/libBeam/BEvent1.h](#)
- [/src/cern/tms/beam/libBeam/BEvent1.cpp](#)

7.31 BEventPipe Class Reference

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

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

Public Member Functions

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

Private Attributes

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

7.31.1 Detailed Description

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

7.31.2 Constructor & Destructor Documentation

7.31.2.1 BEventPipe::BEventPipe ()

7.31.2.2 BEventPipe::~~BEventPipe ()

7.31.3 Member Function Documentation

7.31.3.1 void BEventPipe::clear ()

Clear events pending.

7.31.3.2 int BEventPipe::getFd ()

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

Get an item from the queue.

7.31.3.4 BUInt BEventPipe::readAvailable () const

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

Append an item onto the queue.

7.31.3.6 BUInt BEventPipe::writeAvailable () const

7.31.4 Member Data Documentation

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

File descriptors for pipe.

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

- [/src/cern/tms/beam/libBeam/BEvent.h](#)
- [/src/cern/tms/beam/libBeam/BEvent.cpp](#)

7.32 BFifo< Type > Class Template Reference

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

Public Member Functions

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

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

Protected Attributes

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

7.32.1 Constructor & Destructor Documentation

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

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

7.32.2 Member Function Documentation

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

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

Direct access to read samples in buffer.

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

Read one item.

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

Read a set of items.

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

How many items are available to read.

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

How many items are available to read in a chunk.

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

Returns a pointer to the data.

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

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

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

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

Read item at given offset from current read position.

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

Resize FIFO, clears it as well.

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

Returns fifo size.

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

Write a single item.

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

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

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

How many items that can be written.

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

How many items that can be written in a chunk.

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

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

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

Returns a pointer to the data.

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

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

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

Indicates when write is complete.

7.32.3 Member Data Documentation

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

FIFO memory buffer.

7.32.3.2 `template<class Type> BMutex BFifo< Type >::oLock` [protected]

7.32.3.3 `template<class Type> BUInt BFifo< Type >::oReadPos` [protected]

The read pointer.

7.32.3.4 `template<class Type> BUInt BFifo< Type >::oSize` [protected]

The size of the FIFO.

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

The write pointer.

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

- `/src/cern/tms/beam/libBeam/BFifo.h`

7.33 BFifoCirc< Type > Class Template Reference

This class implements a thread safe FIFO buffer.

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

Public Types

- enum { `defaultSize` = 1024 }

Public Member Functions

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

Protected Member Functions

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

Protected Attributes

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

7.33.1 Detailed Description

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

This class implements a thread safe FIFO buffer.

7.33.2 Member Enumeration Documentation

7.33.2.1 `template<class Type > anonymous enum`

Enumerator

defaultSize

7.33.3 Constructor & Destructor Documentation

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

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

7.33.4 Member Function Documentation

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

Clear all of the data in the buffer.

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

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

Direct access to read samples in buffer.

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

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

Returns the number of bytes of data available.

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

Pointer to raw data.

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

Updates read pointer.

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

Wait for given number of samples.

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

Return the buffers actual size.

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

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

Writes the data to the buffer. Blocks until complete.

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

Returns the space available to write.

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

Return a pointer to the current start of the buffer.

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

Update the write pointer.

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

Wait for the given number of samples.

7.33.5 Member Data Documentation

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

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

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

Current read position.

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

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

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

The number of samples in the FIFO.

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

Current write position.

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

- [/src/cern/tms/beam/libBeam/BFifoCirc.h](#)

7.34 BFifoCircPos Class Reference

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

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

Public Member Functions

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

Private Attributes

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

7.34.1 Detailed Description

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

7.34.2 Constructor & Destructor Documentation

7.34.2.1 `BFifoCircPos::BFifoCircPos (uint32_t size)`

7.34.3 Member Function Documentation

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

Return the difference between the two pointers.

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

Increment the pointer by the given value.

7.34.3.3 `BFifoCircPos::operator int ()`

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

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

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

7.34.3.7 `uint32_t BFifoCircPos::pos ()`

The current position.

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

Sets the position.

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

7.34.4 Member Data Documentation

7.34.4.1 `uint32_t BFifoCircPos::opos` [private]

7.34.4.2 `uint32_t BFifoCircPos::osize` [private]

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

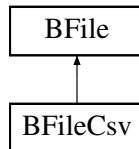
- [/src/cern/tms/beam/libBeam/BFifoCirc.h](#)
- [/src/cern/tms/beam/libBeam/BFifoCirc.cpp](#)

7.35 BFile Class Reference

File operations class.

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

Inheritance diagram for BFile:



Public Member Functions

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

Private Attributes

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

7.35.1 Detailed Description

File operations class.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 BFile::BFile ()

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

Create opened specified file.

7.35.2.3 BFile::~~BFile ()

7.35.3 Member Function Documentation

7.35.3.1 BError BFile::close ()

Close file.

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

7.35.3.3 BString BFile::fileName ()

Return file name.

7.35.3.4 BError BFile::flush ()

Flush the file.

7.35.3.5 FILE * BFile::getFd ()

File descriptor.

7.35.3.6 int BFile::isEnd ()

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

7.35.3.7 int BFile::isOpen ()

Returns 1 if the file is open.

7.35.3.8 BUInt64 BFile::length ()

File size in bytes.

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

Open file.

7.35.3.10 BError BFile::open (FILE * file)

Assign object to opened file handle.

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

Assign object to opened file descriptor.

7.35.3.12 BFile & BFile::operator= (const BFile & file)**7.35.3.13 BUInt64 BFile::position ()**

The files position.

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

Formatted print into the file.

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

Read from file.

7.35.3.16 int BFile::readString (BString & str)

Read string. (ref fgets)

7.35.3.17 int BFile::seek (BUInt64 pos)

Set seek position.

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

Set stream buffering options.

7.35.3.19 BError BFile::truncate ()

Truncate the file.

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

Write to file.

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

Write string to file.

7.35.4 Member Data Documentation

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

7.35.4.2 `BString BFile::ofilename` [private]

7.35.4.3 `BString BFile::omode` [private]

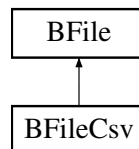
The documentation for this class was generated from the following files:

- [/src/cern/tms/beam/libBeam/BFile.h](#)
- [/src/cern/tms/beam/libBeam/BFile.cpp](#)

7.36 BFileCsv Class Reference

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

Inheritance diagram for BFileCsv:



Public Member Functions

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

Private Attributes

- char [oseparator](#)

7.36.1 Constructor & Destructor Documentation

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

7.36.2 Member Function Documentation

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

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

7.36.3 Member Data Documentation

7.36.3.1 `char BFileCsv::oseparator` [private]

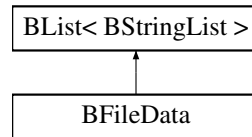
The documentation for this class was generated from the following files:

- [/src/cern/tms/beam/libBeam/BFileCsv.h](#)
- [/src/cern/tms/beam/libBeam/BFileCsv.cpp](#)

7.37 BFileData Class Reference

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

Inheritance diagram for BFileData:



Public Member Functions

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

Private Member Functions

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

Private Attributes

- [BString ofilename](#)

Additional Inherited Members

7.37.1 Member Function Documentation

7.37.1.1 [BError BFileData::del](#) (int *id*)

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

7.37.1.3 [BError BFileData::getNextId](#) (int & *id*)

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

7.37.1.5 [BError BFileData::read](#) () [private]

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

7.37.1.7 [BError BFileData::write](#) () [private]

7.37.2 Member Data Documentation

7.37.2.1 [BString BFileData::ofilename](#) [private]

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

- [/src/cern/tms/beam/libBeam/BFileData.h](#)
- [/src/cern/tms/beam/libBeam/BFileData.cpp](#)

7.38 Blter Class Reference

Iterator for [BList](#).

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

Public Member Functions

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

Private Attributes

- [BNode * oi](#)

7.38.1 Detailed Description

Iterator for [BList](#).

7.38.2 Constructor & Destructor Documentation

7.38.2.1 [Blter::Blter \(BNode * i = 0 \)](#) [inline]

7.38.3 Member Function Documentation

7.38.3.1 [Blter::operator BNode * \(\)](#) [inline]

7.38.3.2 [int Blter::operator== \(const Blter & i \)](#) [inline]

7.38.3.3 [int Blter::valid \(\)](#) [inline]

7.38.4 Member Data Documentation

7.38.4.1 [BNode* Blter::oi](#) [private]

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

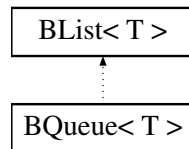
- [/src/cern/tms/beam/libBeam/BList.h](#)

7.39 BList< T > Class Template Reference

Template based list class.

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

Inheritance diagram for BList< T >:



Classes

- class [Node](#)

Public Types

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

Public Member Functions

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

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

Protected Member Functions

- virtual Node * `nodeGet` (BIter i)
- virtual const Node * `nodeGet` (BIter i) const
- virtual Node * `nodeCreate` (const T &item)

Protected Attributes

- Node * `onodes`
- unsigned int `olength`

Private Member Functions

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

7.39.1 Detailed Description

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

Template based list class.

7.39.2 Member Typedef Documentation

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

Prototype for sorting function.

7.39.3 Constructor & Destructor Documentation

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

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

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

7.39.4 Member Function Documentation

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

Append item to list.

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

Append list to list.

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

Iterator for start of list.

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

Clear the list.

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

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

Delete specified item.

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

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

Delete first item.

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

Delete last item.

7.39.4.8 `template<class T> BIter BList< T >::end () const`

Iterator for end of list.

7.39.4.9 `template<class T> BIter BList< T >::end (BIter & i) const`

Iterator for end of list.

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

Get first item in list.

7.39.4.11 `template<class T> T & BList< T >::get (BIter i)`

Get item specified by iterator in list.

7.39.4.12 `template<class T> const T & BList< T >::get (BIter i) const`

Get item specified by iterator in list.

7.39.4.13 `template<class T> BIter BList< T >::goTo (int pos) const`

Iterator for pos item in list.

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

Checks if the item is in the list.

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

Insert item before item.

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

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

Insert item after item.

7.39.4.17 `template<class T> int BList< T >::isEnd (BIter & i) const`

True if iterator refers to last item.

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

Iterator for next item in list.

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

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

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

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

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

Number of items in list.

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

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

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

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

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

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

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

Pop item from list deleting item.

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

Position in list item with iterator i.

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

Iterator for previous item in list.

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

Push item onto list.

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

Add item to end of list.

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

Get item from front of list deleting item.

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

Get last item in list.

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

Number of items in list.

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

Sort list based on get(i) values.

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

Sort list based on Sort func.

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

Iterator to start of list.

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

Swap two items in list.

7.39.5 Member Data Documentation

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

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

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

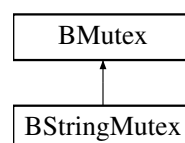
- [/src/cern/tms/beam/libBeam/BList.h](#)
- [/src/cern/tms/beam/libBeam/BList_func.h](#)

7.40 BMutex Class Reference

Mutex class.

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

Inheritance diagram for BMutex:



Public Types

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

Public Member Functions

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

Private Attributes

- `pthread_mutex_t omutex`

7.40.1 Detailed Description

Mutex class.

7.40.2 Member Enumeration Documentation

7.40.2.1 enum `BMutex::Type`

Enumerator

Normal

Recursive

7.40.3 Constructor & Destructor Documentation

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

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

7.40.3.3 `BMutex::~~BMutex` ()

7.40.4 Member Function Documentation

7.40.4.1 `int BMutex::lock` ()

Set lock, wait as necessary.

7.40.4.2 `BMutex & BMutex::operator= (const BMutex & mutex)`

7.40.4.3 `int BMutex::timedLock (int timeoutUs)`

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

7.40.4.4 `int BMutex::tryLock ()`

Test the lock.

7.40.4.5 `int BMutex::unlock ()`

Unlock the lock.

7.40.5 Member Data Documentation

7.40.5.1 `pthread_mutex_t BMutex::omutex [private]`

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

- [/src/cern/tms/beam/libBeam/BMutex.h](#)
- [/src/cern/tms/beam/libBeam/BMutex.cpp](#)

7.41 BMutexLock Class Reference

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

Public Member Functions

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

Private Attributes

- [BMutex & olock](#)

7.41.1 Constructor & Destructor Documentation

7.41.1.1 `BMutexLock::BMutexLock (BMutex & lock, int doLock = 0) [inline]`

7.41.1.2 `BMutexLock::~~BMutexLock () [inline]`

7.41.2 Member Function Documentation

7.41.2.1 `int BMutexLock::lock () [inline]`

7.41.2.2 `int BMutexLock::unlock () [inline]`

7.41.3 Member Data Documentation

7.41.3.1 BMutex& BMutexLock::oLock [private]

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

- /src/cern/tms/beam/libBeam/BMutex.h

7.42 Bmysql Class Reference

```
#include <Bmysql.h>
```

Public Member Functions

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

Private Attributes

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

7.42.1 Constructor & Destructor Documentation

7.42.1.1 Bmysql::Bmysql ()

7.42.1.2 Bmysql::~~Bmysql ()

7.42.2 Member Function Documentation

7.42.2.1 void Bmysql::close ()

7.42.2.2 MYSQL & Bmysql::db ()

7.42.2.3 BError Bmysql::del ([BString](#) table, [BUInt32](#) id)

Delete record from table.

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

Escapes special characters in the string.

7.42.2.5 BError BMySQL::flush ()

Flush all data to disk.

7.42.2.6 BError BMySQL::get (BString *table*, BString *where*, BDictString & *fields*)7.42.2.7 BError BMySQL::insert (BString *table*, BDictString *fields*, BUInt32 * *id* = 0)7.42.2.8 BError BMySQL::open (BString *hostName*, BString *dataBase*, BString *userName*, BString *password*)7.42.2.9 BError BMySQL::query (BString *cmd*, BList< BDictString > & *result*)7.42.2.10 void BMySQL::setDebug (int *debug*)7.42.2.11 BError BMySQL::update (BString *table*, BUInt32 *id*, BDictString *fields*)

7.42.3 Member Data Documentation

7.42.3.1 MYSQL BMySQL::odb [private]

7.42.3.2 int BMySQL::odebug [private]

7.42.3.3 BMutex BMySQL::olock [private]

7.42.3.4 int BMySQL::oopened [private]

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

- [/src/cern/tms/beam/libBeam/BMySQL.h](#)
- [/src/cern/tms/beam/libBeam/BMySQL.cpp](#)

7.43 BNameValue< T > Class Template Reference

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

Public Member Functions

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

Private Attributes

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

7.43.1 Constructor & Destructor Documentation

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

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

7.43.2 Member Function Documentation

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

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

7.43.3 Member Data Documentation

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

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

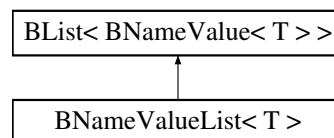
The documentation for this class was generated from the following file:

- [/src/cern/tms/beam/libBeam/BNameValue.h](#)

7.44 BNameValueList< T > Class Template Reference

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

Inheritance diagram for BNameValueList< T >:



Public Member Functions

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

Additional Inherited Members

7.44.1 Member Function Documentation

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

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

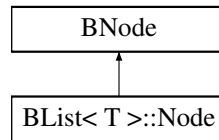
The documentation for this class was generated from the following file:

- [/src/cern/tms/beam/libBeam/BNameValue.h](#)

7.45 BNode Class Reference

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

Inheritance diagram for BNode:



Public Member Functions

- [BNode \(\)](#)

Public Attributes

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

7.45.1 Constructor & Destructor Documentation

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

7.45.2 Member Data Documentation

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

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

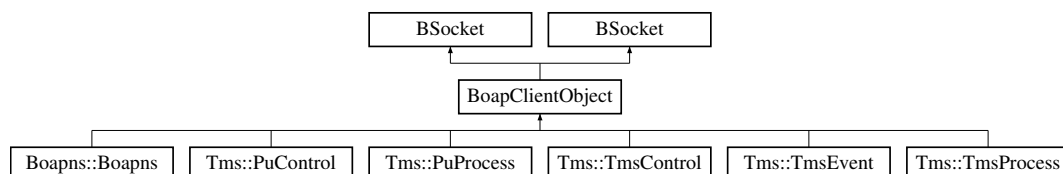
The documentation for this class was generated from the following file:

- [/src/cern/tms/beam/libBeam/BList.h](#)

7.46 BoapClientObject Class Reference

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

Inheritance diagram for BoapClientObject:



Public Member Functions

- [BoapClientObject \(BString name= ""\)](#)
- [virtual ~BoapClientObject \(\)](#)

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

Protected Member Functions

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

Protected Attributes

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

Additional Inherited Members

7.46.1 Constructor & Destructor Documentation

7.46.1.1 `BoapClientObject::BoapClientObject (BString name = " ")`

7.46.1.2 `BoapClientObject::~~BoapClientObject ()` [virtual]

7.46.1.3 `BoapClientObject::BoapClientObject (BString name)`

7.46.2 Member Function Documentation

7.46.2.1 `BError BoapClientObject::checkApiVersion ()` [protected]

7.46.2.2 `BError BoapClientObject::connectService (BString name)`

Connects to the named service.

7.46.2.3 `BError BoapClientObject::connectService (BString name)`

7.46.2.4 `BError BoapClientObject::disconnectService ()`

Disconnects from the named service.

7.46.2.5 `BString BoapClientObject::getServiceName ()`

Get the name of the service.

7.46.2.6 `BError BoapClientObject::handleReconnect (BError err)` [protected],[virtual]

Handle a reconnect performing autorisazion if required.

7.46.2.7 `BError BoapClientObject::performCall (BoapPacket & tx, BoapPacket & rx)` [protected]

Performs a RPC call to the named service.

7.46.2.8 `BError BoapClientObject::performCall (BoapPacket & tx, BoapPacket & rx)` [protected]

7.46.2.9 `BError BoapClientObject::performRecv (BoapPacket & rx)` [protected]

Performs a receive.

7.46.2.10 `BError BoapClientObject::performRecv (BoapPacket & rx)` [protected]

7.46.2.11 `BError BoapClientObject::performSend (BoapPacket & tx)` [protected]

Performs a send to the named service.

7.46.2.12 `BError BoapClientObject::performSend (BoapPacket & tx)` [protected]

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

Pings the connection and finds the remotes version number.

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

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

Sets the connection priority.

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

Sets the maximum packet length.

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

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

7.46.3 Member Data Documentation

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

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

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

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

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

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

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

Handle an automatic reconnect on timeout.

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

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

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

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

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

- [/src/cern/tms/beam/libBeam/Boap.h](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.h](#)
- [/src/cern/tms/beam/libBeam/Boap.cpp](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.cc](#)

7.47 Boapns::BoapEntry Class Reference

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

Public Member Functions

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

Public Attributes

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

7.47.1 Constructor & Destructor Documentation

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

7.47.2 Member Data Documentation

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

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

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

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

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

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

- [/src/cern/tms/beam/libBeam/BoapnsD.h](#)
- [/src/cern/tms/beam/libBeam/BoapnsD.cpp](#)

7.48 BoapFuncEntry Class Reference

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

Public Member Functions

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

Public Attributes

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

7.48.1 Constructor & Destructor Documentation

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

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

7.48.2 Member Data Documentation

7.48.2.1 [UInt32 BoapFuncEntry::ocmd](#)

7.48.2.2 [BUInt32 BoapFuncEntry::ocmd](#)

7.48.2.3 [BoapFunc BoapFuncEntry::ofunc](#)

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

- [/src/cern/tms/beam/libBeam/Boap.h](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.h](#)
- [/src/cern/tms/beam/libBeam/Boap.cpp](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.cc](#)

7.49 BoapMcClientObject Class Reference

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

Public Member Functions

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

Returns the API version.

Protected Member Functions

- [BError performCall \(\)](#)
Performs a RPC call to the named service.
- [BError performSend \(\)](#)
Performs a send to the named service.
- [BError performRecv \(\)](#)
Performs a receive.

Protected Attributes

- [BUInt32 oapiVersion](#)
- [BComms & ocomms](#)
- [BUInt8 oaddressTo](#)
- [BUInt8 oaddressFrom](#)
- [BoapMcPacket opacket](#)

7.49.1 Constructor & Destructor Documentation

7.49.1.1 `BoapMcClientObject::BoapMcClientObject (BComms & comms)`

7.49.1.2 `BoapMcClientObject::~~BoapMcClientObject ()` [virtual]

7.49.2 Member Function Documentation

7.49.2.1 `BUInt32 BoapMcClientObject::getApiVersion ()`

Returns the API version.

7.49.2.2 `BError BoapMcClientObject::performCall ()` [protected]

Performs a RPC call to the named service.

7.49.2.3 `BError BoapMcClientObject::performRecv ()` [protected]

Performs a receive.

7.49.2.4 `BError BoapMcClientObject::performSend ()` [protected]

Performs a send to the named service.

7.49.2.5 `void BoapMcClientObject::setAddress (BUInt8 addressTo, BUInt8 addressFrom)`

7.49.3 Member Data Documentation

7.49.3.1 `BUInt8 BoapMcClientObject::oaddressFrom` [protected]

7.49.3.2 `BUInt8 BoapMcClientObject::oaddressTo` [protected]

7.49.3.3 `BUInt32 BoapMcClientObject::oapiVersion` [protected]

7.49.3.4 `BComms& BoapMcClientObject::ocomms` [protected]

7.49.3.5 `BoapMcPacket BoapMcClientObject::opacket` [protected]

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

- [/src/cern/tms/beam/libBeam/BoapMc.h](#)
- [/src/cern/tms/beam/libBeam/BoapMc.cpp](#)

7.50 BoapMcComms Class Reference

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

Public Member Functions

- [BoapMcComms](#) ([Bool](#) threaded=0, [BUInt](#) rxQueueSize=4)
- virtual [~BoapMcComms](#) ()
- void [setCommsMode](#) ([Bool](#) slave, [BUInt](#) txQueueSize)
 - Sets slave mode.*
- void [setComms](#) ([BComms](#) &comms)
 - Sets the communications interface to use.*
- void [setComms](#) ([BComms](#) *comms)
 - Sets the communications interface to use.*
- void [setAddress](#) ([BUInt8](#) addressTo, [BUInt8](#) addressFrom)
 - Sets the to and from addresses.*
- [BUInt32](#) [getApiVersion](#) ()
 - Returns the API version.*
- [BUInt32](#) [setTimeout](#) ([BUInt32](#) timeoutUs)
 - Sets the call timeout returning the current value.*
- virtual [BError](#) [processRx](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))
 - Process any RX packets queuing them as needed.*
- virtual [BError](#) [processRequests](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))
 - Check and process all requests.*
- virtual [BError](#) [processRequest](#) ([BTimeout](#) timeoutUs=[BTimeoutForever](#))
 - Check and process any request.*
- virtual [BError](#) [processPacket](#) ([BoapMcPacket](#) &rx, [BoapMcPacket](#) &tx)
 - Process a recieved packet.*

Protected Member Functions

- [BError](#) [performCall](#) ()
 - Performs a RPC call to the remote side.*
- [BError](#) [performSend](#) ()
 - Performs a RPC send to the remote side.*
- [BError](#) [packetSend](#) ([BoapMcPacket](#) &packet)
 - Receives a packet.*
- [BError](#) [packetRecv](#) ([BoapMcPacket](#) &packet)
 - Receives a packet.*

Protected Attributes

- [Bool](#) othreaded
- [BMutex](#) olockCall
 - Lock for RPC calls. Only one at a time.*
- [BMutex](#) olockTx
 - Lock for TX.*
- [BComms](#) * ocomms
- [BUInt32](#) oapiVersion
- [Bool](#) oslave

- Set slave mode.*
- [BUInt32 otimeout](#)
 - The timeout in us for calls.*
- [BUInt8 oaddressTo](#)
- [BUInt8 oaddressFrom](#)
- [BoapMcPacket opacket](#)
 - Packet RX buffer.*
- [BoapMcPacket opacketTx](#)
 - Packet TX buffer for calls.*
- [BoapMcPacket opacketRx](#)
 - Packet RX buffer for calls.*
- [BSemaphore opacketRxSema](#)
 - Wait RX semaphore.*
- [BoapMcPacket opacketReqTx](#)
 - Packet TX buffer for requests.*
- [BoapMcPacket opacketReqRx](#)
 - Packet RX buffer for requests.*
- [BQueue< BoapMcPacket > opacketReqQueue](#)
 - Packet RX buffer queue for requests.*
- [BFifo< BoapMcPacket > opacketTxQueue](#)
 - Packet TX Queue.*
- [BSemaphoreCount opacketTxQueueWriteNum](#)
 - Packet TX Queue number.*
- [BSemaphore opacketTxSema](#)
 - Wait for TX semaphore.*

7.50.1 Constructor & Destructor Documentation

7.50.1.1 [BoapMcComms::BoapMcComms \(Bool threaded = 0, BUInt rxQueueSize = 4 \)](#)

7.50.1.2 [BoapMcComms::~~BoapMcComms \(\)](#) [virtual]

7.50.2 Member Function Documentation

7.50.2.1 [BUInt32 BoapMcComms::getApiVersion \(\)](#)

Returns the API version.

7.50.2.2 [BError BoapMcComms::packetRecv \(BoapMcPacket & packet \)](#) [protected]

Receives a packet.

7.50.2.3 [BError BoapMcComms::packetSend \(BoapMcPacket & packet \)](#) [protected]

Receives a packet.

7.50.2.4 [BError BoapMcComms::performCall \(\)](#) [protected]

Performs a RPC call to the remote side.

7.50.2.5 BError BoapMcComms::performSend () [protected]

Performs a RPC send to the remote side.

7.50.2.6 BError BoapMcComms::processPacket (BoapMcPacket & rx, BoapMcPacket & tx) [virtual]

Process a recieved packet.

7.50.2.7 BError BoapMcComms::processRequest (BTimeout timeoutUs = BTimeoutForever) [virtual]

Check and process any request.

7.50.2.8 BError BoapMcComms::processRequests (BTimeout timeoutUs = BTimeoutForever) [virtual]

Check and process all requests.

7.50.2.9 BError BoapMcComms::processRx (BTimeout timeoutUs = BTimeoutForever) [virtual]

Process any RX packets queuing them as needed.

!!! This should wait on comms for timeoutUs !!!

7.50.2.10 void BoapMcComms::setAddress (BUInt8 addressTo, BUInt8 addressFrom)

Sets the to and from addresses.

7.50.2.11 void BoapMcComms::setComms (BComms & comms)

Sets the communications interface to use.

7.50.2.12 void BoapMcComms::setComms (BComms * comms)

Sets the communications interface to use.

7.50.2.13 void BoapMcComms::setCommsMode (Bool slave, BUInt txQueueSize)

Sets slave mode.

7.50.2.14 BUInt32 BoapMcComms::setTimeout (BUInt32 timeoutUs)

Sets the call timeout returning the current value.

7.50.3 Member Data Documentation**7.50.3.1 BUInt8 BoapMcComms::oaddressFrom** [protected]**7.50.3.2 BUInt8 BoapMcComms::oaddressTo** [protected]**7.50.3.3 BUInt32 BoapMcComms::oapiVersion** [protected]

7.50.3.4 **BComms*** BoapMcComms::ocomms [protected]

7.50.3.5 **BMutex** BoapMcComms::olockCall [protected]

Lock for RPC calls. Only one at a time.

7.50.3.6 **BMutex** BoapMcComms::olockTx [protected]

Lock for TX.

7.50.3.7 **BoapMcPacket** BoapMcComms::opacket [protected]

Packet RX buffer.

7.50.3.8 **BQueue<BoapMcPacket>** BoapMcComms::opacketReqQueue [protected]

Packet RX buffer queue for requests.

7.50.3.9 **BoapMcPacket** BoapMcComms::opacketReqRx [protected]

Packet RX buffer for requests.

7.50.3.10 **BoapMcPacket** BoapMcComms::opacketReqTx [protected]

Packet TX buffer for requests.

7.50.3.11 **BoapMcPacket** BoapMcComms::opacketRx [protected]

Packet RX buffer for calls.

7.50.3.12 **BSemaphore** BoapMcComms::opacketRxSema [protected]

Wait RX semaphore.

7.50.3.13 **BoapMcPacket** BoapMcComms::opacketTx [protected]

Packet TX buffer for calls.

7.50.3.14 **BFifo<BoapMcPacket>** BoapMcComms::opacketTxQueue [protected]

Packet TX Queue.

7.50.3.15 **BSemaphoreCount** BoapMcComms::opacketTxQueueWriteNum [protected]

Packet TX Queue number.

7.50.3.16 **BSemaphore** BoapMcComms::opacketTxSema [protected]

Wait for TX semaphore.

7.50.3.17 **Bool** BoapMcComms::oslave [protected]

Set slave mode.

7.50.3.18 **Bool** BoapMcComms::othreaded [protected]

7.50.3.19 **BUInt32** BoapMcComms::otimeout [protected]

The timeout in us for calls.

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

- /src/cern/tms/beam/libBeam/BoapMc.h
- /src/cern/tms/beam/libBeam/BoapMc.cpp

7.51 BoapMcPacket Class Reference

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

Public Attributes

- [BoapMcPacketHead](#) head
- char [data](#) [256-sizeof([BoapMcPacketHead](#))]

7.51.1 Member Data Documentation

7.51.1.1 char [BoapMcPacket::data](#)[256-sizeof([BoapMcPacketHead](#))]

7.51.1.2 [BoapMcPacketHead](#) [BoapMcPacket::head](#)

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

- /src/cern/tms/beam/libBeam/BoapMc.h

7.52 BoapMcPacketHead Struct Reference

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

Public Attributes

- [BUInt8](#) length
- [BUInt8](#) addressTo
- [BUInt8](#) addressFrom
- [BUInt8](#) cmd
- [BUInt16](#) error
- [BUInt16](#) checksum

7.52.1 Member Data Documentation

7.52.1.1 **BUInt8** BoapMcPacketHead::addressFrom

7.52.1.2 **BUInt8** BoapMcPacketHead::addressTo

7.52.1.3 **BUInt16** BoapMcPacketHead::checksum

7.52.1.4 **BUInt8** BoapMcPacketHead::cmd

7.52.1.5 **BUInt16** BoapMcPacketHead::error

7.52.1.6 **BUInt8** BoapMcPacketHead::length

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

- [/src/cern/tms/beam/libBeam/BoapMc.h](#)

7.53 BoapMcServiceObject Class Reference

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

Public Member Functions

- [BoapMcServiceObject](#) ()
- virtual [~BoapMcServiceObject](#) ()
- virtual [BError process](#) ([BoapMcPacket](#) &rx, [BoapMcPacket](#) &tx)
- virtual [BError processEvent](#) ([BoapMcPacket](#) &rx)

Protected Member Functions

- [BError sendEvent](#) ([BoapMcPacket](#) &tx)

Protected Attributes

- [BUInt32 oapiVersion](#)

7.53.1 Constructor & Destructor Documentation

7.53.1.1 [BoapMcServiceObject::BoapMcServiceObject](#) ()

7.53.1.2 [BoapMcServiceObject::~~BoapMcServiceObject](#) () [virtual]

7.53.2 Member Function Documentation

7.53.2.1 [BError BoapMcServiceObject::process](#) ([BoapMcPacket](#) & rx, [BoapMcPacket](#) & tx) [virtual]

7.53.2.2 [BError BoapMcServiceObject::processEvent](#) ([BoapMcPacket](#) & rx) [virtual]

7.53.2.3 [BError BoapMcServiceObject::sendEvent](#) ([BoapMcPacket](#) & tx) [protected]

7.53.3 Member Data Documentation

7.53.3.1 BUInt32 BoapMcServiceObject::oapiVersion [protected]

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

- [/src/cern/tms/beam/libBeam/BoapMc.h](#)
- [/src/cern/tms/beam/libBeam/BoapMc.cpp](#)

7.54 BoapMcSignalObject Class Reference

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

Public Member Functions

- [BoapMcSignalObject](#) ([BComms](#) &comms)

Protected Member Functions

- [BError performSend](#) ([BoapMcPacket](#) &tx)

Protected Attributes

- [BComms](#) & [ocomms](#)

7.54.1 Constructor & Destructor Documentation

7.54.1.1 BoapMcSignalObject::BoapMcSignalObject ([BComms](#) & *comms*)

7.54.2 Member Function Documentation

7.54.2.1 BError BoapMcSignalObject::performSend ([BoapMcPacket](#) & *tx*) [protected]

7.54.3 Member Data Documentation

7.54.3.1 [BComms](#)& BoapMcSignalObject::ocomms [protected]

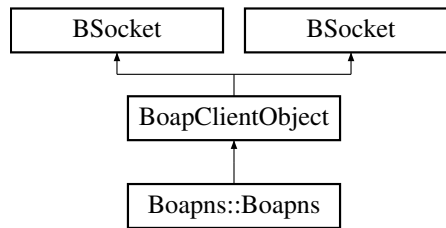
The documentation for this class was generated from the following files:

- [/src/cern/tms/beam/libBeam/BoapMc.h](#)
- [/src/cern/tms/beam/libBeam/BoapMc.cpp](#)

7.55 Boapns::Boapns Class Reference

```
#include <BoapnsC.h>
```

Inheritance diagram for Boapns::Boapns:



Public Member Functions

- [Boapns](#) ([BString](#) name="")
- [BError getVersion](#) ([BString](#) &version)
- [BError getEntryList](#) ([BList](#)< [BoapEntry](#) > &entryList)
- [BError getEntry](#) ([BString](#) name, [BoapEntry](#) &entry)
- [BError addEntry](#) ([BoapEntry](#) entry)
- [BError delEntry](#) ([BString](#) name)
- [BError getNewName](#) ([BString](#) &name)

Additional Inherited Members

7.55.1 Constructor & Destructor Documentation

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

7.55.2 Member Function Documentation

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

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

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

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

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

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

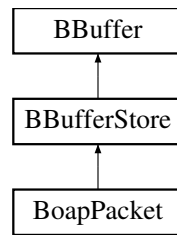
The documentation for this class was generated from the following files:

- [/src/cern/tms/beam/libBeam/BoapnsC.h](#)
- [/src/cern/tms/beam/libBeam/BoapnsC.cpp](#)

7.56 BoapPacket Class Reference

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

Inheritance diagram for BoapPacket:



Public Member Functions

- [BoapPacket](#) ()
- [~BoapPacket](#) ()
- [BUInt32](#) [getCmd](#) ()
- [int](#) [peekHead](#) ([BoapPacketHead](#) &head)
- [int](#) [pushHead](#) ([BoapPacketHead](#) &head)
- [int](#) [popHead](#) ([BoapPacketHead](#) &head)
- [void](#) [updateHead](#) ()
- [BoapPacket](#) ()
- [~BoapPacket](#) ()
- [int](#) [resize](#) ([int](#) size)
- [BError](#) [setData](#) ([void](#) *data, [int](#) nbytes)
- [int](#) [nbytes](#) ()
- [char](#) * [data](#) ()
- [int](#) [pushHead](#) ([BoapPacketHead](#) &head)
- [int](#) [push](#) ([Int8](#) v)
- [int](#) [push](#) ([UInt8](#) v)
- [int](#) [push](#) ([Int16](#) v)
- [int](#) [push](#) ([UInt16](#) v)
- [int](#) [push](#) ([Int32](#) v)
- [int](#) [push](#) ([UInt32](#) v)
- [int](#) [push](#) ([BString](#) &v)
- [int](#) [push](#) ([Double](#) v)
- [int](#) [push](#) ([BError](#) &v)
- [int](#) [push](#) ([UInt32](#) nBytes, [const](#) [void](#) *data)
- [int](#) [popHead](#) ([BoapPacketHead](#) &head)
- [int](#) [pop](#) ([Int8](#) &v)
- [int](#) [pop](#) ([UInt8](#) &v)
- [int](#) [pop](#) ([Int16](#) &v)
- [int](#) [pop](#) ([UInt16](#) &v)
- [int](#) [pop](#) ([Int32](#) &v)
- [int](#) [pop](#) ([UInt32](#) &v)
- [int](#) [pop](#) ([BString](#) &v)
- [int](#) [pop](#) ([Double](#) &v)
- [int](#) [pop](#) ([BError](#) &v)
- [int](#) [pop](#) ([UInt32](#) nBytes, [void](#) *data)

Private Member Functions

- [void](#) [updateLen](#) ()

Private Attributes

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

Additional Inherited Members

7.56.1 Constructor & Destructor Documentation

7.56.1.1 `BoapPacket::BoapPacket ()`

7.56.1.2 `BoapPacket::~~BoapPacket ()`

7.56.1.3 `BoapPacket::BoapPacket ()`

7.56.1.4 `BoapPacket::~~BoapPacket ()`

7.56.2 Member Function Documentation

7.56.2.1 `char * BoapPacket::data ()`

7.56.2.2 `BUInt32 BoapPacket::getCmd ()`

7.56.2.3 `int BoapPacket::nbytes ()`

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

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

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

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

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

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

7.56.2.10 `int BoapPacket::pop (UInt32 & v)`

7.56.2.11 `int BoapPacket::pop (BString & v)`

7.56.2.12 `int BoapPacket::pop (Double & v)`

7.56.2.13 `int BoapPacket::pop (BError & v)`

7.56.2.14 `int BoapPacket::pop (UInt32 nBytes, void * data)`

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

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

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

- 7.56.2.18 `int BoapPacket::push (UInt8 v)`
- 7.56.2.19 `int BoapPacket::push (Int16 v)`
- 7.56.2.20 `int BoapPacket::push (UInt16 v)`
- 7.56.2.21 `int BoapPacket::push (Int32 v)`
- 7.56.2.22 `int BoapPacket::push (UInt32 v)`
- 7.56.2.23 `int BoapPacket::push (BString & v)`
- 7.56.2.24 `int BoapPacket::push (Double v)`
- 7.56.2.25 `int BoapPacket::push (BError & v)`
- 7.56.2.26 `int BoapPacket::push (UInt32 nBytes, const void * data)`
- 7.56.2.27 `int BoapPacket::pushHead (BoapPacketHead & head)`
- 7.56.2.28 `int BoapPacket::pushHead (BoapPacketHead & head)`
- 7.56.2.29 `int BoapPacket::resize (int size)`
- 7.56.2.30 `BError BoapPacket::setData (void * data, int nbytes)`
- 7.56.2.31 `void BoapPacket::updateHead ()`
- 7.56.2.32 `void BoapPacket::updateLen () [private]`

7.56.3 Member Data Documentation

- 7.56.3.1 `char* BoapPacket::odata [private]`
- 7.56.3.2 `int BoapPacket::onbytes [private]`
- 7.56.3.3 `int BoapPacket::opos [private]`
- 7.56.3.4 `int BoapPacket::osize [private]`

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

- [/src/cern/tms/beam/libBeam/Boap.h](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.h](#)
- [/src/cern/tms/beam/libBeam/Boap.cpp](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.cc](#)

7.57 BoapPacketHead Struct Reference

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

Public Attributes

- [BUInt32 type](#)

- [BUInt32 length](#)
- [BUInt32 service](#)
- [BUInt32 cmd](#)
- [UInt32 length](#)
- [BoapType type](#)
- [BoapService service](#)
- [UInt32 cmd](#)
- [UInt32 reserved \[12\]](#)

7.57.1 Member Data Documentation

7.57.1.1 **UInt32** BoapPacketHead::cmd

7.57.1.2 **BUInt32** BoapPacketHead::cmd

7.57.1.3 **UInt32** BoapPacketHead::length

7.57.1.4 **BUInt32** BoapPacketHead::length

7.57.1.5 **UInt32** BoapPacketHead::reserved[12]

7.57.1.6 **BoapService** BoapPacketHead::service

7.57.1.7 **BUInt32** BoapPacketHead::service

7.57.1.8 **BUInt32** BoapPacketHead::type

7.57.1.9 **BoapType** BoapPacketHead::type

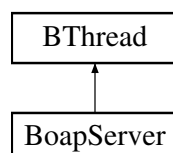
The documentation for this struct was generated from the following files:

- [/src/cern/tms/beam/libBeam/Boap.h](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.h](#)

7.58 BoapServer Class Reference

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

Inheritance diagram for BoapServer:



Public Types

- enum { [NOTHEADS](#) =0, [THREADED](#) =1 }

Public Member Functions

- [BoapServer](#) ()
- virtual [~BoapServer](#) ()
- virtual [BError](#) [init](#) ([BString](#) boapNsHost="", int port=0, int threaded=0, int isBoapns=0)
- virtual [BError](#) [run](#) (int inThread=0)
- virtual [BError](#) [process](#) ([BoapServerConnection](#) *conn, [BoapPacket](#) &rx, [BoapPacket](#) &tx)
- virtual [BError](#) [processEvent](#) ([BoapPacket](#) &rx)
- virtual [BError](#) [addObject](#) ([BoapServiceObject](#) *object)
- virtual [BError](#) [sendEvent](#) ([BoapPacket](#) &tx)
- virtual [BError](#) [processEvent](#) (int fd)
- virtual void [clientGone](#) ([BoapServerConnection](#) *client)
- [BSocket](#) & [getSocket](#) ()
- [BSocket](#) & [getEventSocket](#) ()
- [BString](#) [getHostName](#) ()
- int [getConnectionsNumber](#) ()
- virtual [BoapServerConnection](#) * [newConnection](#) (int fd, [BSocketAddressINET](#) address)
- [BoapServer](#) ()
- [BError](#) [init](#) (int boapNs=0)
- [BError](#) [run](#) ()
- [BError](#) [processEvent](#) ([BoapPacket](#) &rx)
- [BError](#) [addObject](#) ([BoapServiceObject](#) *object)
- [BError](#) [process](#) (int fd)
- [BError](#) [sendEvent](#) ([BoapPacket](#) &tx)
- [BSocket](#) & [getSocket](#) ()
- [BSocket](#) & [getEventSocket](#) ()
- [BError](#) [processEvent](#) (int fd)
- [BString](#) [getHostName](#) ()

Public Attributes

- [BUInt64](#) onumOperations

Private Member Functions

- void * [function](#) ()

Private Attributes

- int othreaded
- int oisBoapns
- [Boapns::Boapns](#) * oboapns
- [BList](#)< [BoapServerConnection](#) * > oclients
- [BEvent1Int](#) oclientGoneEvent
- [BList](#)< [BoapServiceEntry](#) > oservices
- [BPoll](#) opoll
- [BSocket](#) onet
- [BSocket](#) onetEvent
- [BSocketAddressINET](#) onetEventAddress
- [BString](#) ohostName
- int oboapNs
- [BoapPacket](#) orx
- [BoapPacket](#) otx

7.58.1 Member Enumeration Documentation

7.58.1.1 anonymous enum

Enumerator

NOTHEADS

THREADED

7.58.2 Constructor & Destructor Documentation

7.58.2.1 BoapServer::BoapServer ()

7.58.2.2 BoapServer::~BoapServer () [virtual]

7.58.2.3 BoapServer::BoapServer ()

7.58.3 Member Function Documentation

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

7.58.3.2 BError BoapServer::addObject (BoapServiceObject * *object*) [virtual]

7.58.3.3 void BoapServer::clientGone (BoapServerConnection * *client*) [virtual]

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

Reimplemented from [BThread](#).

7.58.3.5 int BoapServer::getConnectionsNumber ()

7.58.3.6 BSocket& BoapServer::getEventSocket ()

7.58.3.7 BSocket & BoapServer::getEventSocket ()

7.58.3.8 BString BoapServer::getHostName ()

7.58.3.9 BString BoapServer::getHostName ()

7.58.3.10 BSocket& BoapServer::getSocket ()

7.58.3.11 BSocket & BoapServer::getSocket ()

7.58.3.12 BError BoapServer::init (int *boapNs* = 0)

7.58.3.13 BError BoapServer::init (BString *boapNsHost* = " ", int *port* = 0, int *threaded* = 0, int *isBoapns* = 0) [virtual]

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

7.58.3.15 BError BoapServer::process (int *fd*)

7.58.3.16 BError BoapServer::process (BoapServerConnection * *conn*, BoapPacket & *rx*, BoapPacket & *tx*) [virtual]

- 7.58.3.17 **BError** BoapServer::processEvent (BoapPacket & rx)
- 7.58.3.18 **BError** BoapServer::processEvent (int fd)
- 7.58.3.19 **BError** BoapServer::processEvent (BoapPacket & rx) [virtual]
- 7.58.3.20 **BError** BoapServer::processEvent (int fd) [virtual]
- 7.58.3.21 **BError** BoapServer::run ()
- 7.58.3.22 **BError** BoapServer::run (int inThread = 0) [virtual]
- 7.58.3.23 **BError** BoapServer::sendEvent (BoapPacket & tx)
- 7.58.3.24 **BError** BoapServer::sendEvent (BoapPacket & tx) [virtual]

7.58.4 Member Data Documentation

- 7.58.4.1 int BoapServer::oBoapNs [private]
- 7.58.4.2 Boapns::Boapns* BoapServer::oBoapns [private]
- 7.58.4.3 BEvent1Int BoapServer::oClientGoneEvent [private]
- 7.58.4.4 BList<BoapServerConnection*> BoapServer::oClients [private]
- 7.58.4.5 BString BoapServer::oHostName [private]
- 7.58.4.6 int BoapServer::oisBoapns [private]
- 7.58.4.7 BSocket BoapServer::oNet [private]
- 7.58.4.8 BSocket BoapServer::oNetEvent [private]
- 7.58.4.9 BSocketAddressINET BoapServer::oNetEventAddress [private]
- 7.58.4.10 BUInt64 BoapServer::oNumOperations
- 7.58.4.11 BPoll BoapServer::oPoll [private]
- 7.58.4.12 BoapPacket BoapServer::oRx [private]
- 7.58.4.13 BList< BoapServiceEntry > BoapServer::oServices [private]
- 7.58.4.14 int BoapServer::oThreaded [private]
- 7.58.4.15 BoapPacket BoapServer::oTx [private]

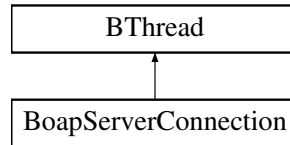
The documentation for this class was generated from the following files:

- /src/cern/tms/beam/libBeam/Boap.h
- /src/cern/tms/beam/libBeam/BoapSimple.h
- /src/cern/tms/beam/libBeam/Boap.cpp
- /src/cern/tms/beam/libBeam/BoapSimple.cc

7.59 BoapServerConnection Class Reference

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

Inheritance diagram for BoapServerConnection:



Public Member Functions

- [BoapServerConnection](#) ([BoapServer](#) &boapServer, int fd)
- virtual [~BoapServerConnection](#) ()
- virtual [BError](#) [init](#) ()
 - Initialise connection.*
- virtual [BError](#) [process](#) ()
- virtual [BSocket](#) & [getSocket](#) ()
- virtual void [setMaxLength](#) ([BUInt32](#) maxLength)
- virtual [BError](#) [getHead](#) ([BoapPacketHead](#) &head)
- virtual [BError](#) [validate](#) ()
 - Validate the connection.*

Private Member Functions

- void * [function](#) ()

Private Attributes

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

7.59.1 Constructor & Destructor Documentation

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

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

7.59.2 Member Function Documentation

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

Reimplemented from [BThread](#).

7.59.2.2 **BError** BoapServerConnection::getHead (**BoapPacketHead & head**) [virtual]

7.59.2.3 **BSocket &** BoapServerConnection::getSocket () [virtual]

7.59.2.4 **BError** BoapServerConnection::init () [virtual]

Initialise connection.

7.59.2.5 **BError** BoapServerConnection::process () [virtual]

7.59.2.6 void BoapServerConnection::setMaxLength (**BUInt32 maxLength**) [virtual]

7.59.2.7 **BError** BoapServerConnection::validate () [virtual]

Validate the connection.

7.59.3 Member Data Documentation

7.59.3.1 **BoapServer&** BoapServerConnection::oboapServer [private]

7.59.3.2 **BUInt32** BoapServerConnection::omaxLength [private]

7.59.3.3 **BoapPacket** BoapServerConnection::orx [private]

7.59.3.4 **BSocket** BoapServerConnection::osocket [private]

7.59.3.5 **BoapPacket** BoapServerConnection::otx [private]

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

- /src/cern/tms/beam/libBeam/Boap.h
- /src/cern/tms/beam/libBeam/Boap.cpp

7.60 BoapServiceEntry Class Reference

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

Public Member Functions

- [BoapServiceEntry](#) ([BoapService](#) service=0, [BoapServiceObject](#) *object=0)
- [BoapServiceEntry](#) ([BoapService](#) service=0, [BoapServiceObject](#) *object=0)

Public Attributes

- [BoapService](#) oservice
- [BoapServiceObject](#) * oobject

7.60.1 Constructor & Destructor Documentation

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

7.60.1.2 `BoapServiceEntry::BoapServiceEntry (BoapService service = 0, BoapServiceObject * object = 0)`
`[inline]`

7.60.2 Member Data Documentation

7.60.2.1 `BoapServiceObject * BoapServiceEntry::oobject`

7.60.2.2 `BoapService BoapServiceEntry::oservice`

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

- [/src/cern/tms/beam/libBeam/Boap.h](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.h](#)

7.61 BoapServiceObject Class Reference

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

Public Member Functions

- [BoapServiceObject \(BoapServer &server, BString name=""\)](#)
- [virtual ~BoapServiceObject \(\)](#)
- [BError setName \(BString name\)](#)
- [BError sendEvent \(BString signalName, BInt32 arg\)](#)
- [virtual BError processEvent \(BString objectName, BString name, BInt32 arg\)](#)
- [BString name \(\)](#)
- [BError doPing \(BoapServerConnection *conn, BoapPacket &rx, BoapPacket &tx\)](#)
- [BError doConnectionPriority \(BoapServerConnection *conn, BoapPacket &rx, BoapPacket &tx\)](#)
- [BError process \(BoapServerConnection *conn, BoapPacket &rx, BoapPacket &tx\)](#)
- [virtual BError processEvent \(BoapPacket &rx\)](#)
- [BoapServiceObject \(BoapServer &server, BString name\)](#)
- [virtual ~BoapServiceObject \(\)](#)
- [BError sendEvent \(BString signalName, Int32 arg\)](#)
- [virtual BError processEvent \(BString objectName, BString name, Int32 arg\)](#)
- [BString name \(\)](#)
- [BError process \(BoapPacket &rx, BoapPacket &tx\)](#)
- [virtual BError processEvent \(BoapPacket &rx\)](#)

Protected Member Functions

- [BError sendEvent \(BoapPacket &tx\)](#)
- [BError sendEvent \(BoapPacket &tx\)](#)

Protected Attributes

- [BoapServer & oserver](#)
- [BString oname](#)
- [BUInt32 oapiVersion](#)
- [BList< BoapFuncEntry > ofuncList](#)

7.61.1 Constructor & Destructor Documentation

7.61.1.1 `BoapServiceObject::BoapServiceObject (BoapServer & server, BString name = " ")`

7.61.1.2 `BoapServiceObject::~~BoapServiceObject ()` [virtual]

7.61.1.3 `BoapServiceObject::BoapServiceObject (BoapServer & server, BString name)`

7.61.1.4 `virtual BoapServiceObject::~~BoapServiceObject ()` [virtual]

7.61.2 Member Function Documentation

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

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

7.61.2.3 `BString BoapServiceObject::name ()`

7.61.2.4 `BString BoapServiceObject::name ()`

7.61.2.5 `BError BoapServiceObject::process (BoapPacket & rx, BoapPacket & tx)`

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

7.61.2.7 `virtual BError BoapServiceObject::processEvent (BString objectName, BString name, Int32 arg)`
[virtual]

7.61.2.8 `virtual BError BoapServiceObject::processEvent (BoapPacket & rx)` [virtual]

7.61.2.9 `BError BoapServiceObject::processEvent (BString objectName, BString name, BInt32 arg)` [virtual]

7.61.2.10 `BError BoapServiceObject::processEvent (BoapPacket & rx)` [virtual]

7.61.2.11 `BError BoapServiceObject::sendEvent (BString signalName, Int32 arg)`

7.61.2.12 `BError BoapServiceObject::sendEvent (BoapPacket & tx)` [protected]

7.61.2.13 `BError BoapServiceObject::sendEvent (BString signalName, BInt32 arg)`

7.61.2.14 `BError BoapServiceObject::sendEvent (BoapPacket & tx)` [protected]

7.61.2.15 `BError BoapServiceObject::setName (BString name)`

7.61.3 Member Data Documentation

7.61.3.1 `BUInt32 BoapServiceObject::oapiVersion` [protected]

7.61.3.2 `BList< BoapFuncEntry > BoapServiceObject::ofuncList` [protected]

7.61.3.3 `BString BoapServiceObject::oname` [protected]

7.61.3.4 `BoapServer & BoapServiceObject::oserver` [protected]

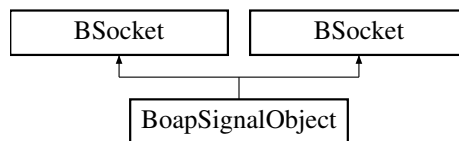
The documentation for this class was generated from the following files:

- [/src/cern/tms/beam/libBeam/Boap.h](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.h](#)
- [/src/cern/tms/beam/libBeam/Boap.cpp](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.cc](#)

7.62 BoapSignalObject Class Reference

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

Inheritance diagram for BoapSignalObject:



Public Member Functions

- [BoapSignalObject \(\)](#)
- [BoapSignalObject \(\)](#)

Protected Member Functions

- [BError performSend \(BoapPacket &tx\)](#)
- [BError performSend \(BoapPacket &tx\)](#)

Protected Attributes

- [BoapPacket otx](#)
- [BoapPacket orx](#)

Additional Inherited Members

7.62.1 Constructor & Destructor Documentation

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

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

7.62.2 Member Function Documentation

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

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

7.62.3 Member Data Documentation

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

7.62.3.2 BoapPacket BoapSignalObject::otx [protected]

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

- [/src/cern/tms/beam/libBeam/Boap.h](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.h](#)
- [/src/cern/tms/beam/libBeam/Boap.cpp](#)
- [/src/cern/tms/beam/libBeam/BoapSimple.cc](#)

7.63 BObj Class Reference

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

Public Member Functions

- [BObj \(\)](#)
- virtual [~BObj \(\)](#)
- virtual const char * [getType \(\)](#) const
- virtual const [BObjMember](#) * [getMembers \(\)](#) const
- virtual [BError](#) [getMembers \(BDictString &members\)](#)
- virtual [BError](#) [getMember \(BString name, BString &value\)](#)
- virtual [BError](#) [setMembers \(BDictString &members\)](#)
- virtual [BError](#) [setMember \(BString name, BString value\)](#)
- virtual void [membersPrint \(\)](#) const
Prints out members.
- virtual [BString](#) [getDebugString \(\)](#)
Returns contents as a debug string.

7.63.1 Constructor & Destructor Documentation

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

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

7.63.2 Member Function Documentation

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

Returns contents as a debug string.

7.63.2.2 [BError BObj::getMember \(BString name, BString & value \)](#) [virtual]

7.63.2.3 [const BObjMember * BObj::getMembers \(\)](#) const [virtual]

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

7.63.2.5 [const char * BObj::getType \(\)](#) const [virtual]

7.63.2.6 [void BObj::membersPrint \(\)](#) const [virtual]

Prints out members.

7.63.2.7 **BError** BObj::setMember (BString *name*, BString *value*) [virtual]

7.63.2.8 **BError** BObj::setMembers (BDictString & *members*) [virtual]

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

- /src/cern/tms/beam/libBeam/BObj.h
- /src/cern/tms/beam/libBeam/BObj.cpp

7.64 BObjMember Struct Reference

```
#include <BTypes.h>
```

Public Attributes

- BType [type](#)
- BTypeComp [typeComp](#)
- BUInt16 [dataOffset](#)
- BUInt16 [size](#)
- const char * [typeName](#)
- const char * [name](#)

7.64.1 Member Data Documentation

7.64.1.1 BUInt16 BObjMember::dataOffset

7.64.1.2 const char* BObjMember::name

7.64.1.3 BUInt16 BObjMember::size

7.64.1.4 BType BObjMember::type

7.64.1.5 BTypeComp BObjMember::typeComp

7.64.1.6 const char* BObjMember::typeName

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

- /src/cern/tms/beam/libBeam/BTypes.h

7.65 BPoll Class Reference

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

```
#include <BPoll.h>
```

Public Types

- typedef struct pollfd [PollFd](#)

Public Member Functions

- [BPoll](#) ()
- [~BPoll](#) ()
- void [append](#) (int fd, int events=POLLIN|POLLERR|POLLHUP|POLLNVAL)
Append a file descriptor to polling list.
- void [delFd](#) (int fd)
Remove a file descriptor from polling list.
- [BError doPoll](#) (int &fd, int timeoutUs=-1)
Perform polling operation.
- [BError doPollEvents](#) (int &fd, int &events, int timeoutUs=-1)
Perform polling operation and return events.
- int [getPollFdsNum](#) ()
- [PollFd *](#) [getPollFds](#) ()
- void [clear](#) ()

Private Member Functions

- int [nextFd](#) (int i)

Private Attributes

- int [ofdsNum](#)
The number of FD's in list.
- [PollFd *](#) [ofds](#)
The list of poll fd's.
- int [ofdsNext](#)
The next list entry for round robin polling.

7.65.1 Detailed Description

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

7.65.2 Member Typedef Documentation

7.65.2.1 typedef struct pollfd [BPoll::PollFd](#)

7.65.3 Constructor & Destructor Documentation

7.65.3.1 [BPoll::BPoll](#) ()

7.65.3.2 [BPoll::~~BPoll](#) ()

7.65.4 Member Function Documentation

7.65.4.1 void [BPoll::append](#) (int fd, int events = POLLIN|POLLERR|POLLHUP|POLLNVAL)

Append a file descriptor to polling list.

7.65.4.2 void BPoll::clear ()

7.65.4.3 void BPoll::delFd (int *fd*)

Remove a file descriptor from polling list.

7.65.4.4 BError BPoll::doPoll (int & *fd*, int *timeoutUs* = -1)

Perform polling operation.

7.65.4.5 BError BPoll::doPollEvents (int & *fd*, int & *events*, int *timeoutUs* = -1)

Perform polling operation and return events.

7.65.4.6 BPoll::PollFd * BPoll::getPollFds ()

7.65.4.7 int BPoll::getPollFdsNum ()

7.65.4.8 int BPoll::nextFd (int *i*) [private]

7.65.5 Member Data Documentation

7.65.5.1 PollFd* BPoll::ofds [private]

The list of poll fd's.

7.65.5.2 int BPoll::ofdsNext [private]

The next list entry for round robin polling.

7.65.5.3 int BPoll::ofdsNum [private]

The number of FD's in list.

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

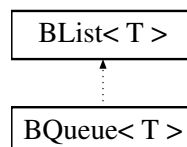
- [/src/cern/tms/beam/libBeam/BPoll.h](#)
- [/src/cern/tms/beam/libBeam/BPoll.cpp](#)

7.66 BQueue< T > Class Template Reference

Queue class.

```
#include <BQueue.h>
```

Inheritance diagram for BQueue< T >:



Public Member Functions

- [BQueue](#) ([BUInt](#) *size*)
- [~BQueue](#) ()
- void [clear](#) ()
Clear the queue.
- [BUInt](#) [writeAvailable](#) () const
- [BError](#) [write](#) (const T &v, [BTimeout](#) *timeout=BTimeoutForever*)
Append an item onto the queue.
- [BUInt](#) [readAvailable](#) () const
- [BError](#) [read](#) (T &v, [BTimeout](#) *timeout=BTimeoutForever*)
Get an item from the queue.

Private Attributes

- [BMutex](#) *oLock*
- [BUInt](#) *oSize*
- [BCondInt](#) *oNumber*

Additional Inherited Members

7.66.1 Detailed Description

`template<class T>class BQueue< T >`

Queue class.

7.66.2 Constructor & Destructor Documentation

7.66.2.1 `template<class T > BQueue< T >::BQueue (BUInt size)`

7.66.2.2 `template<class T > BQueue< T >::~~BQueue ()`

7.66.3 Member Function Documentation

7.66.3.1 `template<class T > void BQueue< T >::clear () [virtual]`

Clear the queue.

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

7.66.3.2 `template<class T> BError BQueue< T >::read (T & v, BTimeout timeout = BTimeoutForever)`

Get an item from the queue.

7.66.3.3 `template<class T > BUInt BQueue< T >::readAvailable () const`

7.66.3.4 `template<class T> BError BQueue< T >::write (const T & v, BTimeout timeout = BTimeoutForever)`

Append an item onto the queue.

7.66.3.5 `template<class T> BUInt BQueue<T>::writeAvailable () const`

7.66.4 Member Data Documentation

7.66.4.1 `template<class T> BMutex BQueue<T>::olock [private]`

7.66.4.2 `template<class T> BCondInt BQueue<T>::onumber [private]`

7.66.4.3 `template<class T> BUInt BQueue<T>::osize [private]`

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

- `/src/cern/tms/beam/libBeam/BQueue.h`

7.67 BRefData Class Reference

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

Public Member Functions

- [BRefData \(\)](#)
- [BRefData \(int len\)](#)
- [BRefData \(const BRefData &refData\)](#)
- [~BRefData \(\)](#)
- [BRefData * copy \(\)](#)
Create a copy of this reference for writing, if necessary.
- [BRefData * addRef \(\)](#)
Increment the reference counter.
- [int deleteRef \(\)](#)
Decrement the reference counter.
- [char * data \(\)](#)
Return the raw data pointer.
- [int len \(\)](#)
Return the length in bytes.
- [BRefData & operator= \(const BRefData &refData\)](#)
- [void setLen \(int len\)](#)
Set the length in bytes. Note should only be used if orefCount = 1.

Private Attributes

- [BAtomicCount orefCount](#)
The reference count, how many users.
- [int olen](#)
The actual length of data in oData.
- [void * odata](#)
Pointer to the data.

7.67.1 Detailed Description

Referenced data storage. This is Thread safe to a degree. The reference counting is protected. However, [setLen\(\)](#) is not and should be protected at a higher level.

7.67.2 Constructor & Destructor Documentation

7.67.2.1 `BRefData::BRefData ()`

7.67.2.2 `BRefData::BRefData (int len)`

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

7.67.2.4 `BRefData::~~BRefData ()`

7.67.3 Member Function Documentation

7.67.3.1 `BRefData * BRefData::addRef ()`

Increment the reference counter.

7.67.3.2 `BRefData * BRefData::copy ()`

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

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

Return the raw data pointer.

7.67.3.4 `int BRefData::deleteRef ()`

Decrement the reference counter.

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

Return the length in bytes.

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

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

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

7.67.4 Member Data Documentation

7.67.4.1 `void* BRefData::odata [private]`

Pointer to the data.

7.67.4.2 `int BRefData::olen [private]`

The actual length of data in `oData`.

7.67.4.3 BAtomicCount BRefData::orefCount [private]

The reference count, how many users.

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

- [/src/cern/tms/beam/libBeam/BRefData.h](#)
- [/src/cern/tms/beam/libBeam/BRefData.cpp](#)

7.68 BRtc Class Reference

Realtime clock.

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

Public Member Functions

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

Private Attributes

- int [ofd](#)
- int [orate](#)

7.68.1 Detailed Description

Realtime clock.

7.68.2 Constructor & Destructor Documentation

7.68.2.1 [BRtc::BRtc \(\)](#)

7.68.2.2 [BRtc::~BRtc \(\)](#)

7.68.3 Member Function Documentation

7.68.3.1 [BError BRtc::init \(int rate \)](#)

Setup interrupt rate.

7.68.3.2 [void BRtc::wait \(int delayUs \)](#)

Wait specified uS.

7.68.4 Member Data Documentation

7.68.4.1 `int BRtc::ofd` [private]

7.68.4.2 `int BRtc::orate` [private]

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

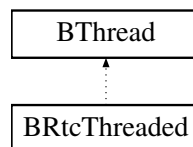
- [/src/cern/tms/beam/libBeam/BRtc.h](#)
- [/src/cern/tms/beam/libBeam/BRtc.cpp](#)

7.69 BRtcThreaded Class Reference

Threaded real time clock.

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

Inheritance diagram for BRtcThreaded:



Public Member Functions

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

Private Member Functions

- void * [function](#) ()

Private Attributes

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

7.69.1 Detailed Description

Threaded real time clock.

7.69.2 Constructor & Destructor Documentation

7.69.2.1 `BRtcThreaded::BRtcThreaded ()`

7.69.2.2 `BRtcThreaded::~~BRtcThreaded ()`

7.69.3 Member Function Documentation

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

Reimplemented from [BThread](#).

7.69.3.2 `BError BRtcThreaded::init (int rate)`

Setup interrupt rate.

7.69.3.3 `void BRtcThreaded::wait (int delayUs)`

Wait specified uS.

7.69.4 Member Data Documentation

7.69.4.1 `BCond BRtcThreaded::ocond` `[private]`

7.69.4.2 `int BRtcThreaded::orate` `[private]`

7.69.4.3 `BRtc BRtcThreaded::ortc` `[private]`

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

- `/src/cern/tms/beam/libBeam/BRtc.h`
- `/src/cern/tms/beam/libBeam/BRtc.cpp`

7.70 BRWLock Class Reference

thread read-write locks

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

Public Member Functions

- [BRWLock](#) ()
- [BRWLock](#) (const [BRWLock](#) &rwlock)
- [~BRWLock](#) ()
- `int rdLock ()`
Set lock, wait if necessary.
- `int tryRdLock ()`
Test the lock.
- `int wrLock ()`
Set lock, wait if necessary.
- `int tryWrLock ()`
Test the lock.

- int [unlock](#) ()
Unlock the lock.
- [BRWLock](#) & [operator=](#) (const [BRWLock](#) &rwlock)

Private Attributes

- pthread_rwlock_t [olock](#)

7.70.1 Detailed Description

thread read-write locks

7.70.2 Constructor & Destructor Documentation

7.70.2.1 [BRWLock::BRWLock](#) ()

7.70.2.2 [BRWLock::BRWLock](#) (const [BRWLock](#) & *rwlock*)

7.70.2.3 [BRWLock::~~BRWLock](#) ()

7.70.3 Member Function Documentation

7.70.3.1 [BRWLock](#) & [BRWLock::operator=](#) (const [BRWLock](#) & *rwlock*)

7.70.3.2 int [BRWLock::rdLock](#) ()

Set lock, wait if necessary.

7.70.3.3 int [BRWLock::tryRdLock](#) ()

Test the lock.

7.70.3.4 int [BRWLock::tryWrLock](#) ()

Test the lock.

7.70.3.5 int [BRWLock::unlock](#) ()

Unlock the lock.

7.70.3.6 int [BRWLock::wrLock](#) ()

Set lock, wait if necessary.

7.70.4 Member Data Documentation

7.70.4.1 pthread_rwlock_t [BRWLock::olock](#) [private]

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

- /src/cern/tms/beam/libBeam/[BRWLock.h](#)

- [/src/cern/tms/beam/libBeam/BRWLock.cpp](#)

7.71 BSema Class Reference

Sempahore class.

```
#include <BSema.h>
```

Public Member Functions

- [BSema](#) (int value=0)
- [BSema](#) (const [BSema](#) &sema)
- [~BSema](#) ()
- int [post](#) ()
Post condition.
- int [wait](#) ()
Wait for contition.
- int [timedWait](#) (int timeUs)
Wait for condition with timeout.
- int [tryWait](#) ()
Test for the condition.
- int [getValue](#) () const
- [BSema](#) & [operator=](#) (const [BSema](#) &sema)

Private Attributes

- [sem_t](#) [osema](#)

7.71.1 Detailed Description

Sempahore class.

7.71.2 Constructor & Destructor Documentation

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

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

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

7.71.3 Member Function Documentation

7.71.3.1 int [BSema::getValue](#) () const

7.71.3.2 [BSema](#) & [BSema::operator=](#) (const [BSema](#) & *sema*)

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

Post condition.

7.71.3.4 `int BSema::timedWait (int timeUs)`

Wait for condition with timeout.

7.71.3.5 `int BSema::tryWait ()`

Test for the condition.

7.71.3.6 `int BSema::wait ()`

Wait for contition.

7.71.4 Member Data Documentation

7.71.4.1 `sem_t BSema::osema [private]`

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

- [/src/cern/tms/beam/libBeam/BSema.h](#)
- [/src/cern/tms/beam/libBeam/BSema.cpp](#)

7.72 BSemaphore Class Reference

Semaphore class.

```
#include <BSemaphore.h>
```

Public Member Functions

- [BSemaphore \(\)](#)
- [BSemaphore \(const BSemaphore &semaphore\)](#)
- [~BSemaphore \(\)](#)
- [Bool wait \(BTimeout timeoutUs=BTimeoutForever\)](#)
Wait for the semaphore.
- `void set \(\)`
Set the semaphore.
- `int getValue \(\) const`
- [BSemaphore & operator= \(const BSemaphore &semaphore\)](#)

Private Attributes

- `sem_t osema`

7.72.1 Detailed Description

Semaphore class.

7.72.2 Constructor & Destructor Documentation

7.72.2.1 BSemaphore::BSemaphore ()

7.72.2.2 BSemaphore::BSemaphore (const BSemaphore & semaphore)

7.72.2.3 BSemaphore::~~BSemaphore ()

7.72.3 Member Function Documentation

7.72.3.1 int BSemaphore::getValue () const

7.72.3.2 BSemaphore & BSemaphore::operator= (const BSemaphore & semaphore)

7.72.3.3 void BSemaphore::set ()

Set the semaphore.

7.72.3.4 Bool BSemaphore::wait (BTimeout timeoutUs = BTimeoutForever)

Wait for the semaphore.

7.72.4 Member Data Documentation

7.72.4.1 sem_t BSemaphore::osema [private]

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

- [/src/cern/tms/beam/libBeam/BSemaphore.h](#)
- [/src/cern/tms/beam/libBeam/BSemaphore.cpp](#)

7.73 BSemaphoreCount Class Reference

```
#include <BSemaphore.h>
```

Public Member Functions

- [BSemaphoreCount](#) ()
- [BSemaphoreCount](#) (const [BSemaphoreCount](#) &semaphore)
- [~BSemaphoreCount](#) ()
- void [setValue](#) (BUInt v)
- void [add](#) (int v=1)
Set the semaphore.
- Bool [wait](#) (BUInt v=1, BTimeout timeoutUs=BTimeoutForever)
Wait for the semaphore.
- Bool [take](#) (BUInt v=1, BTimeout timeoutUs=BTimeoutForever)
Take for the semaphore.
- BUInt [value](#) ()
- [BSemaphoreCount](#) & [operator=](#) (const [BSemaphoreCount](#) &semaphore)

Private Attributes

- [BMutex olock](#)
- [BSemaphore osema](#)
- volatile [BUInt ovalue](#)

7.73.1 Constructor & Destructor Documentation

7.73.1.1 `BSemaphoreCount::BSemaphoreCount ()`

7.73.1.2 `BSemaphoreCount::BSemaphoreCount (const BSemaphoreCount & semaphore)`

7.73.1.3 `BSemaphoreCount::~~BSemaphoreCount ()`

7.73.2 Member Function Documentation

7.73.2.1 `void BSemaphoreCount::add (int v = 1)`

Set the semaphore.

7.73.2.2 `BSemaphoreCount & BSemaphoreCount::operator= (const BSemaphoreCount & semaphore)`

7.73.2.3 `void BSemaphoreCount::setValue (BUInt v)`

7.73.2.4 `Bool BSemaphoreCount::take (BUInt v = 1, BTimeout timeoutUs = BTimeoutForever)`

Take for the semaphore.

7.73.2.5 `BUInt BSemaphoreCount::value ()`

7.73.2.6 `Bool BSemaphoreCount::wait (BUInt v = 1, BTimeout timeoutUs = BTimeoutForever)`

Wait for the semaphore.

7.73.3 Member Data Documentation

7.73.3.1 `BMutex BSemaphoreCount::olock [private]`

7.73.3.2 `BSemaphore BSemaphoreCount::osema [private]`

7.73.3.3 `volatile BUInt BSemaphoreCount::ovalue [private]`

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

- [/src/cern/tms/beam/libBeam/BSemaphore.h](#)
- [/src/cern/tms/beam/libBeam/BSemaphore.cpp](#)

7.74 BSignal Class Reference

```
#include <SigGen.h>
```

Public Types

- enum { [NumChannels](#) = 9 }

Public Member Functions

- [BSignal](#) (int *id*=0, int *numSamples*=0, int *numRepeat*=0, int *nextId*=0)
- [BSignal](#) (const [BSignal](#) &*sig*)
- [~BSignal](#) ()
- [BSignal](#) & *operator=* (const [BSignal](#) &*sig*)

Public Attributes

- int *id*
- int *numSamples*
- int *numRepeat*
- int *nextId*
- [Sample](#) * *data* [[NumChannels](#)]

7.74.1 Member Enumeration Documentation

7.74.1.1 anonymous enum

Enumerator

NumChannels

7.74.2 Constructor & Destructor Documentation

7.74.2.1 [BSignal::BSignal](#) (int *id* = 0, int *numSamples* = 0, int *numRepeat* = 0, int *nextId* = 0)

7.74.2.2 [BSignal::BSignal](#) (const [BSignal](#) & *sig*)

7.74.2.3 [BSignal::~~BSignal](#) ()

7.74.3 Member Function Documentation

7.74.3.1 [BSignal](#) & [BSignal::operator=](#) (const [BSignal](#) & *sig*)

7.74.4 Member Data Documentation

7.74.4.1 [Sample](#)* [BSignal::data](#)[[NumChannels](#)]

7.74.4.2 int [BSignal::id](#)

7.74.4.3 int [BSignal::nextId](#)

7.74.4.4 int [BSignal::numRepeat](#)

7.74.4.5 int [BSignal::numSamples](#)

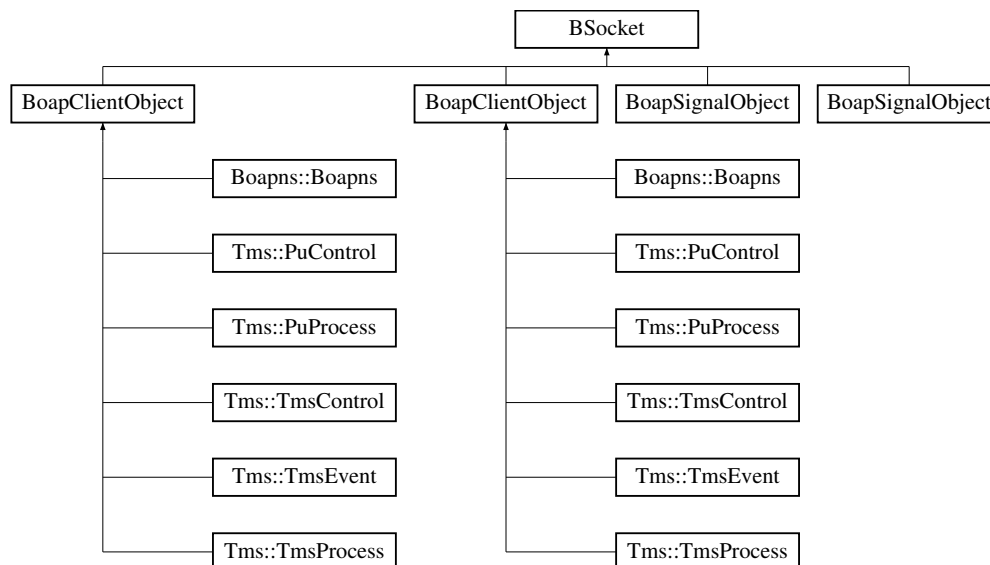
The documentation for this class was generated from the following files:

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

7.75 BSocket Class Reference

```
#include <BSocket.h>
```

Inheritance diagram for BSocket:



Public Types

- enum [NType](#) { [STREAM](#), [DGRAM](#) }
- enum [Priority](#) { [PriorityLow](#), [PriorityNormal](#), [PriorityHigh](#) }

Public Member Functions

- [BSocket](#) ()
- [BSocket](#) (int fd)
- [BSocket](#) (NType type)
- [BSocket](#) (int domain, int type, int protocol)
- [~BSocket](#) ()
- [BError init](#) (int domain, int type, int protocol)
- [BError init](#) (NType type)
- void [setFd](#) (int fd)
- int [getFd](#) ()
- [BError bind](#) (const [BSocketAddress](#) &add)
- [BError connect](#) (const [BSocketAddress](#) &add)
- [BError shutdown](#) (int how)
- [BError close](#) ()
- [BError listen](#) (int backlog=5)
- [BError accept](#) (int &fd)
- [BError accept](#) (int &fd, [BSocketAddress](#) &address)
- [BError send](#) (const void *buf, [BSize](#) nbytes, [BSize](#) &nbytesSent, int flags=0)
- [BError sendTo](#) (const [BSocketAddress](#) &address, const void *buf, [BSize](#) nbytes, [BSize](#) &nbytesSent, int flags=0)
- [BError recv](#) (void *buf, [BSize](#) maxbytes, [BSize](#) &nbytesRecv, int flags=0)
- [BError recvFrom](#) ([BSocketAddress](#) &address, void *buf, [BSize](#) maxbytes, [BSize](#) &nbytesRecv, int flags=0)
- [BError recvWithTimeout](#) (void *buf, [BSize](#) maxbytes, [BSize](#) &nbytesRecv, int timeout, int flags=0)

- [BError recvFromWithTimeout](#) ([BSocketAddress](#) &address, void *buf, [BSize](#) maxbytes, [BSize](#) &nbytesRecv, int timeout, int flags=0)
- [BError setSockOpt](#) (int level, int optname, void *optval, unsigned int optlen)
- [BError getSockOpt](#) (int level, int optname, void *optval, unsigned int *optlen)
- [BError setReuseAddress](#) (int on)
- [BError setBroadCast](#) (int on)
- [BError setPriority](#) ([Priority](#) priority)
- [BError getMTU](#) (uint32_t &mtu)
- [BError getAddress](#) ([BSocketAddress](#) &address)

Private Attributes

- int [osocket](#)

7.75.1 Member Enumeration Documentation

7.75.1.1 enum BSocket::NType

Enumerator

STREAM

DGRAM

7.75.1.2 enum BSocket::Priority

Enumerator

PriorityLow

PriorityNormal

PriorityHigh

7.75.2 Constructor & Destructor Documentation

7.75.2.1 [BSocket::BSocket](#) ()

7.75.2.2 [BSocket::BSocket](#) (int *fd*)

7.75.2.3 [BSocket::BSocket](#) (*NType type*)

7.75.2.4 [BSocket::BSocket](#) (int *domain*, int *type*, int *protocol*)

7.75.2.5 [BSocket::~~BSocket](#) ()

7.75.3 Member Function Documentation

7.75.3.1 [BError BSocket::accept](#) (int &*fd*)

7.75.3.2 [BError BSocket::accept](#) (int &*fd*, [BSocketAddress](#) &*address*)

7.75.3.3 [BError BSocket::bind](#) (const [BSocketAddress](#) &*add*)

7.75.3.4 [BError BSocket::close](#) ()

- 7.75.3.5 **BError** BSocket::connect (const BSocketAddress & add)
- 7.75.3.6 **BError** BSocket::getAddress (BSocketAddress & address)
- 7.75.3.7 int BSocket::getFd ()
- 7.75.3.8 **BError** BSocket::getMTU (uint32_t & mtu)
- 7.75.3.9 **BError** BSocket::getSockOpt (int level, int optname, void * optval, unsigned int * optlen)
- 7.75.3.10 **BError** BSocket::init (int domain, int type, int protocol)
- 7.75.3.11 **BError** BSocket::init (NType type)
- 7.75.3.12 **BError** BSocket::listen (int backlog = 5)
- 7.75.3.13 **BError** BSocket::recv (void * buf, BSize maxbytes, BSize & nbytesRecv, int flags = 0)
- 7.75.3.14 **BError** BSocket::recvFrom (BSocketAddress & address, void * buf, BSize maxbytes, BSize & nbytesRecv, int flags = 0)
- 7.75.3.15 **BError** BSocket::recvFromWithTimeout (BSocketAddress & address, void * buf, BSize maxbytes, BSize & nbytesRecv, int timeout, int flags = 0)
- 7.75.3.16 **BError** BSocket::recvWithTimeout (void * buf, BSize maxbytes, BSize & nbytesRecv, int timeout, int flags = 0)
- 7.75.3.17 **BError** BSocket::send (const void * buf, BSize nbytes, BSize & nbytesSent, int flags = 0)
- 7.75.3.18 **BError** BSocket::sendTo (const BSocketAddress & address, const void * buf, BSize nbytes, BSize & nbytesSent, int flags = 0)
- 7.75.3.19 **BError** BSocket::setBroadCast (int on)
- 7.75.3.20 void BSocket::setFd (int fd)
- 7.75.3.21 **BError** BSocket::setPriority (Priority priority)
- 7.75.3.22 **BError** BSocket::setReuseAddress (int on)
- 7.75.3.23 **BError** BSocket::setSockOpt (int level, int optname, void * optval, unsigned int optlen)
- 7.75.3.24 **BError** BSocket::shutdown (int how)

7.75.4 Member Data Documentation

- 7.75.4.1 int BSocket::osocket [private]

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

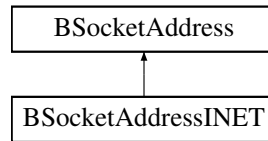
- /src/cern/tms/beam/libBeam/BSocket.h
- /src/cern/tms/beam/libBeam/BSocket.cpp

7.76 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

7.76.1 Detailed Description

Socket Address.

7.76.2 Member Typedef Documentation

7.76.2.1 typedef struct sockaddr [BSocketAddress::SockAddr](#)

7.76.3 Constructor & Destructor Documentation

7.76.3.1 [BSocketAddress::BSocketAddress](#) ()

7.76.3.2 [BSocketAddress::BSocketAddress](#) (const [BSocketAddress](#) & add)

7.76.3.3 [BSocketAddress::BSocketAddress](#) ([SockAddr](#) * address, int len)

7.76.3.4 [BSocketAddress::~BSocketAddress](#) ()

7.76.4 Member Function Documentation

7.76.4.1 `int BSocketAddress::len () const`

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

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

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

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

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

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

7.76.5 Member Data Documentation

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

7.76.5.2 `int BSocketAddress::olen [private]`

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

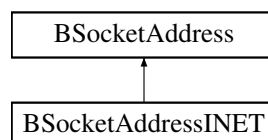
- [/src/cern/tms/beam/libBeam/BSocket.h](#)
- [/src/cern/tms/beam/libBeam/BSocket.cpp](#)

7.77 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.

7.77.1 Detailed Description

IP aware socket address.

7.77.2 Member Typedef Documentation

7.77.2.1 typedef struct sockaddr_in BSocketAddressINET::SockAddrIP

7.77.3 Member Function Documentation

7.77.3.1 [uint32_t BSocketAddressINET::address \(\)](#)

Returns socket ip address.

7.77.3.2 [BString BSocketAddressINET::getHostName \(\)](#) [static]

Get this hosts network name.

7.77.3.3 [BList< uint32_t > BSocketAddressINET::getIpAddresses \(\)](#) [static]

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

7.77.3.4 [BList< BString > BSocketAddressINET::getIpAddressList \(\)](#) [static]

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

7.77.3.5 [BList< BString > BSocketAddressINET::getIpAddressListAll \(\)](#) [static]

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

7.77.3.6 [BString BSocketAddressINET::getString \(\)](#)

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

7.77.3.7 `uint32_t BSocketAddressINET::port ()`

Returns socket port.

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

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

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

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

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

- [/src/cern/tms/beam/libBeam/BSocket.h](#)
- [/src/cern/tms/beam/libBeam/BSocket.cpp](#)

7.78 BSpI Class Reference

[BSpI](#) class.

```
#include <BSpI.h>
```

Public Types

- enum [Mode](#) { [Mode0](#) = 0, [Mode1](#) = 1, [Mode2](#) = 2, [Mode3](#) = 3 }

Public Member Functions

- [BSpI](#) ()
- [BError](#) [init](#) ([BString](#) devName, [BUInt](#) speed=1000000, [Mode](#) mode=[Mode1](#), [Bool](#) csActive=0)
- [BError](#) [transact](#) ([BUInt8](#) dev, void *txBuf, int txLen, int pad, void *rxBuf, int rxLen)

Private Attributes

- [BString](#) [odevName](#)
- int [odev](#)

7.78.1 Detailed Description

[BSpI](#) class.

7.78.2 Member Enumeration Documentation

7.78.2.1 enum [BSpI::Mode](#)

Enumerator

Mode0

Mode1

Mode2

Mode3

7.78.3 Constructor & Destructor Documentation

7.78.3.1 `BSpi::BSpi ()`

7.78.4 Member Function Documentation

7.78.4.1 `BError BSpi::init (BString devName, BUInt speed = 1000000, Mode mode = Mode1, Bool csActive = 0)`

7.78.4.2 `BError BSpi::transact (BUInt8 dev, void * txBuf, int txLen, int pad, void * rxBuf, int rxLen)`

7.78.5 Member Data Documentation

7.78.5.1 `int BSpi::odev [private]`

7.78.5.2 `BString BSpi::odevName [private]`

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

- [/src/cern/tms/beam/libBeam/BSpi.h](#)
- [/src/cern/tms/beam/libBeam/BSpi.cpp](#)

7.79 BString Class Reference

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

Public Member Functions

- `BString ()`
- `BString (const BString &string)`
- `BString (const char *str)`
- `BString (const char *str, unsigned int len)`
- `BString (char ch)`
- `BString (BInt v)`
- `BString (BUInt v)`
- `BString (BUInt64 v)`
- `BString (double v)`
- `~BString ()`
- `BString copy () const`
Return an independant copy.
- `int len () const`
Length of string.
- `const char * retStr () const`
Ptr to char representation.*
- `char * retStrDup () const`
Ptr to newly malloc'd char.*
- `int retInt () const`
Return string as a int.
- `unsigned int retUInt () const`
Return string as a int.
- `double retDouble () const`
Return string as a double.
- `int compare (const BString &string) const`

- Compare strings.*

 - int `compareWild` (const `BString` &string) const
Compare string to string with wildcards.
 - int `compareWildExpression` (const `BString` &string) const
Compare string to space delimited patterns.
 - int `compareRegex` (const `BString` &pattern, int ignoreCase=0) const
Compare strings.
- `BString` & `truncate` (int len)
Truncate to length len.
- `BString` & `pad` (int len)
Pad to length len.
- void `clear` ()
Clear the string.
- `BString` & `toUpper` ()
Convert to uppercase.
- `BString` & `toLower` ()
Convert to lowercase.
- `BString` `lowerFirst` ()
Return string with lowercase first character.
- void `removeNL` ()
Remove if present NL from last char.
- `BString` `justify` (int leftMargin, int width)
Justify the string to the given width.
- `BString` `fixedLen` (int length, int rightJustify=0)
return string formatted to fixed length
- `BString` `firstLine` ()
Return first line.
- `BString` `translateChar` (char ch, `BString` replace=" ")
Translate character converting them to the given string.
- `BString` `reverse` () const
Reverse character order.
- `BString` `substring` (int start, int len) const
Returns substring.
- int `del` (int start, int len)
Delete substring.
- int `insert` (int start, `BString` str)
Insert substring.
- int `append` (const `BString` &str)
Append a string.
- `BString` `add` (const `BString` &str) const
Add strings returning result.
- `BString` & `printf` (const char *fmt,...)
Formatted print into the string.
- int `find` (char ch) const
Find ch in string searching forwards.
- int `find` (`BString` str) const
Find string in string searching forwards.
- int `findReverse` (char ch) const
Find ch in string searching backwards.
- `BString` `csvEncode` () const
Encode a string for CSV.

- [BString & csvDecode](#) (const [BString](#) str)
Decode a string from CSV.
- [BString base64Encode](#) () const
Encode a string to base64.
- [BError base64Decode](#) ([BString](#) &str) const
Decode a string from base64.
- [BList< BString > getTokenList](#) ([BString](#) separators)
Break string into tokens.
- [BList< BString > getTokenList](#) (char separator)
Break string into tokens.
- [BString removeSeparators](#) ([BString](#) separators)
Remove any char from sepatators from string.
- [BString pullToken](#) ([BString](#) terminators)
Pull token from start of string.
- [BString pullSeparators](#) ([BString](#) separators)
Pull separators from start of string.
- [BString pullWord](#) ()
Pull a word out of the head of the string.
- [BString pullLine](#) ()
Pull a line out of the head of the string.
- [BList< BString > split](#) (char splitChar)
Split string into an array based on the character separator.
- [BString dirname](#) ()
- [BString basename](#) ()
- [BString extension](#) ()
- [BUInt32 hash](#) () const
- char & [get](#) (int pos)
- const char & [get](#) (int pos) const
- [BString & operator=](#) (const [BString](#) &string)
- char & [operator\[\]](#) (int pos)
- int [operator==](#) (const [BString](#) &s) const
- int [operator==](#) (const char *s) const
- int [operator>](#) (const [BString](#) &s) const
- int [operator>](#) (const char *s) const
- int [operator<](#) (const [BString](#) &s) const
- int [operator<](#) (const char *s) const
- int [operator>=](#) (const [BString](#) &s) const
- int [operator<=](#) (const [BString](#) &s) const
- int [operator!=](#) (const [BString](#) &s) const
- int [operator!=](#) (const char *s) const
- [BString operator+](#) (const [BString](#) &s) const
- [BString operator+](#) (const char *s) const
- [BString operator+=](#) (const [BString](#) &s)
- [BString operator+=](#) (const char *s)
- [BString operator+](#) (char ch) const
- [BString operator+](#) ([BInt](#) i) const
- [BString operator+](#) ([BUInt](#) i) const
- [BString operator+](#) ([BUInt64](#) i) const
- [operator const char *](#) () const
- [BString field](#) (int field) const
- char ** [fields](#) ()

Static Public Member Functions

- static [BString convert](#) (char ch)
Converts char to string.
- static [BString convert](#) (BInt value)
Converts int to string.
- static [BString convert](#) (BUInt value)
Converts uint to string.
- static [BString convert](#) (double value, int eFormat=0)
Converts double to string.
- static [BString convert](#) (BUInt64 value)
Converts long long to string.
- static [BString convertHex](#) (BInt value)
Converts int to string as hex value.
- static [BString convertHex](#) (BUInt value)
Converts uint to string as hex value.

Protected Attributes

- [BRefData](#) * ostr

Private Member Functions

- void [init](#) (const char *str)
- int [inString](#) (int pos) const
- int [isSpace](#) (char ch) const

7.79.1 Constructor & Destructor Documentation

7.79.1.1 [BString::BString](#) ()

7.79.1.2 [BString::BString](#) (const [BString](#) & *string*)

7.79.1.3 [BString::BString](#) (const char * *str*)

7.79.1.4 [BString::BString](#) (const char * *str*, unsigned int *len*)

7.79.1.5 [BString::BString](#) (char *ch*)

7.79.1.6 [BString::BString](#) (BInt *v*)

7.79.1.7 [BString::BString](#) (BUInt *v*)

7.79.1.8 [BString::BString](#) (BUInt64 *v*)

7.79.1.9 [BString::BString](#) (double *v*)

7.79.1.10 [BString::~~BString](#) ()

7.79.2 Member Function Documentation

7.79.2.1 [BString](#) [BString::add](#) (const [BString](#) & *str*) const

Add strings returning result.

7.79.2.2 `int BString::append (const BString & str)`

Append a string.

7.79.2.3 `BError BString::base64Decode (BString & str) const`

Decode a string from base64.

7.79.2.4 `BString BString::base64Encode () const`

Encode a string to base64.

7.79.2.5 `BString BString::basename ()`

7.79.2.6 `void BString::clear ()`

Clear the string.

7.79.2.7 `int BString::compare (const BString & string) const`

Compare strings.

7.79.2.8 `int BString::compareRegex (const BString & pattern, int ignoreCase = 0) const`

Compare strings.

7.79.2.9 `int BString::compareWild (const BString & string) const`

Compare string to string with wildcards.

7.79.2.10 `int BString::compareWildExpression (const BString & string) const`

Compare string to space delimited patterns.

7.79.2.11 `BString BString::convert (char ch) [static]`

Converts char to string.

7.79.2.12 `BString BString::convert (BInt value) [static]`

Converts int to string.

7.79.2.13 `BString BString::convert (BUInt value) [static]`

Converts uint to string.

7.79.2.14 `BString BString::convert (double value, int eFormat = 0) [static]`

Converts double to string.

7.79.2.15 **BString** **BString::convert** (**BUInt64** *value*) [static]

Converts long long to string.

7.79.2.16 **BString** **BString::convertHex** (**BInt** *value*) [static]

Converts int to string as hex value.

7.79.2.17 **BString** **BString::convertHex** (**BUInt** *value*) [static]

Converts uint to string as hex value.

7.79.2.18 **BString** **BString::copy** () const

Return an independant copy.

7.79.2.19 **BString** & **BString::csvDecode** (const **BString** *str*)

Decode a string from CSV.

7.79.2.20 **BString** **BString::csvEncode** () const

Encode a string for CSV.

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

Delete substring.

7.79.2.22 **BString** **BString::dirname** ()

7.79.2.23 **BString** **BString::extension** ()

7.79.2.24 **BString** **BString::field** (**int** *field*) const

7.79.2.25 **char **** **BString::fields** ()

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

Find *ch* in string searching forwards.

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

Find string in string searching forwards.

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

Find *ch* in string searching backwards.

7.79.2.29 **BString** **BString::firstLine** ()

Return first line.

7.79.2.30 **BString** **BString::fixedLen** (*int length*, *int rightJustify = 0*)

return string formatted to fixed length

7.79.2.31 **char &** **BString::get** (*int pos*)

7.79.2.32 **const char &** **BString::get** (*int pos*) **const**

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

Break string into tokens.

7.79.2.34 **BList< BString >** **BString::getTokenList** (**char separator**)

Break string into tokens.

7.79.2.35 **BUInt32** **BString::hash** () **const**

7.79.2.36 **void** **BString::init** (**const char * str**) [private]

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

Insert substring.

7.79.2.38 **int** **BString::inString** (*int pos*) **const** [private]

7.79.2.39 **int** **BString::isSpace** (**char ch**) **const** [private]

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

Justify the string to the given width.

7.79.2.41 **int** **BString::len** () **const**

Length of string.

7.79.2.42 **BString** **BString::lowerFirst** ()

Return string with lowercase first character.

7.79.2.43 **BString::operator const char * () const** [inline]

7.79.2.44 **int** **BString::operator!= (const BString & s) const** [inline]

7.79.2.45 **int** **BString::operator!= (const char * s) const** [inline]

7.79.2.46 **BString** **BString::operator+ (const BString & s) const** [inline]

7.79.2.47 **BString** **BString::operator+ (const char * s) const** [inline]

7.79.2.48 **BString** **BString::operator+ (char ch) const** [inline]

- 7.79.2.49 **BString BString::operator+(BInt *i*) const** [inline]
- 7.79.2.50 **BString BString::operator+(BUInt *i*) const** [inline]
- 7.79.2.51 **BString BString::operator+(BUInt64 *i*) const** [inline]
- 7.79.2.52 **BString BString::operator+=(const BString & *s*)** [inline]
- 7.79.2.53 **BString BString::operator+=(const char * *s*)** [inline]
- 7.79.2.54 **int BString::operator<(const BString & *s*) const** [inline]
- 7.79.2.55 **int BString::operator<(const char * *s*) const** [inline]
- 7.79.2.56 **int BString::operator<=(const BString & *s*) const** [inline]
- 7.79.2.57 **BString & BString::operator=(const BString & *string*)**
- 7.79.2.58 **int BString::operator==(const BString & *s*) const** [inline]
- 7.79.2.59 **int BString::operator==(const char * *s*) const** [inline]
- 7.79.2.60 **int BString::operator>(const BString & *s*) const** [inline]
- 7.79.2.61 **int BString::operator>(const char * *s*) const** [inline]
- 7.79.2.62 **int BString::operator>=(const BString & *s*) const** [inline]
- 7.79.2.63 **char & BString::operator[](int *pos*)**
- 7.79.2.64 **BString & BString::pad (int *len*)**

Pad to length *len*.

- 7.79.2.65 **BString & BString::printf (const char * *fmt*, ...)**

Formatted print into the string.

- 7.79.2.66 **BString BString::pullLine ()**

Pull a line out of the head of the string.

- 7.79.2.67 **BString BString::pullSeparators (BString *separators*)**

Pull separators from start of string.

- 7.79.2.68 **BString BString::pullToken (BString *terminators*)**

Pull token from start of string.

- 7.79.2.69 **BString BString::pullWord ()**

Pull a word out of the head of the string.

7.79.2.70 void BString::removeNL ()

Remove if present NL from last char.

7.79.2.71 BString BString::removeSeparators (BString *separators*)

Remove any char from separators from string.

7.79.2.72 double BString::retDouble () const

Return string as a double.

7.79.2.73 int BString::retInt () const

Return string as a int.

7.79.2.74 const char * BString::retStr () const

Ptr to char* representation.

7.79.2.75 char * BString::retStrDup () const

Ptr to newly malloc'd char*.

7.79.2.76 unsigned int BString::retUInt () const

Return string as a int.

7.79.2.77 BString BString::reverse () const

Reverse character order.

7.79.2.78 BList< BString > BString::split (char *splitChar*)

Split string into an array based on the character separator.

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

Returns substring.

7.79.2.80 BString & BString::toLower ()

Convert to lowercase.

7.79.2.81 BString & BString::toUpper ()

Convert to uppercase.

7.79.2.82 BString BString::translateChar (char *ch*, BString *replace* = " ")

Translate character converting them to the given string.

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

Truncate to length *len*.

7.79.3 Member Data Documentation

7.79.3.1 BRefData* BString::ostr [protected]

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

- [/src/cern/tms/beam/libBeam/BString.h](#)
- [/src/cern/tms/beam/libBeam/BString.cpp](#)

7.80 BStringLocked Class Reference

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

Public Member Functions

- [BStringLocked](#) ()
- [BStringLocked](#) (const [BStringLocked](#) &*s*)
- [BStringLocked](#) (const [BString](#) &*s*)
- int [len](#) () const
Length of string.
- [operator BString](#) () const
- [BStringLocked operator+](#) (const [BStringLocked](#) &*s*) const
- [BStringLocked & operator=](#) (const [BStringLocked](#) &*s*)

Private Attributes

- [BStringMutex](#) *olock*
- [BString](#) *ostr*

7.80.1 Constructor & Destructor Documentation

7.80.1.1 [BStringLocked::BStringLocked](#) () [inline]

7.80.1.2 [BStringLocked::BStringLocked](#) (const [BStringLocked](#) & *s*) [inline]

7.80.1.3 [BStringLocked::BStringLocked](#) (const [BString](#) & *s*) [inline]

7.80.2 Member Function Documentation

7.80.2.1 int [BStringLocked::len](#) () const [inline]

Length of string.

7.80.2.2 `BStringLocked::operator BString () const` [inline]

7.80.2.3 `BStringLocked BStringLocked::operator+ (const BStringLocked & s) const` [inline]

7.80.2.4 `BStringLocked& BStringLocked::operator= (const BStringLocked & s)` [inline]

7.80.3 Member Data Documentation

7.80.3.1 `BStringMutex BStringLocked::oLock` [mutable],[private]

7.80.3.2 `BString BStringLocked::ostr` [private]

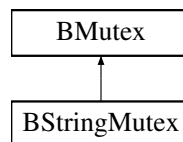
The documentation for this class was generated from the following file:

- [/src/cern/tms/beam/libBeam/BStringLocked.h](#)

7.81 BStringMutex Class Reference

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

Inheritance diagram for BStringMutex:



Public Member Functions

- [BStringMutex \(\)](#)

Additional Inherited Members

7.81.1 Constructor & Destructor Documentation

7.81.1.1 `BStringMutex::BStringMutex ()` [inline]

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

- [/src/cern/tms/beam/libBeam/BStringLocked.h](#)

7.82 BTable Class Reference

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

Public Member Functions

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

- void [setTitle](#) (BArray< BString > title)
- void [addRow](#) (BArray< BString > data)
- void [print](#) ()

Private Member Functions

- void [calculateWidths](#) ()
- void [printLine](#) (BArray< BString > line, int comment=0)

Private Attributes

- BArray< BString > [otitle](#)
- BList< BArray< BString > > [odata](#)
- BArray< int > [ocolumnWidths](#)

7.82.1 Constructor & Destructor Documentation

7.82.1.1 [BTable::BTable](#) ()

7.82.1.2 [BTable::~~BTable](#) ()

7.82.2 Member Function Documentation

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

7.82.2.2 void [BTable::calculateWidths](#) () [private]

7.82.2.3 void [BTable::clear](#) ()

7.82.2.4 void [BTable::print](#) ()

7.82.2.5 void [BTable::printLine](#) (BArray< BString > *line*, int *comment* = 0) [private]

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

7.82.3 Member Data Documentation

7.82.3.1 BArray<int> [BTable::ocolumnWidths](#) [private]

7.82.3.2 BList<BArray<BString>> [BTable::odata](#) [private]

7.82.3.3 BArray<BString> [BTable::otitle](#) [private]

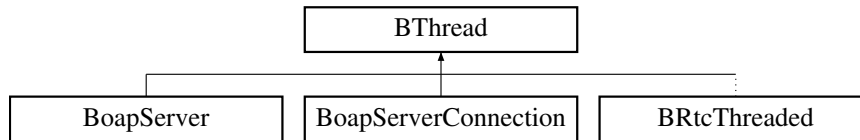
The documentation for this class was generated from the following files:

- [/src/cern/tms/beam/libBeam/BTable.h](#)
- [/src/cern/tms/beam/libBeam/BTable.cpp](#)

7.83 BThread Class Reference

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

Inheritance diagram for BThread:



Public Member Functions

- [BThread](#) ()
- virtual [~BThread](#) ()
- int [setInitPriority](#) (int policy, int priority)
- int [setInitStackSize](#) (size_t stackSize)
- int [start](#) ()
- void * [result](#) ()
- int [running](#) ()
- int [setPriority](#) (int policy, int priority)
- int [cancel](#) ()
- void * [waitForCompletion](#) ()
- pthread_t [getThread](#) ()
- virtual void * [function](#) ()

Static Private Member Functions

- static void * [startFunc](#) (void *)

Private Attributes

- pthread_t [othread](#)
- size_t [ostackSize](#)
- int [opolicy](#)
- int [opriority](#)
- int [orunning](#)
- void * [oreult](#)

7.83.1 Constructor & Destructor Documentation

7.83.1.1 [BThread::BThread](#) ()

7.83.1.2 [BThread::~~BThread](#) () [virtual]

7.83.2 Member Function Documentation

7.83.2.1 [int BThread::cancel](#) ()

7.83.2.2 [void * BThread::function](#) () [virtual]

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

- 7.83.2.3 pthread_t BThread::getThread ()
- 7.83.2.4 void * BThread::result ()
- 7.83.2.5 int BThread::running ()
- 7.83.2.6 int BThread::setNnitPriority (int policy, int priority)
- 7.83.2.7 int BThread::setNnitStackSize (size_t stackSize)
- 7.83.2.8 int BThread::setPriority (int policy, int priority)
- 7.83.2.9 int BThread::start ()
- 7.83.2.10 void * BThread::startFunc (void * arg) [static],[private]
- 7.83.2.11 void * BThread::waitForCompletion ()

7.83.3 Member Data Documentation

- 7.83.3.1 int BThread::opolicy [private]
- 7.83.3.2 int BThread::opriority [private]
- 7.83.3.3 void* BThread::oresult [private]
- 7.83.3.4 int BThread::orunning [private]
- 7.83.3.5 size_t BThread::ostackSize [private]
- 7.83.3.6 pthread_t BThread::othread [private]

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

- [/src/cern/tms/beam/libBeam/BThread.h](#)
- [/src/cern/tms/beam/libBeam/BThread.cpp](#)

7.84 BTime Class Reference

```
#include <BTime.h>
```

Public Member Functions

- [BTime](#) (BUInt32 t=0)
- void [set](#) (BUInt32 seconds)
Set the date and time.
- void [set](#) (BUInt year, BUInt month, BUInt day, BUInt hour=0, BUInt minute=0, BUInt second=0)
Set the date and time.
- void [setYearDay](#) (BUInt year, BUInt yearDay, BUInt hour=0, BUInt minute=0, BUInt second=0)
Set the date and time.
- void [getDate](#) (BUInt &year, BUInt &month, BUInt &day) const
Retun the date information.
- void [getTime](#) (BUInt &hour, BUInt &minute, BUInt &second) const

Return the time information.

- BUInt32 `getSeconds` () const
Return the number of seconds.
- int `isSet` () const
Check if set.
- int `isLeapYear` ()
Returns if a leap year.
- void `addSeconds` (int seconds)
Add the given number of seconds.
- BString `getString` (BString format="isoT") const
Gets the date/time in string format.
- BError `setString` (const BString dateTime)
Sets the date/time from string format.
- int `operator==` (const BTime &time) const
- int `operator!=` (const BTime &time) const
- int `operator>` (const BTime &time) const
- int `operator>=` (const BTime &time) const
- int `operator<` (const BTime &time) const
- int `operator<=` (const BTime &time) const
- BTime `operator+` (int seconds) const
- BTime & `operator+=` (int seconds)

Private Attributes

- BUInt32 `otime`
Time in seconds since 1970. range 1970-01-02 to 2106-02-07.

7.84.1 Constructor & Destructor Documentation

7.84.1.1 BTime::BTime (BUInt32 t = 0)

7.84.2 Member Function Documentation

7.84.2.1 void BTime::addSeconds (int seconds)

Add the given number of seconds.

7.84.2.2 void BTime::getDate (BUInt & year, BUInt & month, BUInt & day) const

Return the date information.

7.84.2.3 BUInt32 BTime::getSeconds () const

Return the number of seconds.

7.84.2.4 BString BTime::getString (BString format = "isoT") const

Gets the date/time in string format.

7.84.2.5 `void BTime::getTime (BUInt & hour, BUInt & minute, BUInt & second) const`

Return the time information.

7.84.2.6 `int BTime::isLeapYear ()`

Returns if a leap year.

7.84.2.7 `int BTime::isSet () const [inline]`

Check if set.

7.84.2.8 `int BTime::operator!=(const BTime & time) const [inline]`

7.84.2.9 `BTime BTime::operator+ (int seconds) const [inline]`

7.84.2.10 `BTime& BTime::operator+=(int seconds) [inline]`

7.84.2.11 `int BTime::operator< (const BTime & time) const [inline]`

7.84.2.12 `int BTime::operator<= (const BTime & time) const [inline]`

7.84.2.13 `int BTime::operator== (const BTime & time) const [inline]`

7.84.2.14 `int BTime::operator> (const BTime & time) const [inline]`

7.84.2.15 `int BTime::operator>= (const BTime & time) const [inline]`

7.84.2.16 `void BTime::set (BUInt32 seconds)`

Set the date and time.

7.84.2.17 `void BTime::set (BUInt year, BUInt month, BUInt day, BUInt hour = 0, BUInt minute = 0, BUInt second = 0)`

Set the date and time.

7.84.2.18 `BError BTime::setString (const BString dateTime)`

Sets the date/time from string format.

7.84.2.19 `void BTime::setYearDay (BUInt year, BUInt yearDay, BUInt hour = 0, BUInt minute = 0, BUInt second = 0)`

Set the date and time.

7.84.3 Member Data Documentation

7.84.3.1 `BUInt32 BTime::otime [private]`

Time in seconds since 1970. range 1970-01-02 to 2106-02-07.

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

- [/src/cern/tms/beam/libBeam/BTime.h](#)

- [/src/cern/tms/beam/libBeam/BTime.cpp](#)

7.85 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](#)

7.85.1 Detailed Description

Stopwatch style timer.

7.85.2 Constructor & Destructor Documentation

7.85.2.1 `BTimer::BTimer ()`

7.85.2.2 `BTimer::~~BTimer ()`

7.85.3 Member Function Documentation

7.85.3.1 `void BTimer::add (BTimer & timer)`

Add two timers.

7.85.3.2 `double BTimer::average ()`

Average time is duration between `start()` and `stop()` / number of stops.

7.85.3.3 `void BTimer::clear ()`

Clear timer.

7.85.3.4 `double BTimer::getElapsedTime ()`

Returns the elapsed time from the last start.

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

7.85.3.6 `double BTimer::peak ()`

Peak time.

7.85.3.7 `void BTimer::start ()`

Start timer.

7.85.3.8 `void BTimer::stop ()`

Stop timer.

7.85.4 Member Data Documentation

7.85.4.1 `double BTimer::oaverage [private]`

7.85.4.2 `double BTimer::oendTime [private]`

7.85.4.3 `BMutex BTimer::olock [private]`

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

7.85.4.5 `double BTimer::opeak [private]`

7.85.4.6 double BTimer::ostartTime [private]

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

- /src/cern/tms/beam/libBeam/BTimer.h
- /src/cern/tms/beam/libBeam/BTimer.cpp

7.86 BTimeStamp Class Reference

```
#include <BTimeStamp.h>
```

Public Member Functions

- [BTimeStamp](#) ()
- [BTimeStamp](#) (int year, int month=1, int day=1, int hour=0, int minute=0, int second=0, int microsecond=0)
- [BTimeStamp](#) (const BString str)
- [~BTimeStamp](#) ()
- void [clear](#) ()
 - Clear the date/time.*
- void [setFirst](#) ()
 - Set the first date available.*
- void [setLast](#) ()
 - Set the last date available.*
- void [set](#) (time_t time, int microseconds)
 - Set time using Unix time (seconds from 1970-01-01)*
- void [set](#) (int year=0, int month=1, int day=1, int hour=0, int minute=0, int second=0, int microsecond=0)
- void [set](#) (const BTimeStampMs &timeStamp)
 - Set the timeStamp to given MS time stamp.*
- void [setYDay](#) (int year=0, int yday=0, int hour=0, int minute=0, int second=0, int microsecond=0)
- void [setTime](#) (int hour=0, int minute=0, int second=0, int microsecond=0)
- void [setNow](#) ()
 - Set the timeStamp to now.*
- int [year](#) () const
- int [yday](#) () const
- int [month](#) () const
- int [day](#) () const
- int [hour](#) () const
- int [minute](#) () const
- int [second](#) () const
- int [microSecond](#) () const
- void [getDate](#) (int &year, int &mon, int &day) const
- [BString](#) [getString](#) (BString separator="T") const
 - Get the time as an ISO date/time string.*
- [BError](#) [setString](#) (const BString dateTime)
 - Set the time from an ISO date/time.*
- [BString](#) [getStringNoMs](#) (BString separator="T") const
 - Get the time as an ISO date/time string without microseconds.*
- [BString](#) [getStringFormatted](#) (BString format) const
 - Gets the time in a string form as per the format. Format syntax as per strftime()*
- void [addMilliSeconds](#) (int milliSeconds)
 - Add the given number of milli seconds. This should be less than a year.*

- void `addMicroSeconds` (int64_t microSeconds)
Add the given number of micro seconds. This should be less than a year.
- void `addSeconds` (int seconds)
Add the given number of seconds. This should be less than a year.
- uint32_t `getYearSeconds` () const
Get number of seconds within the year.
- uint64_t `getYearMicroSeconds` () const
Get number of micro seconds within the year.
- int `isSet` () const
- int `compare` (const `BTimeStamp` &timeStamp) const
Compare two dates.
- `operator BString` () const
- `BTimeStamp` & `operator=` (const `BTimeStampMs` &timeStamp)
- int `operator==` (const `BTimeStamp` &timeStamp) const
- int `operator!=` (const `BTimeStamp` &timeStamp) const
- int `operator>` (const `BTimeStamp` &timeStamp) const
- int `operator>=` (const `BTimeStamp` &timeStamp) const
- int `operator<` (const `BTimeStamp` &timeStamp) const
- int `operator<=` (const `BTimeStamp` &timeStamp) const

Static Public Member Functions

- static int `isLeap` (int year)
- static `Blnt64` `difference` (`BTimeStamp` t2, `BTimeStamp` t1)

Public Attributes

- uint16_t `oyear`
Year (0 .. 65535)
- uint16_t `oyday`
Day in year (0 .. 365)
- uint8_t `ohour`
Hour (0 .. 23)
- uint8_t `ominute`
Minute (0 .. 59)
- uint8_t `osecond`
Second (0 .. 59)
- uint8_t `ospare`
Padding.
- uint32_t `omicroSecond`
MicroSecond (0 .. 999999)

7.86.1 Constructor & Destructor Documentation

7.86.1.1 `BTimeStamp::BTimeStamp` ()

7.86.1.2 `BTimeStamp::BTimeStamp` (int year, int month = 1, int day = 1, int hour = 0, int minute = 0, int second = 0, int microsecond = 0)

7.86.1.3 `BTimeStamp::BTimeStamp` (const `BString` str)

7.86.1.4 `BTimeStamp::~~BTimeStamp ()`

7.86.2 Member Function Documentation

7.86.2.1 `void BTimeStamp::addMicroSeconds (int64_t microSeconds)`

Add the given number of micro seconds. This should be less that a year.

7.86.2.2 `void BTimeStamp::addMilliSeconds (int milliSeconds)`

Add the given number of milli seconds. This should be less that a year.

7.86.2.3 `void BTimeStamp::addSeconds (int seconds)`

Add the given number of seconds. This should be less that a year.

7.86.2.4 `void BTimeStamp::clear ()`

Clear the date/time.

7.86.2.5 `int BTimeStamp::compare (const BTimeStamp & timeStamp) const`

Compare two dates.

7.86.2.6 `int BTimeStamp::day () const`

7.86.2.7 `BInt64 BTimeStamp::difference (BTimeStamp t2, BTimeStamp t1) [static]`

7.86.2.8 `void BTimeStamp::getDate (int & year, int & mon, int & day) const`

7.86.2.9 `BString BTimeStamp::getString (BString separator = "T") const`

Get the time as an ISO date/time string.

7.86.2.10 `BString BTimeStamp::getStringFormatted (BString format) const`

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

7.86.2.11 `BString BTimeStamp::getStringNoMs (BString separator = "T") const`

Get the time as an ISO date/time string without microseconds.

7.86.2.12 `uint64_t BTimeStamp::getYearMicroSeconds () const`

Get number of micro seconds within the year.

7.86.2.13 `uint32_t BTimeStamp::getYearSeconds () const`

Get number of seconds within the year.

- 7.86.2.14 `int BTimeStamp::hour () const`
- 7.86.2.15 `int BTimeStamp::isLeap (int year) [static]`
- 7.86.2.16 `int BTimeStamp::isSet () const [inline]`
- 7.86.2.17 `int BTimeStamp::microSecond () const`
- 7.86.2.18 `int BTimeStamp::minute () const`
- 7.86.2.19 `int BTimeStamp::month () const`
- 7.86.2.20 `BTimeStamp::operator BString () const [inline]`
- 7.86.2.21 `int BTimeStamp::operator!= (const BTimeStamp & timeStamp) const [inline]`
- 7.86.2.22 `int BTimeStamp::operator< (const BTimeStamp & timeStamp) const [inline]`
- 7.86.2.23 `int BTimeStamp::operator<= (const BTimeStamp & timeStamp) const [inline]`
- 7.86.2.24 `BTimeStamp & BTimeStamp::operator= (const BTimeStampMs & timeStamp) [inline]`
- 7.86.2.25 `int BTimeStamp::operator== (const BTimeStamp & timeStamp) const [inline]`
- 7.86.2.26 `int BTimeStamp::operator> (const BTimeStamp & timeStamp) const [inline]`
- 7.86.2.27 `int BTimeStamp::operator>= (const BTimeStamp & timeStamp) const [inline]`
- 7.86.2.28 `int BTimeStamp::second () const`
- 7.86.2.29 `void BTimeStamp::set (time_t time, int microseconds)`

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

- 7.86.2.30 `void BTimeStamp::set (int year = 0, int month = 1, int day = 1, int hour = 0, int minute = 0, int second = 0, int microsecond = 0)`

- 7.86.2.31 `void BTimeStamp::set (const BTimeStampMs & timeStamp)`

Set the timeStamp to given MS time stamp.

- 7.86.2.32 `void BTimeStamp::setFirst ()`

Set the first date available.

- 7.86.2.33 `void BTimeStamp::setLast ()`

Set the last date available.

- 7.86.2.34 `void BTimeStamp::setNow ()`

Set the timeStamp to now.

7.86.2.35 BError BTimeStamp::setString (const BString *dateTime*)

Set the time from an ISO date/time.

7.86.2.36 void BTimeStamp::setTime (int *hour* = 0, int *minute* = 0, int *second* = 0, int *microsecond* = 0)

7.86.2.37 void BTimeStamp::setYDay (int *year* = 0, int *yday* = 0, int *hour* = 0, int *minute* = 0, int *second* = 0, int *microsecond* = 0)

7.86.2.38 int BTimeStamp::yday () const

7.86.2.39 int BTimeStamp::year () const

7.86.3 Member Data Documentation

7.86.3.1 uint8_t BTimeStamp::ohour

Hour (0 .. 23)

7.86.3.2 uint32_t BTimeStamp::omicroSecond

MicroSecond (0 .. 999999)

7.86.3.3 uint8_t BTimeStamp::ominate

Minute (0 .. 59)

7.86.3.4 uint8_t BTimeStamp::osecond

Second (0 .. 59)

7.86.3.5 uint8_t BTimeStamp::ospare

Padding.

7.86.3.6 uint16_t BTimeStamp::oyday

Day in year (0 .. 365)

7.86.3.7 uint16_t BTimeStamp::oyear

Year (0 .. 65535)

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

- [/src/cern/tms/beam/libBeam/BTimeStamp.h](#)
- [/src/cern/tms/beam/libBeam/BTimeStamp.cpp](#)

7.87 BTimeStampMs Class Reference

```
#include <BTimeStampMs.h>
```

Public Member Functions

- [BTimeStampMs](#) ([BString](#) str="")
- [~BTimeStampMs](#) ()
- void [clear](#) ()
 - Clear the date/time.*
- void [setNow](#) ()
 - Set the timeStamp to now.*
- [BTimeStampMs](#) & [addMilliseconds](#) (int milliseconds)
 - Add the given number of milli seconds. This should be less than a year.*
- [BTimeStampMs](#) & [subMilliseconds](#) (int milliseconds)
 - Add the given number of milli seconds. This should be less than a year.*
- [BTimeStampMs](#) & [addSeconds](#) (int seconds)
 - Add the given number of seconds. This should be less than a year.*
- [BTimeStampMs](#) & [subSeconds](#) (int seconds)
 - Subtract the given number of seconds. This should be less than a year.*
- [uint32_t](#) [getYearSeconds](#) ()
 - Get number of seconds within the year.*
- [uint64_t](#) [getYearMilliseconds](#) ()
 - Get number of seconds within the year.*
- [BString](#) [getString](#) ([BString](#) separator="T")
 - Get the time as an ISO date/time string.*
- [BString](#) [getStringNoMs](#) ([BString](#) separator="T")
 - Get the time as an ISO date/time string with no ms.*
- [BError](#) [setString](#) ([BString](#) dateTime)
 - Set the time from an ISO date/time.*
- [BString](#) [getDurationString](#) ([BString](#) separator="T")
 - Get the time as an ISO date/time string but with month's and days starting from 0.*
- [BString](#) [getDurationStringNoMs](#) ([BString](#) separator="T")
 - Get the time as an ISO date/time string but with month's and days starting from 0 with no ms.*
- [BError](#) [setDurationString](#) ([BString](#) dateTime)
 - Set the time from an ISO date/time string but with month's and days starting from 0.*
- [BString](#) [getStringRaw](#) ()
- void [getDate](#) (int &year, int &mon, int &day)
 - Get the year, month and day.*
- int [compare](#) (const [BTimeStampMs](#) &timeStamp)
 - Compare two dates.*
- int [operator>](#) (const [BTimeStampMs](#) &timeStamp)
- int [operator>=](#) (const [BTimeStampMs](#) &timeStamp)
- int [operator<](#) (const [BTimeStampMs](#) &timeStamp)
- int [operator<=](#) (const [BTimeStampMs](#) &timeStamp)

Static Public Member Functions

- static int [isLeap](#) (int year)
- static [BUInt64](#) [difference](#) ([BTimeStampMs](#) t2, [BTimeStampMs](#) t1)

Public Attributes

- uint16_t [year](#)
Year (2000 .. 3000)
- uint16_t [yday](#)
Day in year (0 .. 365)
- uint16_t [hour](#)
Hour (0 .. 23)
- uint16_t [minute](#)
Minute (0 .. 59)
- uint16_t [second](#)
Second (0 .. 59)
- uint16_t [milliSecond](#)
MilliSecond (0 .. 999)
- int32_t [sampleNumber](#)
The sample number this time refers to.

7.87.1 Constructor & Destructor Documentation

7.87.1.1 `BTimeStampMs::BTimeStampMs (BString str = " ")`

7.87.1.2 `BTimeStampMs::~~BTimeStampMs ()`

7.87.2 Member Function Documentation

7.87.2.1 `BTimeStampMs & BTimeStampMs::addMilliseconds (int milliseconds)`

Add the given number of milli seconds. This should be less that a year.

7.87.2.2 `BTimeStampMs & BTimeStampMs::addSeconds (int seconds)`

Add the given number of seconds. This should be less that a year.

7.87.2.3 `void BTimeStampMs::clear ()`

Clear the date/time.

7.87.2.4 `int BTimeStampMs::compare (const BTimeStampMs & timeStamp)`

Compare two dates.

7.87.2.5 `BUInt64 BTimeStampMs::difference (BTimeStampMs t2, BTimeStampMs t1)` `[static]`

7.87.2.6 `void BTimeStampMs::getDate (int & year, int & mon, int & day)`

Get the year, month and day.

7.87.2.7 `BString BTimeStampMs::getDurationString (BString separator = "T")`

Get the time as an ISO date/time string but with month's and days starting from 0.

7.87.2.8 BString BTimeStampMs::getDurationStringNoMs (BString separator = "T")

Get the time as an ISO date/time string but with month's and days starting from 0 with no ms.

7.87.2.9 BString BTimeStampMs::getString (BString separator = "T")

Get the time as an ISO date/time string.

7.87.2.10 BString BTimeStampMs::getStringNoMs (BString separator = "T")

Get the time as an ISO date/time string with no ms.

7.87.2.11 BString BTimeStampMs::getStringRaw ()**7.87.2.12 uint64_t BTimeStampMs::getYearMilliSeconds ()**

Get number of seconds within the year.

7.87.2.13 uint32_t BTimeStampMs::getYearSeconds ()

Get number of seconds within the year.

7.87.2.14 int BTimeStampMs::isLeap (int year) [static]**7.87.2.15 int BTimeStampMs::operator< (const BTimeStampMs & timeStamp) [inline]****7.87.2.16 int BTimeStampMs::operator<= (const BTimeStampMs & timeStamp) [inline]****7.87.2.17 int BTimeStampMs::operator> (const BTimeStampMs & timeStamp) [inline]****7.87.2.18 int BTimeStampMs::operator>= (const BTimeStampMs & timeStamp) [inline]****7.87.2.19 BError BTimeStampMs::setDurationString (BString dateTime)**

Set the time from an ISO date/time string but with month's and days starting from 0.

7.87.2.20 void BTimeStampMs::setNow ()

Set the timeStamp to now.

7.87.2.21 BError BTimeStampMs::setString (BString dateTime)

Set the time from an ISO date/time.

7.87.2.22 BTimeStampMs & BTimeStampMs::subMilliSeconds (int milliSeconds)

Add the given number of milli seconds. This should be less that a year.

7.87.2.23 BTimeStampMs & BTimeStampMs::subSeconds (int seconds)

Subtract the given number of seconds. This should be less that a year.

7.87.3 Member Data Documentation

7.87.3.1 uint16_t BTimeStampMs::hour

Hour (0 .. 23)

7.87.3.2 uint16_t BTimeStampMs::milliSecond

MilliSecond (0 .. 999)

7.87.3.3 uint16_t BTimeStampMs::minute

Minute (0 .. 59)

7.87.3.4 int32_t BTimeStampMs::sampleNumber

The sample number this time refers to.

7.87.3.5 uint16_t BTimeStampMs::second

Second (0 .. 59)

7.87.3.6 uint16_t BTimeStampMs::yday

Day in year (0 .. 365)

7.87.3.7 uint16_t BTimeStampMs::year

Year (2000 .. 3000)

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

- [/src/cern/tms/beam/libBeam/BTimeStampMs.h](#)
- [/src/cern/tms/beam/libBeam/BTimeStampMs.cpp](#)

7.88 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](#)

7.88.1 Detailed Description

Basic access to a Url.

7.88.2 Constructor & Destructor Documentation

7.88.2.1 [BUrl::BUrl \(\)](#)

7.88.2.2 [BUrl::~~BUrl \(\)](#)

7.88.3 Member Function Documentation

7.88.3.1 [BError BUrl::readString \(BString url, BString & str \)](#)

Reads URL.

7.88.3.2 [size_t BUrl::writeData \(void * data, size_t size, size_t e!Size, void * stream \)](#) `[static],[private]`

7.88.4 Member Data Documentation

7.88.4.1 [int BUrl::oinit](#) `[static],[private]`

7.88.4.2 [BString BUrl::ores](#) `[private]`

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

- [/src/cern/tms/beam/libBeam/BUrl.h](#)
- [/src/cern/tms/beam/libBeam/BUrl.cpp](#)

7.89 Tms::ConfigInfo Class Reference

This class describes the configuration of the TMS.

```
#include <TmsD.h>
```

Public Member Functions

- [ConfigInfo \(BArray< PuChannel > ppuReferences=BArray< PuChannel >\(\)\)](#)

Public Attributes

- [BArray< PuChannel > puReferences](#)

The logical to physical Pick-Up table. Each PuReference includes a Module Controller identifier, a Physical Pick-Up number and a Physical Channel.

7.89.1 Detailed Description

This class describes the configuration of the TMS.

7.89.2 Constructor & Destructor Documentation

7.89.2.1 `Tms::ConfigInfo::ConfigInfo (BArray< PuChannel > ppuReferences = BArray<PuChannel > ())`

7.89.3 Member Data Documentation

7.89.3.1 `BArray<PuChannel > Tms::ConfigInfo::puReferences`

The logical to physical Pick-Up table. Each PuReference includes a Module Controller identifier, a Physical Pick-Up number and a Physical Channel.

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

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

7.90 Tms::CycleInformation Class Reference

```
#include <TmsD.h>
```

Public Member Functions

- [CycleInformation](#) (`BUInt32 pcycleNumber=BUInt32()`, `BString pcycleType=BString()`, `BList< CycleInformationPeriod > pperiods=BList< CycleInformationPeriod >()`)

Public Attributes

- `BUInt32 cycleNumber`
The PS Cycle number.
- `BString cycleType`
The Cycle Type Name.
- `BList< CycleInformationPeriod > periods`
The list of cycle periods.

7.90.1 Constructor & Destructor Documentation

7.90.1.1 `Tms::CycleInformation::CycleInformation (BUInt32 pcycleNumber = BUInt32 () , BString pcycleType = BString () , BList< CycleInformationPeriod > pperiods = BList<CycleInformationPeriod > ())`

7.90.2 Member Data Documentation

7.90.2.1 `BUInt32 Tms::CycleInformation::cycleNumber`

The PS Cycle number.

7.90.2.2 `BString Tms::CycleInformation::cycleType`

The Cycle Type Name.

7.90.2.3 BList<CycleInformationPeriod > Tms::CycleInformation::periods

The list of cycle periods.

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

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

7.91 Tms::CycleInformationPeriod Class Reference

Cycle information.

```
#include <TmsD.h>
```

Public Member Functions

- [CycleInformationPeriod](#) ([BUInt32](#) pcyclePeriod=[BUInt32](#)(), [BUInt32](#) pstartTime=[BUInt32](#)(), [BUInt32](#) pendTime=[BUInt32](#)(), [BUInt32](#) pharmonic=[BUInt32](#)(), [BUInt32](#) pnumBunches=[BUInt32](#)(), [BUInt32](#) pbunchMask=[BUInt32](#)(), [BUInt32](#) pnumValues=[BUInt32](#)())

Public Attributes

- [BUInt32](#) cyclePeriod
The Cycle Period.
- [BUInt32](#) startTime
The start time in ms.
- [BUInt32](#) endTime
The end time in ms.
- [BUInt32](#) harmonic
The Machines harmonic number.
- [BUInt32](#) numBunches
The number of bunches.
- [BUInt32](#) bunchMask
Bitmask defining which buckets the bunches are captured from. Bit 0 is bucket 1, bit 1 is bucket 2 etc.
- [BUInt32](#) numValues
The total number of raw data values available.

7.91.1 Detailed Description

Cycle information.

7.91.2 Constructor & Destructor Documentation

- 7.91.2.1 [Tms::CycleInformationPeriod::CycleInformationPeriod](#) ([BUInt32](#) pcyclePeriod = [BUInt32](#) (), [BUInt32](#) pstartTime = [BUInt32](#) (), [BUInt32](#) pendTime = [BUInt32](#) (), [BUInt32](#) pharmonic = [BUInt32](#) (), [BUInt32](#) pnumBunches = [BUInt32](#) (), [BUInt32](#) pbunchMask = [BUInt32](#) (), [BUInt32](#) pnumValues = [BUInt32](#) ())

7.91.3 Member Data Documentation

7.91.3.1 BUInt32 Tms::CycleInformationPeriod::bunchMask

Bitmask defining which buckets the bunches are captured from. Bit 0 is bucket 1, bit 1 is bucket 2 etc.

7.91.3.2 BUInt32 Tms::CycleInformationPeriod::cyclePeriod

The Cycle Period.

7.91.3.3 BUInt32 Tms::CycleInformationPeriod::endTime

The end time in ms.

7.91.3.4 BUInt32 Tms::CycleInformationPeriod::harmonic

The Machines harmonic number.

7.91.3.5 BUInt32 Tms::CycleInformationPeriod::numBunches

The number of bunches.

7.91.3.6 BUInt32 Tms::CycleInformationPeriod::numValues

The total number of raw data values available.

7.91.3.7 BUInt32 Tms::CycleInformationPeriod::startTime

The start time in ms.

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

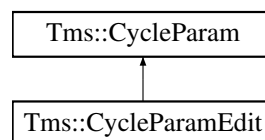
- [TmsD.h](#)
- [TmsD.cc](#)

7.92 Tms::CycleParam Class Reference

This class defines the parameters for a PS processing cycle.

```
#include <TmsD.h>
```

Inheritance diagram for Tms::CycleParam:



Public Member Functions

- [CycleParam](#) ([BString](#) pcycleType=[BString](#)(), [BString](#) pname=[BString](#)(), [BString](#) pinfo=[BString](#)(), [BUInt32](#) pring=[BUInt32](#)(), [BUInt32](#) pchannel=[BUInt32](#)(), [BUInt32](#) ppllCycleStartFrequency=[BUInt32](#)(), [BUInt32](#) ppllInitialFrequency=[BUInt32](#)(), [BUInt32](#) ppllInitialFrequencyDelay=[BUInt32](#)(), [BUInt32](#) ppllFrefGain=[BUInt32](#)(), [BUInt32](#) ppllGain=[BUInt32](#)(), [BUInt32](#) ppllDdsMinimum=[BUInt32](#)(), [BUInt32](#) ppllDdsMaximum=[BUInt32](#)(), [BUInt32](#) pstateDelay=[BUInt32](#)(), [BArray](#)< [BInt32](#) > pfrefPhaseDelay=[BArray](#)< [BInt32](#) >(), [BArray](#)< [PuStateTable](#) > pstateTable=[BArray](#)< [PuStateTable](#) >(), [BArray](#)< [BString](#) > psettings=[BArray](#)< [BString](#) >())

Public Attributes

- [BString cycleType](#)
The Cycle Type Name of this parameter set, normally the BEAM type the set of parameters is designed to measure.
- [BString name](#)
The name of this set of parameters.
- [BString info](#)
Information on this parameter set.
- [BUInt32 ring](#)
The ring this configuration is for, 0 defines all rings.
- [BUInt32 channel](#)
The channel number this configuration is for, 0 defines all channels.
- [BUInt32 pllCycleStartFrequency](#)
This defines the initial PLL frequency. This is loaded on START_CYCLE.
- [BUInt32 pllInitialFrequency](#)
This defines the initial PLL frequency. This is loaded after the delay given in pllInitialFrequencyDelay.
- [BUInt32 pllInitialFrequencyDelay](#)
This defines the delay in milliseconds from START_CYCLE when the pllInitialFrequency is loaded.
- [BUInt32 pllFrefGain](#)
The gain the FREF signal. This is a value in the range +-8191. A normal value would be around 4096.
- [BUInt32 pllGain](#)
The gain of the PLL feedback system. This is the gain reduction of the PLL in terms of right bit shifts. A bit shift of 7 is about unity gain.
- [BUInt32 pllDdsMinimum](#)
PLL DDS minimum frequency. If this and pllDdsMaximum is set to 0, this feature is disabled.
- [BUInt32 pllDdsMaximum](#)
PLL DDS maximum frequency. If this and pllDdsMinimum is set to 0, this feature is disabled.
- [BUInt32 stateDelay](#)
The delay in FREF periods when in the delay state.
- [BArray< BInt32 > frefPhaseDelay](#)
The phase delay parameters for the Fref timing signal for each of the Pick-Up channels. This is set based on the position of the Pick-Up's in the PS ring. Its value is based of Fref / 512.
- [BArray< PuStateTable > stateTable](#)
The array of State Table entries for the processing run.
- [BArray< BString > settings](#)
A string array defining the settings for the states. Used for [CycleParam](#) editors.

7.92.1 Detailed Description

This class defines the parameters for a PS processing cycle.

7.92.2 Constructor & Destructor Documentation

- 7.92.2.1 `Tms::CycleParam::CycleParam (BString pcycleType = BString (), BString pname = BString (), BString pinfo = BString (), BUInt32 pring = BUInt32 (), BUInt32 pchannel = BUInt32 (), BUInt32 ppllCycleStartFrequency = BUInt32 (), BUInt32 ppllInitialFrequency = BUInt32 (), BUInt32 ppllInitialFrequencyDelay = BUInt32 (), BUInt32 ppllFrefGain = BUInt32 (), BUInt32 ppllGain = BUInt32 (), BUInt32 ppllDdsMinimum = BUInt32 (), BUInt32 ppllDdsMaximum = BUInt32 (), BUInt32 pstateDelay = BUInt32 (), BArray< BInt32 > pfrefPhaseDelay = BArray< BInt32 > (), BArray< PuStateTable > pstateTable = BArray< PuStateTable > (), BArray< BString > psettings = BArray< BString > ())`

7.92.3 Member Data Documentation

7.92.3.1 BUInt32 Tms::CycleParam::channel

The channel number this configuration is for, 0 defines all channels.

7.92.3.2 BString Tms::CycleParam::cycleType

The Cycle Type Name of this parameter set, normally the BEAM type the set of parameters is designed to measure.

7.92.3.3 BArray<BInt32 > Tms::CycleParam::frefPhaseDelay

The phase delay parameters for the Fref timing signal for each of the Pick-Up channels. This is set based on the position of the Pick-Up's in the PS ring. Its value is based of Fref / 512.

7.92.3.4 BString Tms::CycleParam::info

Information on this parameter set.

7.92.3.5 BString Tms::CycleParam::name

The name of this set of parameters.

7.92.3.6 BUInt32 Tms::CycleParam::pllCycleStartFrequency

This defines the initial PLL frequency. This is loaded on START_CYCLE.

7.92.3.7 BUInt32 Tms::CycleParam::pllDdsMaximum

PLL DDS maximum frequency. If this and pllDdsMinimum is set to 0, this feature is disabled.

7.92.3.8 BUInt32 Tms::CycleParam::pllDdsMinimum

PLL DDS minimum frequency. If this and pllDdsMaximum is set to 0, this feature is disabled.

7.92.3.9 BUInt32 Tms::CycleParam::pllFrefGain

The gain the FREF signal. This is a value in the range +-8191. A normal value would be around 4096.

7.92.3.10 BUInt32 Tms::CycleParam::pllGain

The gain of the PLL feedback system. This is the gain reduction of the PLL in terms of right bit shifts. A bit shift of 7 is about unity gain.

7.92.3.11 BUInt32 Tms::CycleParam::pllInitialFrequency

This defines the initial PLL frequency. This is loaded after the delay given in pllInitialFrequencyDelay.

7.92.3.12 BUInt32 Tms::CycleParam::pllInitialFrequencyDelay

This defines the delay in milliseconds from START_CYCLE when the pllInitialFrequency is loaded.

7.92.3.13 BUInt32 Tms::CycleParam::ring

The ring this configuration is for, 0 defines all rings.

7.92.3.14 BArray<BString > Tms::CycleParam::settings

A string array defining the settings for the states. Used for [CycleParam](#) editors.

7.92.3.15 BUInt32 Tms::CycleParam::stateDelay

The delay in FREF periods when in the delay state.

7.92.3.16 BArray<PuStateTable > Tms::CycleParam::stateTable

The array of State Table entries for the processing run.

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

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

7.93 Tms::CycleParamDb Class Reference

Internal CycleParameter management class.

```
#include <TmsLib.h>
```

Public Member Functions

- [CycleParamDb](#) (BString baseDir=".")
- [BError getCycleTypes](#) (BList< BString > &typeList)
Get the list of CycleParameter types in the directory.
- [BError getFileNames](#) (BList< BString > &fileList)
Get a list of all of the CycleParameter file names.
- [BError getCycleParams](#) (BString fileName, Tms::CycleParam ¶m)
Get the CycleParameter from the given file name.
- [BError setCycleParams](#) (Tms::CycleParam param)
Set the CycleParameters. Writes to the appropriate file name.
- [BError deleteCycleParams](#) (BString cycleType, BUInt32 ring, BUInt32 puChannel)
Deletes all CycleParameter definition files for the given ring/channel.
- [BError readCycleParams](#) (BString fileName, Tms::CycleParam ¶m)
Reads a set of CycleParameters from a file.
- [BError writeCycleParams](#) (BString fileName, Tms::CycleParam param)
Writes a set of CycleParameters to a file.

Private Attributes

- [BString obaseDir](#)

7.93.1 Detailed Description

Internal CycleParameter management class.

7.93.2 Constructor & Destructor Documentation

7.93.2.1 `Tms::CycleParamDb::CycleParamDb (BString baseDir = " . ")`

7.93.3 Member Function Documentation

7.93.3.1 `BError Tms::CycleParamDb::deleteCycleParams (BString cycleType, BUInt32 ring, BUInt32 puChannel)`

Deletes all CycleParameter definition files for the given ring/channel.

7.93.3.2 `BError Tms::CycleParamDb::getCycleParams (BString fileName, Tms::CycleParam & param)`

Get the CycleParameter from the given file name.

7.93.3.3 `BError Tms::CycleParamDb::getCycleTypes (BList< BString > & typeList)`

Get the list of CycleParameter types in the directory.

7.93.3.4 `BError Tms::CycleParamDb::getFileNames (BList< BString > & fileList)`

Get a list of all of the CycleParameter file names.

7.93.3.5 `BError Tms::CycleParamDb::readCycleParams (BString fileName, Tms::CycleParam & param)`

Reads a set of CycleParameters from a file.

7.93.3.6 `BError Tms::CycleParamDb::setCycleParams (Tms::CycleParam param)`

Set the CycleParameters. Writes to the appropriate file name.

7.93.3.7 `BError Tms::CycleParamDb::writeCycleParams (BString fileName, Tms::CycleParam param)`

Writes a set of CycleParameters to a file.

7.93.4 Member Data Documentation

7.93.4.1 `BString Tms::CycleParamDb::obaseDir` [private]

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

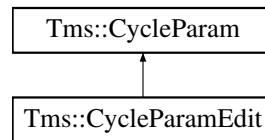
- [TmsLib.h](#)
- [TmsLib.cc](#)

7.94 Tms::CycleParamEdit Class Reference

Cycle Parameter management class.

```
#include <TmsCycleParam.h>
```

Inheritance diagram for Tms::CycleParamEdit:



Public Member Functions

- [CycleParamEdit](#) ()
- [CycleParamEdit](#) (const [CycleParam](#) ¶m)
- void [clear](#) ()
- [BString](#) [getString](#) ()
 - Gets the Cycle Parameters in a string format for writing to a file or display.*
- [BError](#) [setString](#) ([BString](#) str)
 - Sets the Cycle Parameters from a string. For reading from a file.*
- [BError](#) [readFromFile](#) ([BString](#) fileName)
 - Reads the Cycle Parameters from a file.*
- [BError](#) [writeToFile](#) ([BString](#) fileName)
 - Writes the Cycle Parameters to a file.*
- [BError](#) [setStates](#) ([BList](#)< [CycleParamState](#) > cycleStates)
 - Sets the Cycle Parameter states given the state information list.*
- [BError](#) [getStates](#) ([BList](#)< [CycleParamState](#) > &cycleStates)
 - Returns the state information list describing the Cycle Parameter states. These may not be present.*

Static Public Member Functions

- static void [getDefaultState](#) ([CycleParamState](#) &state)
 - Get the default settings for a state.*
- static void [getdefaultPickupPositions](#) ([BArray](#)< [BInt32](#) > &pos)
 - Get the default pickup positions.*

Private Member Functions

- double [value](#) (int numSamples, int harmonic, double phase, int sample)
- int [bunch](#) (int numSamples, int harmonic, double phase, int sample)
- [BError](#) [generateState](#) ([CycleParamState](#) stateParam)

Additional Inherited Members

7.94.1 Detailed Description

Cycle Parameter management class.

7.94.2 Constructor & Destructor Documentation

7.94.2.1 Tms::CycleParamEdit::CycleParamEdit ()

7.94.2.2 Tms::CycleParamEdit::CycleParamEdit (const CycleParam & param)

7.94.3 Member Function Documentation

7.94.3.1 int Tms::CycleParamEdit::bunch (int numSamples, int harmonic, double phase, int sample) [private]

7.94.3.2 void Tms::CycleParamEdit::clear ()

7.94.3.3 BError Tms::CycleParamEdit::generateState (CycleParamState stateParam) [private]

This function will generate the phase tables for a given state. It is passed the parameters for the LO1 reference and the LO2 reference. If lo?Harmonic is 1, then FREF is generated.

7.94.3.4 void Tms::CycleParamEdit::getDefaultPickupPositions (BArray< BInt32 > & pos) [static]

Get the default pickup positions.

Calculates the base pickup phase values for the PS ring.

7.94.3.5 void Tms::CycleParamEdit::getDefaultState (CycleParamState & state) [static]

Get the default settings for a state.

7.94.3.6 BError Tms::CycleParamEdit::getStates (BList< CycleParamState > & cycleStates)

Returns the state information list describing the Cycle Parameter states. These may not be present.

7.94.3.7 BString Tms::CycleParamEdit::getString ()

Gets the Cycle Parameters in a string format for writing to a file or display.

7.94.3.8 BError Tms::CycleParamEdit::readFromFile (BString fileName)

Reads the Cycle Parameters from a file.

7.94.3.9 BError Tms::CycleParamEdit::setStates (BList< CycleParamState > cycleStates)

Sets the Cycle Parameter states given the state information list.

7.94.3.10 BError Tms::CycleParamEdit::setString (BString str)

Sets the Cycle Parameters from a string. For reading from a file.

7.94.3.11 double Tms::CycleParamEdit::value (int numSamples, int harmonic, double phase, int sample) [private]

7.94.3.12 BError Tms::CycleParamEdit::writeToFile (BString fileName)

Writes the Cycle Parameters to a file.

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

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

7.95 Tms::CycleParamItem Class Reference

```
#include <TmsD.h>
```

Public Member Functions

- [CycleParamItem](#) ([BString](#) pcycleType=[BString](#)(), [BUInt32](#) pring=[BUInt32](#)(), [BUInt32](#) pchannel=[BUInt32](#)())

Public Attributes

- [BString](#) cycleType
The Cycle Type Name of this parameter set, normally the BEAM type the set of parameters is designed to measure.
- [BUInt32](#) ring
The ring number this configuration is for, 0 defines all rings.
- [BUInt32](#) channel
The channel number this configuration is for, 0 defines all channels.

7.95.1 Constructor & Destructor Documentation

7.95.1.1 [Tms::CycleParamItem::CycleParamItem](#) ([BString](#) pcycleType = [BString](#) () , [BUInt32](#) pring = [BUInt32](#) () , [BUInt32](#) pchannel = [BUInt32](#) ())

7.95.2 Member Data Documentation

7.95.2.1 [BUInt32](#) [Tms::CycleParamItem::channel](#)

The channel number this configuration is for, 0 defines all channels.

7.95.2.2 [BString](#) [Tms::CycleParamItem::cycleType](#)

The Cycle Type Name of this parameter set, normally the BEAM type the set of parameters is designed to measure.

7.95.2.3 [BUInt32](#) [Tms::CycleParamItem::ring](#)

The ring number this configuration is for, 0 defines all rings.

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

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

7.96 Tms::CycleParamState Class Reference

```
#include <TmsCycleParam.h>
```

Public Member Functions

- [CycleParamState](#) ()
- void [clear](#) ()
 - Clear the entry.*
- void [setNext](#) (int nextNum, [BUInt32](#) nextPeriod, bool f1RefSigma, bool f1LoMsb, bool f2RefSigma, bool f2LoMsb, bool plIF2, bool acquire)
- [BString](#) [getString](#) ()
 - Returns the [CycleParamState](#) in string form.*
- [BError](#) [setString](#) ([BString](#) str)
 - Sets the [CycleParamState](#) from a string.*

Public Attributes

- [BUInt32](#) [num](#)
 - The state number.*
- [BUInt32](#) [period](#)
 - The cycle period.*
- [TmsState](#) [state](#)
 - The TMS State.*
- [BUInt32](#) [bunchMask](#)
 - The set of bunches to capture bit mask.*
- [BUInt32](#) [mean1Mask](#)
 - The set of bunches to pass through meanFilter1.*
- [BUInt32](#) [mean2Mask](#)
 - The set of bunches to pass through meanFilter2.*
- [BUInt32](#) [lo1Harmonic](#)
 - The LO harmonic number used in this state.*
- double [lo1Phase](#)
 - The phase offset of the LO as a fraction of FREF (+-1.0)*
- [BUInt32](#) [lo2Harmonic](#)
 - The LO harmonic number used in this state.*
- double [lo2Phase](#)
 - The phase offset of the LO as a fraction of FREF (+-1.0)*
- double [gateWidth](#)
 - The gate pulse width as a fraction of LO (0 - 1.0)*
- double [gatePhase](#)
 - The gate phase offset as a fraction of LO (0 - 1.0)*
- double [blrWidth](#)
 - The gate pulse width as a fraction of LO (0 - 1.0)*
- double [blrPhase](#)
 - The gate phase offset as a fraction of LO (0 - 1.0)*

7.96.1 Constructor & Destructor Documentation

7.96.1.1 Tms::CycleParamState::CycleParamState ()

7.96.2 Member Function Documentation

7.96.2.1 void Tms::CycleParamState::clear ()

Clear the entry.

7.96.2.2 BString Tms::CycleParamState::getString ()

Returns the [CycleParamState](#) in string form.

7.96.2.3 void Tms::CycleParamState::setNext (int *nextNum*, BUInt32 *nextPeriod*, bool *f1RefSigma*, bool *f1LoMsb*, bool *f2RefSigma*, bool *f2LoMsb*, bool *plIF2*, bool *acquire*)

7.96.2.4 BError Tms::CycleParamState::setString (BString *str*)

Sets the [CycleParamState](#) from a string.

7.96.3 Member Data Documentation

7.96.3.1 double Tms::CycleParamState::blrPhase

The gate phase offset as a fraction of LO (0 - 1.0)

7.96.3.2 double Tms::CycleParamState::blrWidth

The gate pulse width as a fraction of LO (0 - 1.0)

7.96.3.3 BUInt32 Tms::CycleParamState::bunchMask

The set of bunches to capture bit mask.

7.96.3.4 double Tms::CycleParamState::gatePhase

The gate phase offset as a fraction of LO (0 - 1.0)

7.96.3.5 double Tms::CycleParamState::gateWidth

The gate pulse width as a fraction of LO (0 - 1.0)

7.96.3.6 BUInt32 Tms::CycleParamState::lo1Harmonic

The LO harmonic number used in this state.

7.96.3.7 double Tms::CycleParamState::lo1Phase

The phase offset of the LO as a fraction of FREF (+/-1.0)

7.96.3.8 BUInt32 Tms::CycleParamState::lo2Harmonic

The LO harmonic number used in this state.

7.96.3.9 double Tms::CycleParamState::lo2Phase

The phase offset of the LO as a fraction of FREF (+/-1.0)

7.96.3.10 BUInt32 Tms::CycleParamState::mean1Mask

The set of bunches to pass through meanFilter1.

7.96.3.11 BUInt32 Tms::CycleParamState::mean2Mask

The set of bunches to pass through meanFilter2.

7.96.3.12 BUInt32 Tms::CycleParamState::num

The state number.

7.96.3.13 BUInt32 Tms::CycleParamState::period

The cycle period.

7.96.3.14 TmsState Tms::CycleParamState::state

The TMS State.

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

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

7.97 Tms::CycleTypeInfo Class Reference

```
#include <TmsD.h>
```

Public Member Functions

- [CycleTypeInfo](#) ([BString](#) pcycleType=[BString](#)(), [BString](#) pinfo=[BString](#)(), [BList](#)< [CycleTypeInfo-Period](#) > pperiods=[BList](#)< [CycleTypeInfoPeriod](#) >())

Public Attributes

- [BString](#) cycleType
The Cycle Type Name.
- [BString](#) info
Information string on this cycle type.
- [BList](#)< [CycleTypeInfoPeriod](#) > periods
The list of cycle periods.

7.97.1 Constructor & Destructor Documentation

7.97.1.1 Tms::CycleTypeInfo::CycleTypeInfo ([BString](#) pcycleType = [BString](#) (), [BString](#) pinfo = [BString](#) (), [BList](#)< [CycleTypeInfoPeriod](#) > pperiods = [BList](#)<[CycleTypeInfoPeriod](#) > ())

7.97.2 Member Data Documentation

7.97.2.1 BString Tms::CycleTypeInfo::cycleType

The Cycle Type Name.

7.97.2.2 BString Tms::CycleTypeInfo::info

Information string on this cycle type.

7.97.2.3 BList<CycleTypeInfoPeriod > Tms::CycleTypeInfo::periods

The list of cycle periods.

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

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

7.98 Tms::CycleTypeInfoPeriod Class Reference

Cycle Type information.

```
#include <TmsD.h>
```

Public Member Functions

- [CycleTypeInfoPeriod](#) ([BUInt32](#) pcyclePeriod=[BUInt32](#)(), [BUInt32](#) pharmonic=[BUInt32](#)(), [BUInt32](#) pnumBunches=[BUInt32](#)(), [BUInt32](#) pbunchMask=[BUInt32](#)())

Public Attributes

- [BUInt32](#) cyclePeriod
The Cycle Period.
- [BUInt32](#) harmonic
The Machines harmonic number.
- [BUInt32](#) numBunches
The number of bunches.
- [BUInt32](#) bunchMask
Bitmask defining which buckets the bunches are captured from. Bit 0 is bucket 1, bit 1 is bucket 2 etc.

7.98.1 Detailed Description

Cycle Type information.

7.98.2 Constructor & Destructor Documentation

- 7.98.2.1 `Tms::CycleTypeInfoPeriod::CycleTypeInfoPeriod (BUInt32 pcyclePeriod = BUInt32 () , BUInt32 pharmonic = BUInt32 () , BUInt32 pnumBunches = BUInt32 () , BUInt32 pbunchMask = BUInt32 ())`

7.98.3 Member Data Documentation

7.98.3.1 BUInt32 Tms::CycleTypeInfoPeriod::bunchMask

Bitmask defining which buckets the bunches are captured from. Bit 0 is bucket 1, bit 1 is bucket 2 etc.

7.98.3.2 BUInt32 Tms::CycleTypeInfoPeriod::cyclePeriod

The Cycle Period.

7.98.3.3 BUInt32 Tms::CycleTypeInfoPeriod::harmonic

The Machines harmonic number.

7.98.3.4 BUInt32 Tms::CycleTypeInfoPeriod::numBunches

The number of bunches.

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

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

7.99 Tms::Data Class Reference

This class stores the raw data.

```
#include <TmsD.h>
```

Public Member Functions

- [Data](#) ([BUInt32](#) pnumValues=[BUInt32](#)(), [BUInt32](#) pDataType=[BUInt32](#)(), [BUInt32](#) pnumBunches=[BUInt32](#)(), [BUInt32](#) pnumChannels=[BUInt32](#)(), [BArray](#)< [DataValue](#) > pDataValues=[BArray](#)< [DataValue](#) >(), [BArray](#)< [BError](#) > perrors=[BArray](#)< [BError](#) >())

Public Attributes

- [BUInt32](#) numValues
The total number of data samples.
- [BUInt32](#) dataType
The type of data in the data block.
- [BUInt32](#) numBunches
The number of bunches.
- [BUInt32](#) numChannels
The number of channels.
- [BArray](#)< [DataValue](#) > dataValues
The data.
- [BArray](#)< [BError](#) > errors
Individual errors for each channel within dataValues.

7.99.1 Detailed Description

This class stores the raw data.

7.99.2 Constructor & Destructor Documentation

7.99.2.1 `Tms::Data::Data (BUInt32 pnumValues = BUInt32 (), BUInt32 pdataType = BUInt32 (), BUInt32 pnumBunches = BUInt32 (), BUInt32 pnumChannels = BUInt32 (), BArray< DataValue > pdataValues = BArray<DataValue > (), BArray< BError > perrors = BArray<BError > ())`

7.99.3 Member Data Documentation

7.99.3.1 `BUInt32 Tms::Data::dataType`

The type of data in the data block.

7.99.3.2 `BArray<DataValue > Tms::Data::dataValues`

The data.

7.99.3.3 `BArray<BError > Tms::Data::errors`

Individual errors for each channel within dataValues.

7.99.3.4 `BUInt32 Tms::Data::numBunches`

The number of bunches.

7.99.3.5 `BUInt32 Tms::Data::numChannels`

The number of channels.

7.99.3.6 `BUInt32 Tms::Data::numValues`

The total number of data samples.

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

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

7.100 Tms::DataInfo Class Reference

This class defines the data to be acquired and/or fetched.

```
#include <TmsD.h>
```

Public Member Functions

- `DataInfo (BUInt32 pcycleNumber=BUInt32(), BUInt32 pchannel=BUInt32(), BUInt32 pcyclePeriod=BUInt32(), BUInt32 pstartTime=BUInt32(), BUInt32 porbitNumber=BUInt32(), BUInt32 pbunchNumber=BUInt32(), BUInt32 pfunction=BUInt32(), BUInt32 pargument=BUInt32(), BUInt32 pnumValues=BUInt32(), BInt32 pbeyondPeriod=BInt32())`

Public Attributes

- [BUInt32 cycleNumber](#)
The PS Cycle number.
- [BUInt32 channel](#)
The pick-up channel number.
- [BUInt32 cyclePeriod](#)
The cycle period the data is from.
- [BUInt32 startTime](#)
The start time in milli-seconds in the cycle period (starting from 0)
- [BUInt32 orbitNumber](#)
The starting orbit number (starting from 0)
- [BUInt32 bunchNumber](#)
The Bunch number (starting from 1 (0 is all bunches))
- [BUInt32 function](#)
The data processing function to perform or performed. (0 normal data)
- [BUInt32 argument](#)
The Argument to the data processing function.
- [BUInt32 numValues](#)
The total number of data points to return.
- [BInt32 beyondPeriod](#)
If set allows reads of data beyond the end of the period.

7.100.1 Detailed Description

This class defines the data to be acquired and/or fetched.

7.100.2 Constructor & Destructor Documentation

7.100.2.1 `Tms::DataInfo::DataInfo (BUInt32 pcycleNumber = BUInt32 (), BUInt32 pchannel = BUInt32 (), BUInt32 pcyclePeriod = BUInt32 (), BUInt32 pstartTime = BUInt32 (), BUInt32 porbitNumber = BUInt32 (), BUInt32 pbunchNumber = BUInt32 (), BUInt32 pfunction = BUInt32 (), BUInt32 pargument = BUInt32 (), BUInt32 pnumValues = BUInt32 (), BInt32 pbeyondPeriod = BInt32 ())`

7.100.3 Member Data Documentation

7.100.3.1 BUInt32 Tms::DataInfo::argument

The Argument to the data processing function.

7.100.3.2 BInt32 Tms::DataInfo::beyondPeriod

If set allows reads of data beyond the end of the period.

7.100.3.3 BUInt32 Tms::DataInfo::bunchNumber

The Bunch number (starting from 1 (0 is all bunches))

7.100.3.4 BUInt32 Tms::DataInfo::channel

The pick-up channel number.

7.100.3.5 BUInt32 Tms::DataInfo::cycleNumber

The PS Cycle number.

7.100.3.6 BUInt32 Tms::DataInfo::cyclePeriod

The cycle period the data is from.

7.100.3.7 BUInt32 Tms::DataInfo::function

The data processing function to perform or performed. (0 normal data)

7.100.3.8 BUInt32 Tms::DataInfo::numValues

The total number of data points to return.

7.100.3.9 BUInt32 Tms::DataInfo::orbitNumber

The starting orbit number (starting from 0)

7.100.3.10 BUInt32 Tms::DataInfo::startTime

The start time in milli-seconds in the cycle period (starting from 0)

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

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

7.101 Tms::DataValue Class Reference

This is the definition of a single data value.

```
#include <TmsD.h>
```

Public Member Functions

- [DataValue](#) ([BInt16](#) psigma=[BInt16\(\)](#), [BInt16](#) pdeltaX=[BInt16\(\)](#), [BInt16](#) pdeltaY=[BInt16\(\)](#), [BInt16](#) ptime=[BInt16\(\)](#))

Public Attributes

- [BInt16](#) sigma
The Sigma value.
- [BInt16](#) deltaX
The DeltaX value.
- [BInt16](#) deltaY
The DeltaY value.
- [BInt16](#) time
The Time in ms this sample was processed.

7.101.1 Detailed Description

This is the definition of a single data value.

7.101.2 Constructor & Destructor Documentation

7.101.2.1 `Tms::DataValue::DataValue (BInt16 psigma = BInt16 (), BInt16 pdeltaX = BInt16 (), BInt16 pdeltaY = BInt16 (), BInt16 ptime = BInt16 ())`

7.101.3 Member Data Documentation

7.101.3.1 `BInt16 Tms::DataValue::deltaX`

The DeltaX value.

7.101.3.2 `BInt16 Tms::DataValue::deltaY`

The DeltaY value.

7.101.3.3 `BInt16 Tms::DataValue::sigma`

The Sigma value.

7.101.3.4 `BInt16 Tms::DataValue::time`

The Time in ms this sample was processed.

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

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

7.102 Tms::NameValue Class Reference

```
#include <TmsD.h>
```

Public Member Functions

- `NameValue (BString pname=BString(), BString pvalue=BString())`

Public Attributes

- `BString name`
The Name of the value.
- `BString value`
The actual value in string form.

7.102.1 Constructor & Destructor Documentation

7.102.1.1 `Tms::NameValue::NameValue (BString pname = BString (), BString pvalue = BString ())`

7.102.2 Member Data Documentation

7.102.2.1 `BString Tms::NameValue::name`

The Name of the value.

7.102.2.2 `BString Tms::NameValue::value`

The actual value in string form.

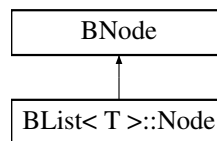
The documentation for this class was generated from the following files:

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

7.103 BList< T >::Node Class Reference

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

Inheritance diagram for BList< T >::Node:



Public Member Functions

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

Public Attributes

- [T item](#)

7.103.1 Constructor & Destructor Documentation

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

7.103.2 Member Data Documentation

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

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

- `/src/cern/tms/beam/libBeam/BList.h`

7.104 Tms::PuChannel Class Reference

This class stores a Physical Pick-Up channel id.

```
#include <TmsD.h>
```

Public Member Functions

- [PuChannel](#) ([BUInt8](#) pmoduleNum=[BUInt8](#)(), [BUInt8](#) ppupeNum=[BUInt8](#)(), [BUInt8](#) ppupeChan=[BUInt8](#)())

Public Attributes

- [BUInt8](#) moduleNum
The Module number.
- [BUInt8](#) pupeNum
The PUPE number.
- [BUInt8](#) pupeChan
The PUPE channel.

7.104.1 Detailed Description

This class stores a Physical Pick-Up channel id.

7.104.2 Constructor & Destructor Documentation

7.104.2.1 [Tms::PuChannel::PuChannel](#) ([BUInt8](#) pmoduleNum = [BUInt8](#) () , [BUInt8](#) ppupeNum = [BUInt8](#) () , [BUInt8](#) ppupeChan = [BUInt8](#) ())

7.104.3 Member Data Documentation

7.104.3.1 [BUInt8](#) [Tms::PuChannel::moduleNum](#)

The Module number.

7.104.3.2 [BUInt8](#) [Tms::PuChannel::pupeChan](#)

The PUPE channel.

7.104.3.3 [BUInt8](#) [Tms::PuChannel::pupeNum](#)

The PUPE number.

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

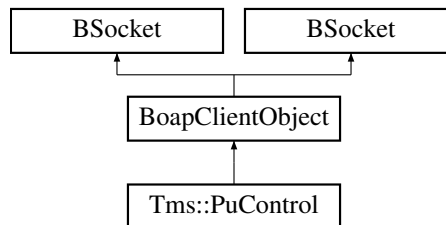
- [TmsD.h](#)
- [TmsD.cc](#)

7.105 Tms::PuControl Class Reference

This class defines the parameters for a test data capture.

```
#include <TmsC.h>
```

Inheritance diagram for Tms::PuControl:



Public Member Functions

- [PuControl](#) (BString name="")
- [BError getVersion](#) (BString &version)
Gets the software version.
- [BError init](#) ()
Initialises the system including loading all of the PUPE engines firmware. The call will return an error object indicating success or an error condition as appropriate.
- [BError setProcessPriority](#) (BUInt32 priority)
Sets the priority of the process servicing this service.
- [BError configure](#) (BUInt32 ring, [ConfigInfo](#) configInfo)
Configure the system for use. This includes mapping the individual physical Pickup channels to logical pickup channels.
- [BError setControlInfo](#) ([CycleParam](#) params)
Sets the control information for the cycle type given and subsequent cycles. The parameters for the processing cycle are passed, this includes the Phase and State table information. The call will return an error object indicating success or an error.
- [BError setNextCycle](#) (BUInt32 cycleNumber, BString cycleType)
Sets the cycle number and type for the next processing cycle. The call will return an error object indicating success or an error condition as appropriate. This should be called at least 10ms before the next CYCLE_START event.
- [BError test](#) (BList< [BError](#) > &errors)
Performs a basic test of the system returning a list of errors. The call will return an error object indicating success or an error condition as appropriate.
- [BError getStatus](#) (BList< [NameValue](#) > &statusList)
Returns the current status of the system. This information includes the number of Pick-Up's present and their individual status.
- [BError getStatistics](#) (BList< [NameValue](#) > &statsList)
Returns a list of the statistic values as name/value pairs. The call will return an error object indicating success or an error condition as appropriate.
- [BError getMasterPuChannel](#) ([PuChannel](#) &puChannel)
Returns the master PU Channel for timing.
- [BError setTestMode](#) ([PuChannel](#) puChannel, BUInt32 testOutput, BUInt32 timingDisableMask)
The signal source for the digital test output connector. 0: None, 1: FrefLocal. The timingDisableMask bit mask defines which of the timing inputs should be disabled. If a timing input is disabled it can be still operated by software command.
- [BError setTimingSignals](#) ([PuChannel](#) puChannel, BUInt32 timingSignals)
This function sets the given timing signals to the values as defined in the timingSignals bit array.
- [BError captureDiagnostics](#) ([PuChannel](#) puChannel, [TestCaptureInfo](#) captureInfo, BArray< BUInt64 > &data)

This function will capture test data.

- **BError setTestData** (**PuChannel** puChannel, **BInt32** on, **BArray**< **BUInt32** > data)

This function will set a PU channel to sample data from memory rather than the ADC's.

- **BError setPupeConfig** (**PuChannel** puPhysChannel, **PupeConfig** pupeConfig)

Sets special PUPE configuration for test purposes.

- **BError getPupeConfig** (**PuChannel** puPhysChannel, **PupeConfig** &pupeConfig)

Gets special PUPE configuration for test purposes.

Additional Inherited Members

7.105.1 Detailed Description

This class defines the parameters for a test data capture.

This class stores a Physical Pick-Up channel id This class stores the status of an individual Pick-Up This class describes the configuration of the TMS This class defines the data to be acquired and/or fetched This is the definition of a single data value This class stores the raw data This class defines the Pick-Up state table This class defines the parameters for a PS processing cycleCycle information Cycle Type information This interface provides functions to control, test and get statistics from an individual pick-up

7.105.2 Constructor & Destructor Documentation

7.105.2.1 **Tms::PuControl::PuControl** (**BString** name = " ")

7.105.3 Member Function Documentation

7.105.3.1 **BError Tms::PuControl::captureDiagnostics** (**PuChannel** puChannel, **TestCaptureInfo** captureInfo, **BArray**< **BUInt64** > & data)

This function will capture test data.

7.105.3.2 **BError Tms::PuControl::configure** (**BUInt32** ring, **ConfigInfo** configInfo)

Configure the system for use. This includes mapping the individual physical Pickup channels to logical pickup channels.

7.105.3.3 **BError Tms::PuControl::getMasterPuChannel** (**PuChannel** & puChannel)

Returns the master PU Channel for timing.

7.105.3.4 **BError Tms::PuControl::getPupeConfig** (**PuChannel** puPhysChannel, **PupeConfig** & pupeConfig)

Gets special PUPE configuration for test purposes.

7.105.3.5 **BError Tms::PuControl::getStatistics** (**BList**< **NameValue** > & statsList)

Returns a list of the statistic values as name/value pairs. The call will return an error object indicating success or an error condition as appropriate.

7.105.3.6 **BError Tms::PuControl::getStatus (BList< NameValue > & statusList)**

Returns the current status of the system. This information includes the number of Pick-Up's present and their individual status.

7.105.3.7 **BError Tms::PuControl::getVersion (BString & version)**

Gets the software version.

Parameters

<i>version</i>	A string variable filled in with the version number string.
----------------	---

7.105.3.8 **BError Tms::PuControl::init ()**

Initialises the system including loading all of the PUPE engines firmware. The call will return an error object indicating success or an error condition as appropriate.

7.105.3.9 **BError Tms::PuControl::setControllInfo (CycleParam params)**

Sets the control information for the cycle type given and subsequent cycles. The parameters for the processing cycle are passed, this includes the Phase and State table information. The call will return an error object indicating success or an error.

7.105.3.10 **BError Tms::PuControl::setNextCycle (BUInt32 cycleNumber, BString cycleType)**

Sets the cycle number and type for the next processing cycle. The call will return an error object indicating success or an error condition as appropriate. This should be called at least 10ms before the next CYCLE_START event.

7.105.3.11 **BError Tms::PuControl::setProcessPriority (BUInt32 priority)**

Sets the priority of the process servicing this service.

7.105.3.12 **BError Tms::PuControl::setPupeConfig (PuChannel puPhysChannel, PupeConfig pupeConfig)**

Sets special PUPE configuration for test purposes.

7.105.3.13 **BError Tms::PuControl::setTestData (PuChannel puChannel, BInt32 on, BArray< BUInt32 > data)**

This function will set a PU channel to sample data from memory rather than the ADC's.

7.105.3.14 **BError Tms::PuControl::setTestMode (PuChannel puChannel, BUInt32 testOutput, BUInt32 timingDisableMask)**

The signal source for the digital test output connector. 0: None, 1: FrefLocal. The timingDisableMask bit mask defines which of the timing inputs should be disabled. If a timing input is disabled it can be still operated by software command.

7.105.3.15 **BError Tms::PuControl::setTimingSignals (PuChannel puChannel, BUInt32 timingSignals)**

This function sets the given timing signals to the values as defined in the timingSignals bit array.

7.105.3.16 BError Tms::PuControl::test (BList< BError > & errors)

Performs a basic test of the system returning a list of errors. The call will return an error object indicating success or an error condition as appropriate.

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

- [TmsC.h](#)
- [TmsC.cc](#)
- [tmsFunctions.dox](#)

7.106 Tms::PupeConfig Class Reference

```
#include <TmsD.h>
```

Public Member Functions

- [PupeConfig](#) (BUInt32 pinternalTimingMask=BUInt32(), BInt32 pdoubleInjection=BInt32(), BInt32 padcSysclkSync=BInt32(), BInt32 pdisableBlr=BInt32())

Public Attributes

- [BUInt32 internalTimingMask](#)
Use internal, software/hardware generated, timing signals for the given signals.
- [BInt32 doubleInjection](#)
Simulate double injection.
- [BInt32 adcSysclkSync](#)
Sets the ADC clock to be synchronised with the SYCLK timing clock.
- [BInt32 disableBlr](#)
Disable the BLR algorithm.

7.106.1 Constructor & Destructor Documentation

7.106.1.1 Tms::PupeConfig::PupeConfig (BUInt32 pinternalTimingMask = BUInt32 () , BInt32 pdoubleInjection = BInt32 () , BInt32 padcSysclkSync = BInt32 () , BInt32 pdisableBlr = BInt32 ())

7.106.2 Member Data Documentation

7.106.2.1 BInt32 Tms::PupeConfig::adcSysclkSync

Sets the ADC clock to be synchronised with the SYCLK timing clock.

7.106.2.2 BInt32 Tms::PupeConfig::disableBlr

Disable the BLR algorithm.

7.106.2.3 BInt32 Tms::PupeConfig::doubleInjection

Simulate double injection.

7.106.2.4 BUInt32 Tms::PupeConfig::internalTimingMask

Use internal, software/hardware generated, timing signals for the given signals.

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

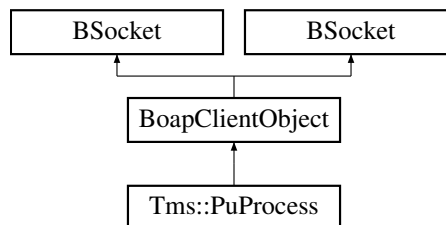
- [TmsD.h](#)
- [TmsD.cc](#)

7.107 Tms::PuProcess Class Reference

This interface provides functions to configure and capture data from individual pick-up.

```
#include <TmsC.h>
```

Inheritance diagram for Tms::PuProcess:



Public Member Functions

- [PuProcess](#) (BString name="")
- [BError getVersion](#) (BString &version)

Gets the software version.
- [BError getCycleInformation](#) (BUInt32 cycleNumber, [CycleInformation](#) &cycleInformation)

Gets information on given cycle number.
- [BError getStatus](#) (PuChannel puChannel, [PuStatus](#) &puStatus)
- [BError getData](#) (PuChannel puChannel, [DataInfo](#) dataInfo, [Data](#) &data, BUInt32 &orbitNumber)

This function returns a set of data from the data present in the data cache or directly from the Pick-Up processing engines. The [DataInfo](#) object describes the data required. The call will return the required data along with an error object indicating success or an error condition as appropriate. The call will block until data is ready.
- [BError addEventServer](#) (BString name)

Adds an event server.
- [BError requestData](#) (PuChannel puChannel, [DataInfo](#) dataInfo)

This adds a request for some data. The [DataInfo](#) object defines the data required. This request can be made at any time. If the data is present in cache the data will be available immediately, if not the system will await the data from a subsequent processing cycle. When the data is available a "data" event will be sent to the client. Not that it is not necessary to use requestData. The client can call [getData\(\)](#) directly although this call will block until the data is actually ready.

Additional Inherited Members

7.107.1 Detailed Description

This interface provides functions to configure and capture data from individual pick-up.

7.107.2 Constructor & Destructor Documentation

7.107.2.1 Tms::PuProcess::PuProcess (BString name = " ")

7.107.3 Member Function Documentation

7.107.3.1 BError Tms::PuProcess::addEventServer (BString name)

Adds an event server.

7.107.3.2 BError Tms::PuProcess::getCycleInformation (BUInt32 cycleNumber, CycleInformation & cycleInformation)

Gets information on given cycle number.

7.107.3.3 BError Tms::PuProcess::getData (PuChannel puChannel, DataInfo dataInfo, Data & data, BUInt32 & orbitNumber)

This function returns a set of data from the data present in the data cache or directly from the Pick-Up processing engines. The [DataInfo](#) object describes the data required. The call will return the required data along with an error object indicating success or an error condition as appropriate. The call will block until data is ready.

7.107.3.4 BError Tms::PuProcess::getStatus (PuChannel puChannel, PuStatus & puStatus)

7.107.3.5 BError Tms::PuProcess::getVersion (BString & version)

Gets the software version.

7.107.3.6 BError Tms::PuProcess::requestData (PuChannel puChannel, DataInfo dataInfo)

This adds a request for some data. The [DataInfo](#) object defines the data required. This request can be made at any time. If the data is present in cache the data will be available immediately, if not the system will await the data from a subsequent processing cycle. When the data is available a "data" event will be sent to the client. Not that it is not necessary to use requestData. The client can call [getData\(\)](#) directly although this call will block until the data is actually ready.

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

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

7.108 Tms::PuStateTable Class Reference

This class defines the Pick-Up state table.

```
#include <TmsD.h>
```

Public Member Functions

- [PuStateTable](#) (BUInt32 pperiod=BUInt32(), BUInt32 pstate=BUInt32(), BUInt32 pharmonic=BUInt32(), BUInt32 pnumBunches=BUInt32(), BUInt32 pbunchMask=BUInt32(), BArray< BUInt8 > pphaseTable=BArray< BUInt8 >())

Public Attributes

- [BUInt32 period](#)
The Cycle period this state is used for.
- [BUInt32 state](#)
The State table entry.
- [BUInt32 harmonic](#)
The harmonic number for this state.
- [BUInt32 numBunches](#)
The number of bunches to capture.
- [BUInt32 bunchMask](#)
Bitmask defining which buckets the bunches are captured from. Bit 0 is bucket 1, bit 1 is bucket 2 etc.
- [BArray< BUInt8 > phaseTable](#)
The Phase table for this state.

7.108.1 Detailed Description

This class defines the Pick-Up state table.

7.108.2 Constructor & Destructor Documentation

7.108.2.1 `Tms::PuStateTable::PuStateTable (BUInt32 pperiod = BUInt32 (), BUInt32 pstate = BUInt32 (), BUInt32 pharmonic = BUInt32 (), BUInt32 pnumBunches = BUInt32 (), BUInt32 pbunchMask = BUInt32 (), BArray< BUInt8 > pphaseTable = BArray< BUInt8 > ())`

7.108.3 Member Data Documentation

7.108.3.1 BUInt32 Tms::PuStateTable::bunchMask

Bitmask defining which buckets the bunches are captured from. Bit 0 is bucket 1, bit 1 is bucket 2 etc.

7.108.3.2 BUInt32 Tms::PuStateTable::harmonic

The harmonic number for this state.

7.108.3.3 BUInt32 Tms::PuStateTable::numBunches

The number of bunches to capture.

7.108.3.4 BUInt32 Tms::PuStateTable::period

The Cycle period this state is used for.

7.108.3.5 BArray< BUInt8 > Tms::PuStateTable::phaseTable

The Phase table for this state.

7.108.3.6 BUInt32 Tms::PuStateTable::state

The State table entry.

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

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

7.109 Tms::PuStatus Class Reference

This class stores the status of an individual Pick-Up.

```
#include <TmsD.h>
```

Public Member Functions

- [PuStatus](#) ([BUInt32](#) pruning=[BUInt32](#)(), [BError](#) perror=[BError](#)())

Public Attributes

- [BUInt32](#) running
The Pick-Up is currently running.
- [BError](#) error
The Pick-Up's current error status.

7.109.1 Detailed Description

This class stores the status of an individual Pick-Up.

7.109.2 Constructor & Destructor Documentation

7.109.2.1 [Tms::PuStatus::PuStatus](#) ([BUInt32](#) pruning = [BUInt32](#) () , [BError](#) perror = [BError](#) ())

7.109.3 Member Data Documentation

7.109.3.1 [BError](#) [Tms::PuStatus::error](#)

The Pick-Up's current error status.

7.109.3.2 [BUInt32](#) [Tms::PuStatus::running](#)

The Pick-Up is currently running.

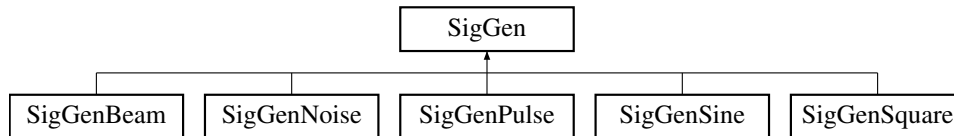
The documentation for this class was generated from the following files:

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

7.110 SigGen Class Reference

```
#include <SigGen.h>
```

Inheritance diagram for SigGen:



Public Member Functions

- [SigGen](#) ()
- virtual [~SigGen](#) ()
- [BError config](#) (double *sampleRate*)
- virtual [BError generate](#) ([Sample](#) *data, int numSamples)

Protected Attributes

- double [osampleRate](#)
- unsigned long long [ox](#)

7.110.1 Constructor & Destructor Documentation

7.110.1.1 [SigGen::SigGen](#) ()

7.110.1.2 [SigGen::~~SigGen](#) () [[virtual](#)]

7.110.2 Member Function Documentation

7.110.2.1 [BError SigGen::config](#) (double *sampleRate*)

7.110.2.2 [BError SigGen::generate](#) ([Sample](#) * *data*, int *numSamples*) [[virtual](#)]

Reimplemented in [SigGenBeam](#), [SigGenPulse](#), [SigGenNoise](#), [SigGenSquare](#), and [SigGenSine](#).

7.110.3 Member Data Documentation

7.110.3.1 double [SigGen::osampleRate](#) [[protected](#)]

7.110.3.2 unsigned long long [SigGen::ox](#) [[protected](#)]

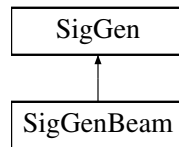
The documentation for this class was generated from the following files:

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

7.111 SigGenBeam Class Reference

```
#include <SigGen.h>
```

Inheritance diagram for SigGenBeam:



Public Member Functions

- [SigGenBeam](#) ()
- virtual [~SigGenBeam](#) ()
- [BError config](#) (double *sampleRate*, double *fref*, int *harmonic*, int *bunchSet*, double *reduce*, int *blr*, double *amplitude*)
- [BError generate](#) (Sample **data*, int *numSamples*)
- [BError generateIntegrated](#) (Sample **data*, int *numSamples*)

Public Attributes

- int [oharmonic](#)
- int [obunchSet](#)
- double [oreduce](#)
- int [oblr](#)
- double [oamplitude](#)
- double [ofref](#)

Additional Inherited Members

7.111.1 Constructor & Destructor Documentation

7.111.1.1 [SigGenBeam::SigGenBeam](#) ()

7.111.1.2 [SigGenBeam::~~SigGenBeam](#) () [virtual]

7.111.2 Member Function Documentation

7.111.2.1 [BError SigGenBeam::config](#) (double *sampleRate*, double *fref*, int *harmonic*, int *bunchSet*, double *reduce*, int *blr*, double *amplitude*)

7.111.2.2 [BError SigGenBeam::generate](#) (Sample * *data*, int *numSamples*) [virtual]

Reimplemented from [SigGen](#).

7.111.2.3 [BError SigGenBeam::generateIntegrated](#) (Sample * *data*, int *numSamples*)

7.111.3 Member Data Documentation

7.111.3.1 double [SigGenBeam::oamplitude](#)

7.111.3.2 int SigGenBeam::oblr

7.111.3.3 int SigGenBeam::obunchSet

7.111.3.4 double SigGenBeam::ofref

7.111.3.5 int SigGenBeam::oharmonic

7.111.3.6 double SigGenBeam::oreduce

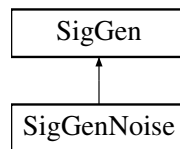
The documentation for this class was generated from the following files:

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

7.112 SigGenNoise Class Reference

```
#include <SigGen.h>
```

Inheritance diagram for SigGenNoise:



Public Member Functions

- [SigGenNoise \(\)](#)
- virtual [~SigGenNoise \(\)](#)
- [BError config](#) (double *sampleRate*, double *amplitude*=1.0)
- [BError generate](#) ([Sample](#) **data*, int *numSamples*)

Public Attributes

- double [oamplitude](#)

Additional Inherited Members

7.112.1 Constructor & Destructor Documentation

7.112.1.1 [SigGenNoise::SigGenNoise \(\)](#)

7.112.1.2 [SigGenNoise::~~SigGenNoise \(\)](#) [virtual]

7.112.2 Member Function Documentation

7.112.2.1 [BError SigGenNoise::config \(double *sampleRate*, double *amplitude* = 1 . 0 \)](#)

7.112.2.2 [BError SigGenNoise::generate \(\[Sample\]\(#\) * *data*, int *numSamples* \)](#) [virtual]

Reimplemented from [SigGen](#).

7.112.3 Member Data Documentation

7.112.3.1 double SigGenNoise::oamplitude

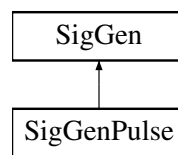
The documentation for this class was generated from the following files:

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

7.113 SigGenPulse Class Reference

```
#include <SigGen.h>
```

Inheritance diagram for SigGenPulse:



Public Member Functions

- [SigGenPulse](#) ()
- virtual [~SigGenPulse](#) ()
- [BError config](#) (double *sampleRate*, double *freq*, double *amplitude*, double *onTime*, double *startTime=0.0*)
- [BError generate](#) ([Sample](#) **data*, int *numSamples*)

Public Attributes

- double [ofreq](#)
- double [oamplitude](#)
- double [oonTime](#)
- double [ostartTime](#)

Additional Inherited Members

7.113.1 Constructor & Destructor Documentation

7.113.1.1 SigGenPulse::SigGenPulse ()

7.113.1.2 SigGenPulse::~~SigGenPulse () [virtual]

7.113.2 Member Function Documentation

7.113.2.1 BError SigGenPulse::config (double *sampleRate*, double *freq*, double *amplitude*, double *onTime*, double *startTime = 0.0*)

7.113.2.2 BError SigGenPulse::generate ([Sample](#) * *data*, int *numSamples*) [virtual]

Reimplemented from [SigGen](#).

7.113.3 Member Data Documentation

7.113.3.1 `double SigGenPulse::oamplitude`

7.113.3.2 `double SigGenPulse::ofreq`

7.113.3.3 `double SigGenPulse::oonTime`

7.113.3.4 `double SigGenPulse::ostartTime`

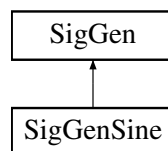
The documentation for this class was generated from the following files:

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

7.114 SigGenSine Class Reference

```
#include <SigGen.h>
```

Inheritance diagram for SigGenSine:



Public Member Functions

- [SigGenSine \(\)](#)
- `virtual ~SigGenSine ()`
- `BError config (double sampleRate, double freq, double amplitude=1.0)`
- `BError generate (Sample *data, int numSamples)`

Public Attributes

- `double ofreq`
- `double oamplitude`

Additional Inherited Members

7.114.1 Constructor & Destructor Documentation

7.114.1.1 `SigGenSine::SigGenSine ()`

7.114.1.2 `SigGenSine::~~SigGenSine ()` `[virtual]`

7.114.2 Member Function Documentation

7.114.2.1 `BError SigGenSine::config (double sampleRate, double freq, double amplitude = 1 . 0)`

7.114.2.2 **BError** SigGenSine::generate (**Sample** * *data*, int *numSamples*) [virtual]

Reimplemented from [SigGen](#).

7.114.3 Member Data Documentation

7.114.3.1 double SigGenSine::oamplitude

7.114.3.2 double SigGenSine::ofreq

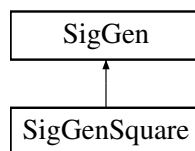
The documentation for this class was generated from the following files:

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

7.115 SigGenSquare Class Reference

```
#include <SigGen.h>
```

Inheritance diagram for SigGenSquare:



Public Member Functions

- [SigGenSquare](#) ()
- virtual [~SigGenSquare](#) ()
- **BError** [config](#) (double *sampleRate*, double *freq*, double *amplitude*=1.0, double *offset*=0.0)
- **BError** [generate](#) (**Sample** **data*, int *numSamples*)

Public Attributes

- double [ofreq](#)
- double [oamplitude](#)
- double [offset](#)

Additional Inherited Members

7.115.1 Constructor & Destructor Documentation

7.115.1.1 **SigGenSquare**::**SigGenSquare** ()

7.115.1.2 **SigGenSquare**::**~SigGenSquare** () [virtual]

7.115.2 Member Function Documentation

7.115.2.1 **BError** **SigGenSquare**::**config** (double *sampleRate*, double *freq*, double *amplitude* = 1.0, double *offset* = 0.0)

7.115.2.2 `BError SigGenSquare::generate (Sample * data, int numSamples)` [virtual]

Reimplemented from [SigGen](#).

7.115.3 Member Data Documentation

7.115.3.1 `double SigGenSquare::oamplitude`

7.115.3.2 `double SigGenSquare::ofreq`

7.115.3.3 `double SigGenSquare::offset`

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

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

7.116 Tms::Simulation Class Reference

```
#include <TmsD.h>
```

Public Member Functions

- [Simulation](#) ([BInt32](#) ptiming=[BInt32](#)(), [BInt32](#) pdoubleInjection=[BInt32](#)(), [BInt32](#) pdata=[BInt32](#)(), [BInt32](#) psetNextCycle=[BInt32](#)(), [BString](#) pcycleType=[BString](#)())

Public Attributes

- [BInt32](#) timing
Simulate timing signals.
- [BInt32](#) doubleInjection
Simulate double injection.
- [BInt32](#) data
Simulate FREF and Analogue Sigma, DeltaX and DeltaY data.
- [BInt32](#) setNextCycle
Simulate setNextCycle.
- [BString](#) cycleType
The cycle type to set.

7.116.1 Constructor & Destructor Documentation

7.116.1.1 `Tms::Simulation::Simulation (BInt32 ptiming = BInt32 (), BInt32 pdoubleInjection = BInt32 (), BInt32 pdata = BInt32 (), BInt32 psetNextCycle = BInt32 (), BString pcycleType = BString ())`

7.116.2 Member Data Documentation

7.116.2.1 `BString Tms::Simulation::cycleType`

The cycle type to set.

7.116.2.2 BInt32 Tms::Simulation::data

Simulate FREF and Analogue Sigma, DeltaX and DeltaY data.

7.116.2.3 BInt32 Tms::Simulation::doubleInjection

Simulate double injection.

7.116.2.4 BInt32 Tms::Simulation::setNextCycle

Simulate setNextCycle.

7.116.2.5 BInt32 Tms::Simulation::timing

Simulate timing signals.

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

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

7.117 Tms::TestCaptureInfo Class Reference

This class defines the parameters for a test data capture.

```
#include <TmsD.h>
```

Public Member Functions

- [TestCaptureInfo](#) (BUInt32 psource=BUInt32(), BUInt32 pclock=BUInt32(), BUInt32 pstartTime=BUInt32(), BUInt32 ppostTriggerDelay=BUInt32(), BUInt32 ptriggerMask=BUInt32(), BInt32 ptriggerAnd=BInt32(), BInt32 ptriggerStore=BInt32(), BInt32 ptriggerSourceData=BInt32(), BInt32 ptriggerStateEnable=BInt32(), BUInt32 ptriggerState=BUInt32())

Public Attributes

- [BUInt32 source](#)
The source data (0 - 3)
- [BUInt32 clock](#)
The Clock source.
- [BUInt32 startTime](#)
The start time in ms from CYCLE_START before trigger is activated.
- [BUInt32 postTriggerDelay](#)
The delay, in clock cycles, after the trigger before capture starts.
- [BUInt32 triggerMask](#)
The Trigger bit mask. This is the bit mask of the 8 timing signals.
- [BInt32 triggerAnd](#)
The Trigger function is an AND rather than an OR.
- [BInt32 triggerStore](#)
Store the trigger in the upper 8 data bits.
- [BInt32 triggerSourceData](#)

Use lower 32bits of data as trigger source rather than timing signals.

- [BInt32 triggerStateEnable](#)

Trigger in state enable.

- [BUInt32 triggerState](#)

Trigger state number.

7.117.1 Detailed Description

This class defines the parameters for a test data capture.

7.117.2 Constructor & Destructor Documentation

7.117.2.1 `Tms::TestCaptureInfo::TestCaptureInfo (BUInt32 psource = BUInt32 (), BUInt32 pclock = BUInt32 (), BUInt32 pstartTime = BUInt32 (), BUInt32 ppostTriggerDelay = BUInt32 (), BUInt32 ptriggerMask = BUInt32 (), BInt32 ptriggerAnd = BInt32 (), BInt32 ptriggerStore = BInt32 (), BInt32 ptriggerSourceData = BInt32 (), BInt32 ptriggerStateEnable = BInt32 (), BUInt32 ptriggerState = BUInt32 ())`

7.117.3 Member Data Documentation

7.117.3.1 BUInt32 Tms::TestCaptureInfo::clock

The Clock source.

7.117.3.2 BUInt32 Tms::TestCaptureInfo::postTriggerDelay

The delay, in clock cycles, after the trigger before capture starts.

7.117.3.3 BUInt32 Tms::TestCaptureInfo::source

The source data (0 - 3)

7.117.3.4 BUInt32 Tms::TestCaptureInfo::startTime

The start time in ms from CYCLE_START before trigger is activated.

7.117.3.5 BInt32 Tms::TestCaptureInfo::triggerAnd

The Trigger function is an AND rather than an OR.

7.117.3.6 BUInt32 Tms::TestCaptureInfo::triggerMask

The Trigger bit mask. This is the bit mask of the 8 timing signals.

7.117.3.7 BInt32 Tms::TestCaptureInfo::triggerSourceData

Use lower 32bits of data as trigger source rather than timing signals.

7.117.3.8 BUInt32 Tms::TestCaptureInfo::triggerState

Trigger state number.

7.117.3.9 BInt32 Tms::TestCaptureInfo::triggerStateEnable

Trigger in state enable.

7.117.3.10 BInt32 Tms::TestCaptureInfo::triggerStore

Store the trigger in the upper 8 data bits.

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

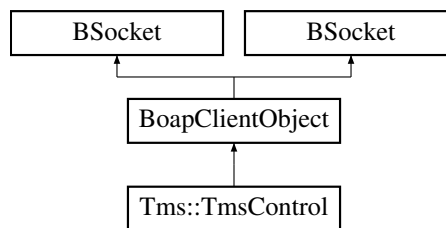
- [TmsD.h](#)
- [TmsD.cc](#)

7.118 Tms::TmsControl Class Reference

This interface provides functions to control, test and get statistics from the TMS as a whole.

```
#include <TmsC.h>
```

Inheritance diagram for Tms::TmsControl:



Public Member Functions

- [TmsControl](#) (BString name="")
- [BError getVersion](#) (BString &version)
Gets the software version.
- [BError setProcessPriority](#) (BUInt32 priority)
Sets the priority of the process servicing this service.
- [BError init](#) ()
Initialises the system including resetting all of the PUPE engines firmware. The call will return an error object indicating success or an error condition as appropriate.
- [BError configure](#) (ConfigInfo configInfo)
Configure the system for use. This includes mapping the individual physical Pickup channels to logical pickup channels.
- [BError getConfiguration](#) (ConfigInfo &configInfo)
Get the current configuration.
- [BError setControllInfo](#) (CycleParam params)
Sets the control information for the cycle type given. The parameters for the processing cycle are passed, this includes the Phase and State table information. The call will return an error object indicating success or an error.
- [BError getControllInfo](#) (BString cycleType, BUInt32 ring, BUInt32 puChannel, CycleParam ¶ms)
Gets the control information for the cycle type and puChannel number given. The call will return an error object indicating success or an error.
- [BError delControllInfo](#) (BString cycleType, BUInt32 ring, BUInt32 puChannel)
Deletes the control information for the cycle type and puChannel number given. The call will return an error object indicating success or an error.

- **BError** `getControllist (BList< CycleParamItem > &itemList)`
Gets the list of Cycle Parameters present in the system.
- **BError** `setNextCycle (BUInt32 cycleNumber, BString cycleType)`
Sets the cycle number and type for the next processing cycle. The call will return an error object indicating success or an error condition as appropriate. This should be called at least 10ms before the next CYCLE_START event.
- **BError** `test (BList< BError > &errors)`
Performs a basic test of the system returning a list of errors. The call will return an error object indicating success or an error condition as appropriate.
- **BError** `getStatus (BList< NameValue > &statusList)`
Returns the current status of the system. This information includes the number of Pick-Up's present and their individual status.
- **BError** `getStatistics (BList< NameValue > &statsList)`
Returns a list of the statistic values as name/value pairs. The call will return an error object indicating success or an error condition as appropriate.
- **BError** `getPuChannel (BUInt32 puChannel, PuChannel &puPhysChannel)`
Returns the physical module/Pupe/Channel number given a logical PickUp id. This can be used so that the individual PickUps test functions can be accessed etc.
- **BError** `setSimulation (Simulation simulation)`
Sets overall simulation modes.
- **BError** `getSimulation (Simulation &simulation)`
Gets current simulation modes.
- **BError** `setTestMode (PuChannel puPhysChannel, BUInt32 testOutput, BUInt32 timingDisableMask)`
The signal source for the digital test output connector. 0: None, 1: FrefLocal. The timingDisableMask bit mask defines which of the timing inputs should be disabled. If a timing input is disabled it can be still operated by software command.
- **BError** `setTimingSignals (PuChannel puPhysChannel, BUInt32 timingSignals)`
This function sets the given timing signals to the values as defined in the timingSignals bit array.
- **BError** `captureDiagnostics (PuChannel puPhysChannel, TestCaptureInfo captureInfo, BArray< BUInt64 > &data)`
This function will capture the diagnostics.
- **BError** `setTestData (PuChannel puPhysChannel, BUInt32 on, BArray< BUInt32 > data)`
This function will set a PU channel to sample data from memory rather than the ADC's.
- **BError** `setPupeConfig (PuChannel puPhysChannel, PupeConfig pupeConfig)`
Sets special PUPE configuration for test purposes.
- **BError** `getPupeConfig (PuChannel puPhysChannel, PupeConfig &pupeConfig)`
Gets special PUPE configuration for test purposes.
- **BError** `puServerStarted (BUInt32 number)`
A TmsPuServer has started.

Additional Inherited Members

7.118.1 Detailed Description

This interface provides functions to control, test and get statistics from the TMS as a whole.

7.118.2 Constructor & Destructor Documentation

7.118.2.1 `Tms::TmsControl::TmsControl (BString name = " ")`

7.118.3 Member Function Documentation

7.118.3.1 BError Tms::TmsControl::captureDiagnostics (PuChannel *puPhysChannel*, TestCaptureInfo *captureInfo*, BArray< BUInt64 > & *data*)

This function will capture the diagnostics.

7.118.3.2 BError Tms::TmsControl::configure (ConfigInfo *configInfo*)

Configure the system for use. This includes mapping the individual physical Pickup channels to logical pickup channels.

Parameters

<i>configInfo</i>	The channel mapping table.
-------------------	----------------------------

This function configures the logical to physical channel mapping table.

7.118.3.3 BError Tms::TmsControl::delControllInfo (BString *cycleType*, BUInt32 *ring*, BUInt32 *puChannel*)

Deletes the control information for the cycle type and puChannel number given. The call will return an error object indicating success or an error.

Parameters

<i>cycleType</i>	This string defines the cycle type to delete from the database.
<i>puChannel</i>	This defines the specific channel to delete the information for. 0 means all channels.

This function will delete a set of Cycle parameters from the TMS's Cycle parameter database.

7.118.3.4 BError Tms::TmsControl::getConfiguration (ConfigInfo & *configInfo*)

Get the current configuration.

Parameters

<i>configInfo</i>	The channel mapping table that is filled in with the current curent channel mapping.
-------------------	--

This function reads the current logical to physical channel mapping table.

7.118.3.5 BError Tms::TmsControl::getControllInfo (BString *cycleType*, BUInt32 *ring*, BUInt32 *puChannel*, CycleParam & *params*)

Gets the control information for the cycle type and puChannel number given. The call will return an error object indicating success or an error.

Parameters

<i>cycleType</i>	This string defines the cycle type for which to get the information.
<i>puChannel</i>	This defines the channel to get the information for. 0 means all channels.
<i>params</i>	The resuting cycle parameters are placed in this object.

This function reads back the set of Cycle parameters for the given cycle type and channel number. Normal the same cycle parameters are used for all PUPE engines. In this case setting the puChannel to 0 reads the Cycle Parameters that are being used on all channels. If a specific channel has other parameters the puChannel variable can be set to the appropriate channel number to get its particular settings.

7.118.3.6 BError Tms::TmsControl::getControlList (BList< CycleParamItem > & *itemList*)

Gets the list of Cycle Parameters present in the system.

Parameters

<i>itemList</i>	The list of CycleType information is returned.
-----------------	--

This function will return a list of entries describing the Cycle Parameter sets present in the TMS database.

7.118.3.7 **BError Tms::TmsControl::getPuChannel (BUInt32 puChannel, PuChannel & puPhysChannel)**

Returns the physical module/Pupe/Channel number given a logical PickUp id. This can be used so that the individual PickUps test functions can be accessed etc.

Parameters

<i>puChannel</i>	The logical channel number.
<i>puPhysChannel</i>	The physical channel identifier is returned in this variable.

This function is given a logical pick-up channel number. It will return the physical module, pupe number and pupe channel that has been allocated to this channel.

7.118.3.8 **BError Tms::TmsControl::getPupeConfig (PuChannel puPhysChannel, PupeConfig & pupeConfig)**

Gets special PUPE configuration for test purposes.

Parameters

<i>puPhysChannel</i>	The physical channel identifier.
<i>pupeConfig</i>	The returned configuration parameters.

This function returns the current configuration of the given channel.

7.118.3.9 **BError Tms::TmsControl::getSimulation (Simulation & simulation)**

Gets current simulation modes.

7.118.3.10 **BError Tms::TmsControl::getStatistics (BList< NameValue > & statsList)**

Returns a list of the statistic values as name/value pairs. The call will return an error object indicating success or an error condition as appropriate.

Parameters

<i>statsList</i>	The statistics list is placed in this list object.
------------------	--

This function gets the statistics values from the TMS system. It returns a list of name/value pairs.

7.118.3.11 **BError Tms::TmsControl::getStatus (BList< NameValue > & statusList)**

Returns the current status of the system. This information includes the number of Pick-Up's present and their individual status.

Parameters

<i>statusList</i>	The list of status items is placed in this list object.
-------------------	---

This function gets the status of the TMS system. It returns a list of name/value pairs.

7.118.3.12 **BError Tms::TmsControl::getVersion (BString & version)**

Gets the software version.

Parameters

<i>version</i>	A string variable filled in with the version number string.
----------------	---

7.118.3.13 BError Tms::TmsControl::init ()

Initialises the system including resetting all of the PUPE engines firmware. The call will return an error object indicating success or an error condition as appropriate.

This function restarts the TMS system. It re-initialises each of the TmsPuServer processes running on the Module Controllers and reboots each of the PUPE boards from scratch loading the current FPGA firmware. All errors and statistics values are reset.

7.118.3.14 BError Tms::TmsControl::puServerStarted (BUInt32 *number*)

A TmsPuServer has started.

Parameters

<i>number</i>	The number of the PuServer started.
---------------	-------------------------------------

This is an internal function called by the TmsPuServer processes to indicate to the TmsServer that they have just started running and are present in the system. The TmsServer will initialise the appropriate tmsPuServer program and its individual PUPE engines on receipt of this call.

7.118.3.15 BError Tms::TmsControl::setControllInfo (CycleParam *params*)

Sets the control information for the cycle type given. The parameters for the processing cycle are passed, this includes the Phase and State table information. The call will return an error object indicating success or an error.

Parameters

<i>params</i>	Cycle information parameters (state/phase table information).
---------------	---

This function over-writes or adds an entry in the Cycle Parameter database. The Cycle Parameters define the setting for each processing cycle including the state and phase tables for the PUPE FPGA engines.

7.118.3.16 BError Tms::TmsControl::setNextCycle (BUInt32 *cycleNumber*, BString *cycleType*)

Sets the cycle number and type for the next processing cycle. The call will return an error object indicating success or an error condition as appropriate. This should be called at least 10ms before the next CYCLE_START event.

Parameters

<i>cycleNumber</i>	This is the next cycle number. This should be an incrementing 32bit unsigned value.
<i>cycleType</i>	This is a string defining the cycle type for the next cycle.

This call configures the TMS system for the next processing cycle. It defines the cycle number that will be used to tag data captured during the cycle and it defines the type of machine cycle. The cycleType is used to lookup the appropriate state/phase table information to use in the FPGA's. The call should be made at least 10ms before the CYCLE_START event for the cycle it refers to. This gives time for the FPGA's to be loaded with the appropriate state/phase table information. As the function is time critical, the communications channel should be set to a high priority using the [setPriority\(\)](#) call and the processing threads priority should be set to high using the [setProcessPriority\(\)](#) call. The call will return the error: "ErrorCycleNumber", "The next cycle has already started" if the call has not completed before the CYCLE_START event. All client data reads, for this cycle, will also return this error message.

7.118.3.17 BError Tms::TmsControl::setProcessPriority (BUInt32 *priority*)

Sets the priority of the process servicing this service.

Parameters

<i>priority</i>	This is the priority of the process. It can be set to one of: PriorityLow, PriorityNormal, Priority-High.
-----------------	---

7.118.3.18 **BError Tms::TmsControl::setPupeConfig (PuChannel *puPhysChannel*, PupeConfig *pupeConfig*)**

Sets special PUPE configuration for test purposes.

Parameters

<i>puPhysChannel</i>	The physical channel identifier.
<i>pupeConfig</i>	The configuration parameters to use.

This functions sets up some special configuration parameters for the PUPE channel. It is used mainly for diagnostics and test purposes. The main settings it can affect are: The ADC Clock sources PLL synchronisation, internal timing for the digital timing signals and the enabling/dissabling of the BLR algorithm.

7.118.3.19 **BError Tms::TmsControl::setSimulation (Simulation *simulation*)**

Sets overall simulation modes.

7.118.3.20 **BError Tms::TmsControl::setTestData (PuChannel *puPhysChannel*, BInt32 *on*, BArray< BUInt32 > *data*)**

This function will set a PU channel to sample data from memory rather than the ADC's.

Parameters

<i>puPhysChannel</i>	The physical channel identifier.
<i>on</i>	Boolean to enable the internal data source. 0 is off, 1 is on.
<i>data</i>	The array of 32bit data values to use as the FREF,Sigma,DeltaX and DeltaY test signal.

This call loads the PUPE systems test data SDRAM with the data passed in the data array. It then sets up the individual channel to sources its FREF, Sigma, DeltaX and DelatY signals from the test SDRAM. The data source should have a multiple of 2 samples. The "on" parameter is used to enable or disable the individual channels inputs from this test data SDRAM.

7.118.3.21 **BError Tms::TmsControl::setTestMode (PuChannel *puPhysChannel*, BUInt32 *testOutput*, BUInt32 *timingDisableMask*)**

The signal source for the digital test output connector. 0: None, 1: FrefLocal. The timingDisableMask bit mask defines which of the timing inputs should be disabled. If a timing input is disabled it can be still operated by software command.

Parameters

<i>puPhysChannel</i>	The physical channel identifier.
<i>testOutput</i>	The signal to output on the test output. 0 is FREF any other value is undefined at the moment.
<i>timingDisable-Mask</i>	This 8 bit mask defines which of the timing input signals are disabled.

This function sets up a particular pick-up channel's digital test output source and allows the channels input timing signals to be set to a software driven mode rather than taken from the hardware timing inputs. The timing mask bits are: 7 - FREF, 6 - HCHANGE, 5 - INJECTION, 4 - CAL_STOP, 3 - CAL_START, 2 - CYCLE_STOP, 1 - CYCLE_START, 0 - SYSCLOCK

7.118.3.22 **BError Tms::TmsControl::setTimingSignals (PuChannel *puPhysChannel*, BUInt32 *timingSignals*)**

This function sets the given timing signals to the values as defined in the timingSignals bit array.

Parameters

<i>puPhysChannel</i>	The physical channel identifier.
<i>timingSignals</i>	The 8 bit mask defining the state of the software driven timing signals.

If the `setTestMode()` function had been used to "enable" particular timing signals to be driven by software, then this function can be used to set/reset particular timing signals for the pick-up channel given. The timing signals bits are: 7 - FREF, 6 - HCHANGE, 5 - INJECTION, 4 - CAL_STOP, 3 - CAL_START, 2 - CYCLE_STOP, 1 - CYCLE_START, 0 - SYSCLOCK

7.118.3.23 BError Tms::TmsControl::test (BList< BError > & errors)

Performs a basic test of the system returning a list of errors. The call will return an error object indicating success or an error condition as appropriate.

Parameters

<i>errors</i>	The list of errors is placed in this list object.
---------------	---

This function will perform a test of the TMS system. It will report each test performed and the status of the test in the `BError` object. A status value of 0 indicates all was Ok, any other value is an error where the number indicates the error. A string gives the test name and the Ok or error condition as a string.

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

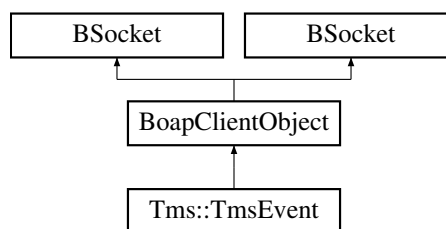
- [TmsC.h](#)
- [TmsC.cc](#)
- [tmsFunctions.dox](#)

7.119 Tms::TmsEvent Class Reference

This interface provides functions for events to be sent to clients from the TMS as a whole.

```
#include <TmsC.h>
```

Inheritance diagram for Tms::TmsEvent:



Public Member Functions

- `TmsEvent (BString name="")`
- `BError errorEvent (BUInt32 cycleNumber, BError error)`
This event function gets called on a system error. The errorEvent object contains an error number and string describing the error. The getStatus() call can be used to fetch further information.
- `BError cycleStartEvent (BUInt32 cycleNumber)`
This event function gets called on the CYCLE_START event with the cycle number about to be processed.
- `BError cycleStopEvent (BUInt32 cycleNumber)`
This event function gets called on the CYCLE_END event with the cycle number completed.
- `BError dataEvent (DataInfo dataInfo)`
This event function gets called when some requested data becomes available. The DataInfo object contains information on the data. The getData() call can be used to fetch the actual data.

Additional Inherited Members

7.119.1 Detailed Description

This interface provides functions for events to be sent to clients from the TMS as a whole.

7.119.2 Constructor & Destructor Documentation

7.119.2.1 `Tms::TmsEvent::TmsEvent (BString name = " ")`

7.119.3 Member Function Documentation

7.119.3.1 `BError Tms::TmsEvent::cycleStartEvent (BUInt32 cycleNumber)`

This event function gets called on the CYCLE_START event with the cycle number about to be processed.

7.119.3.2 `BError Tms::TmsEvent::cycleStopEvent (BUInt32 cycleNumber)`

This event function gets called on the CYCLE_END event with the cycle number completed.

7.119.3.3 `BError Tms::TmsEvent::dataEvent (DataInfo dataInfo)`

This event function gets called when some requested data becomes available. The [DataInfo](#) object contains information on the data. The `getData()` call can be used to fetch the actual data.

7.119.3.4 `BError Tms::TmsEvent::errorEvent (BUInt32 cycleNumber, BError error)`

This event function gets called on a system error. The `errorEvent` object contains an error number and string describing the error. The `getStatus()` call can be used to fetch further information.

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

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

7.120 Tms::TmsEventServerList Class Reference

```
#include <TmsEventServerList.h>
```

Public Member Functions

- [TmsEventServerList \(\)](#)
- [~TmsEventServerList \(\)](#)
- [BError append \(BString name\)](#)
- [BError del \(BString name\)](#)
- [BError errorEvent \(BUInt32 cycleNumber, BError error\)](#)

This event function gets called on a system error. The errorEvent object contains an error number and string describing the error. The getStatus() call can be used to fetch further information.
- [BError cycleStartEvent \(BUInt32 cycleNumber\)](#)

This event function gets called on the CYCLE_START event with the cycle number about to be processed.
- [BError cycleStopEvent \(BUInt32 cycleNumber\)](#)

This event function gets called on the CYCLE_END event with the cycle number completed.

- [BError dataEvent \(DataInfo dataInfo\)](#)

This event function gets called when some requested data becomes available. The [DataInfo](#) object contains information on the data. The `getData()` call can be used to fetch the actual data.

Private Attributes

- [BMutex olock](#)
- [BList< TmsEvent * > oeventServers](#)

7.120.1 Constructor & Destructor Documentation

7.120.1.1 `Tms::TmsEventServerList::TmsEventServerList ()`

7.120.1.2 `Tms::TmsEventServerList::~~TmsEventServerList ()`

7.120.2 Member Function Documentation

7.120.2.1 `BError Tms::TmsEventServerList::append (BString name)`

7.120.2.2 `BError Tms::TmsEventServerList::cycleStartEvent (BUInt32 cycleNumber)`

This event function gets called on the CYCLE_START event with the cycle number about to be processed.

7.120.2.3 `BError Tms::TmsEventServerList::cycleStopEvent (BUInt32 cycleNumber)`

This event function gets called on the CYCLE_END event with the cycle number completed.

7.120.2.4 `BError Tms::TmsEventServerList::dataEvent (DataInfo dataInfo)`

This event function gets called when some requested data becomes available. The [DataInfo](#) object contains information on the data. The `getData()` call can be used to fetch the actual data.

7.120.2.5 `BError Tms::TmsEventServerList::del (BString name)`

7.120.2.6 `BError Tms::TmsEventServerList::errorEvent (BUInt32 cycleNumber, BError error)`

This event function gets called on a system error. The `errorEvent` object contains an error number and string describing the error. The `getStatus()` call can be used to fetch further information.

7.120.3 Member Data Documentation

7.120.3.1 `BList<TmsEvent*> Tms::TmsEventServerList::oeventServers [private]`

7.120.3.2 `BMutex Tms::TmsEventServerList::olock [private]`

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

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

7.121 Tms::TmsPhase Union Reference

The [Tms](#) Phase Table Entry.

```
#include <TmsLib.h>
```

Public Attributes

- struct {
 - unsigned int [lo1](#):1
 - unsigned int [blr](#):1
 - unsigned int [gate](#):1
 - unsigned int [lo2](#):1
 - unsigned int [spare](#):2
 - unsigned int [meanFilter1](#):1
 - unsigned int [meanFilter2](#):1};
- unsigned char [value](#)

7.121.1 Detailed Description

The [Tms](#) Phase Table Entry.

7.121.2 Member Data Documentation

7.121.2.1 [struct { ... }](#)

7.121.2.2 [unsigned int Tms::TmsPhase::blr](#)

7.121.2.3 [unsigned int Tms::TmsPhase::gate](#)

7.121.2.4 [unsigned int Tms::TmsPhase::lo1](#)

7.121.2.5 [unsigned int Tms::TmsPhase::lo2](#)

7.121.2.6 [unsigned int Tms::TmsPhase::meanFilter1](#)

7.121.2.7 [unsigned int Tms::TmsPhase::meanFilter2](#)

7.121.2.8 [unsigned int Tms::TmsPhase::spare](#)

7.121.2.9 [unsigned char Tms::TmsPhase::value](#)

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

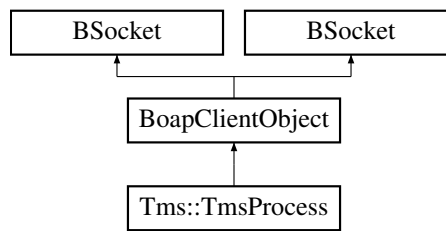
- [TmsLib.h](#)

7.122 Tms::TmsProcess Class Reference

This interface provides functions to capture data from the TMS as a whole.

```
#include <TmsC.h>
```


Inheritance diagram for Tms::TmsProcess:



Public Member Functions

- [TmsProcess](#) (BString name="")
- [BError getVersion](#) (BString &version)

Gets the software version.
- [BError getCycleInfo](#) (BUInt32 &cycleNumber, BString &cycleType)

Gets the current cycle number and type.
- [BError getCycleInformation](#) (BUInt32 cycleNumber, CycleInformation &cycleInformation)

Gets information on given cycle number.
- [BError getCycleTypeInfo](#) (BString cycleType, CycleTypeInfo &cycleTypeInfo)

Gets information on given cycle Type.
- [BError getData](#) (DataInfo dataInfo, Data &data)

This function returns a set of data from the data present in the data cache or directly from the Pick-Up processing engines. The [DataInfo](#) object describes the data required. The call will return the required data along with an error object indicating success or an error condition as appropriate. The call will block until data is ready.
- [BError addEventServer](#) (BString name)

This call adds an event server to call on events such as the "dataEvent" generated by the requestData(0 call as well as error events. The Client will use this to notify the TmsServer of its local [TmsEvent](#) object.
- [BError requestData](#) (DataInfo dataInfo)

This adds a request for some data. The [DataInfo](#) object defines the data required. This request can be made at any time. If the data is present in cache the data will be available immediately, if not the system will await the data from a subsequent processing cycle. When the data is available a "data" event will be sent to the client. Not that it is not necessary to use requestData. The client can call [getData\(\)](#) directly although this call will block until the data is actually ready.

Additional Inherited Members

7.122.1 Detailed Description

This interface provides functions to capture data from the TMS as a whole.

7.122.2 Constructor & Destructor Documentation

7.122.2.1 Tms::TmsProcess::TmsProcess (BString name = " ")

Parameters

<i>name</i>	The name of the TmsProcess BOAP object to connect to.
-------------	---

The BOAP object name has the general form: "//HostName/ObjectName"

7.122.3 Member Function Documentation

7.122.3.1 BError Tms::TmsProcess::addEventServer (BString *name*)

This call adds an event server to call on events such as the "dataEvent" generated by the requestData(0 call as well as error events. The Client will use this to notify the TmsServer of its local [TmsEvent](#) object.

Parameters

<i>name</i>	The BOAP object name to add.
-------------	------------------------------

Adds an event server that gets called on certain TmsServer events such as data ready, CYCLE_START, CYCLE_STOP and errors.

7.122.3.2 **BError** Tms::TmsProcess::getCycleInfo (**BUInt32** & *cycleNumber*, **BString** & *cycleType*)

Gets the current cycle number and type.

Parameters

<i>cycleNumber</i>	The current cycle number is returned here
<i>cycleType</i>	The current cycle type is returned here.

This function returns the current TMS cycle number being processed and the type of the cycle.

7.122.3.3 **BError** Tms::TmsProcess::getCycleInformation (**BUInt32** *cycleNumber*, **CycleInformation** & *cycleInformation*)

Gets information on given cycle number.

Parameters

<i>cycleNumber</i>	The current cycle number to get information on
<i>cycleInformation</i>	The returned cycle information

This function provides information on the given cycle. It interrogates the first PUPE channel and returns the Cycle information based on the internal state of that channel. The main information returned includes the cycle type and a list of all of the Cycle Periods captured and the times and amount of data in each.

7.122.3.4 **BError** Tms::TmsProcess::getCycleTypeInfo (**BString** *cycleType*, **CycleTypeInfo** & *cycleTypeInfo*)

Gets information on given cycle Type.

7.122.3.5 **BError** Tms::TmsProcess::getData (**DataInfo** *dataInfo*, **Data** & *data*)

This function returns a set of data from the data present in the data cache or directly from the Pick-Up processing engines. The [DataInfo](#) object describes the data required. The call will return the required data along with an error object indicating success or an error condition as appropriate. The call will block until data is ready.

Parameters

<i>dataInfo</i>	Information on the type of data required.
<i>data</i>	The raw data is returned in this object.

This is the main user function used by clients of the TMS system. It is used to return portions of the acquired data. The fields of the [dataInfo](#) parameter define which data is required and are defined in the [DataInfo](#) class documentation.

The call will check to see if the data for the cycle number requested is still present in the PUPE memory. The PUPE memory has enough storage for about 3 seconds worth of data (3 processing cycles). If the data has gone the call will return the error "ErrorDataGone". If the system has not processed the requested cycle, but will do so within 256 seconds, the call will block awaiting the data.

If the channel number is given as 0 the call will interrogate each of the Pick-Up channels and return the combined data from all of them. Note that this could take significant time and may not be possible if the parameter numValues is large. Within the [Data](#) structure returned there is an array of error values, one per channel. If an error occurs on any set of the channels the call will return the first error that occurred and the complete list of errors in the errors array. The actual data will be returned for all channels that did not have an error. Those channels that had an error will have data values of 0 returned.

If the bunch number is given as 0, then the system will return the data for all of the bunches.

The data will be returned in the following order, where B - Bunch, C - Channel:

[C1.B1, C1.B2, C1.B3, C1.B4], [C1.B1, C1.B2, C1.B3, C1.B4], ... [C2.B1, C2.B2, C2.B3, C2.B4], [C2.B1, C2.B2, C2.B3, C2.B4], ...

That is the data is ordered by bunch, then sample, then channel. See the TMS Software documentation manual for more details of this functions operation.

7.122.3.6 BError Tms::TmsProcess::getVersion (BString & version)

Gets the software version.

Parameters

<i>version</i>	A string variable filled in with the version number string.
----------------	---

7.122.3.7 BError Tms::TmsProcess::requestData (DataInfo dataInfo)

This adds a request for some data. The [DataInfo](#) object defines the data required. This request can be made at any time. If the data is present in cache the data will be available immediately, if not the system will await the data from a subsequent processing cycle. When the data is available a "data" event will be sent to the client. Not that it is not necessary to use requestData. The client can call [getData\(\)](#) directly although this call will block until the data is actually ready.

Parameters

<i>dataInfo</i>	Information on the type of data required.
-----------------	---

This calls sets up a request for data. The dataInfo parameter works in the same manner as the "getData" call, defining the portion of data required. This call will return immediatly. Assuming the client has informed the TMS system of an event server object using the "addEventServer" call, then the client will receive the "dataEvent" event when the data become available. The client can then fetch the data using the conventional "getData" call. In the current version of the software the "requestData" call simply sends a message when the data for the requested cycle is ready. In future implementations the TMS system could actually fetch the data automatically from the PUPE boards and store it in memory ready for later retrieval by the getData call.

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

- [TmsC.h](#)
- [TmsC.cc](#)
- [tmsFunctions.dox](#)

7.123 Tms::TmsState Union Reference

The [Tms](#) State entry.

```
#include <TmsLib.h>
```

Public Attributes

- struct {
 - unsigned int [acquireData](#):1
 - unsigned int [pllReference1](#):1
 - unsigned int [pllReference2](#):1
 - unsigned int [pllFeedbackSelect](#):1
 - unsigned int [pllLO1FromAddress](#):1
 - unsigned int [pllLO2FromAddress](#):1

```
    unsigned int bit6:1
    unsigned int bit7:1
    unsigned int cycleStop:4
    unsigned int calStop:4
    unsigned int calStart:4
    unsigned int injection:4
    unsigned int hchange:4
    unsigned int delay:4
};
```

- unsigned int [value](#)

7.123.1 Detailed Description

The [Tms](#) State entry.

7.123.2 Member Data Documentation

7.123.2.1 [struct { ... }](#)

7.123.2.2 [unsigned int Tms::TmsState::acquireData](#)

7.123.2.3 [unsigned int Tms::TmsState::bit6](#)

7.123.2.4 [unsigned int Tms::TmsState::bit7](#)

7.123.2.5 [unsigned int Tms::TmsState::calStart](#)

7.123.2.6 [unsigned int Tms::TmsState::calStop](#)

7.123.2.7 [unsigned int Tms::TmsState::cycleStop](#)

7.123.2.8 [unsigned int Tms::TmsState::delay](#)

7.123.2.9 [unsigned int Tms::TmsState::hchange](#)

7.123.2.10 [unsigned int Tms::TmsState::injection](#)

7.123.2.11 [unsigned int Tms::TmsState::pllFeedbackSelect](#)

7.123.2.12 [unsigned int Tms::TmsState::pllLO1FromAddress](#)

7.123.2.13 [unsigned int Tms::TmsState::pllLO2FromAddress](#)

7.123.2.14 [unsigned int Tms::TmsState::pllReference1](#)

7.123.2.15 [unsigned int Tms::TmsState::pllReference2](#)

7.123.2.16 [unsigned int Tms::TmsState::value](#)

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

- [TmsLib.h](#)

Chapter 8

File Documentation

8.1 /src/cern/tms/beam/libBeam/BArray.h File Reference

```
#include <BTypes.h>
#include <vector>
#include <algorithm>
```

Classes

- class [BArray< T >](#)

Macros

- #define [BArrayLoop](#)(list, i) for([BUInt](#) i = 0; i < list.number(); i++)

8.1.1 Macro Definition Documentation

8.1.1.1 #define [BArrayLoop](#)(*list*, *i*) for([BUInt](#) i = 0; i < list.number(); i++)

8.2 /src/cern/tms/beam/libBeam/BAtomic.h File Reference

```
#include <BTypes.h>
```

Classes

- class [BAtomic< Type >](#)
BAtomic class.

Typedefs

- typedef [BAtomic< BInt32 >](#) [BAtomicInt32](#)
- typedef [BAtomic< BInt64 >](#) [BAtomicInt64](#)
- typedef [BAtomic< BUInt32 >](#) [BAtomicUInt32](#)
- typedef [BAtomic< BUInt64 >](#) [BAtomicUInt64](#)

8.2.1 Typedef Documentation

8.2.1.1 typedef `BAtomic<BInt32>` `BAtomicInt32`

8.2.1.2 typedef `BAtomic<BInt64>` `BAtomicInt64`

8.2.1.3 typedef `BAtomic<BUInt32>` `BAtomicUInt32`

8.2.1.4 typedef `BAtomic<BUInt64>` `BAtomicUInt64`

8.3 `/src/cern/tms/beam/libBeam/BAtomicCount.h` File Reference

```
#include <bits/atomicity.h>
```

Classes

- class [BAtomicCount](#)
BAtomicCount class.

8.4 `/src/cern/tms/beam/libBeam/BBuffer.cpp` File Reference

```
#include <stdlib.h>  
#include <memory.h>  
#include <BBuffer.h>  
#include <BEndian.h>  
#include <BTimeStamp.h>  
#include <BComplex.h>
```

Variables

- const int `roundSize` = 256

8.4.1 Variable Documentation

8.4.1.1 const int `roundSize` = 256

8.5 `/src/cern/tms/beam/libBeam/BBuffer.h` File Reference

```
#include <BTypes.h>  
#include <BString.h>  
#include <BError.h>  
#include <BComplex.h>  
#include <BEndian.h>
```

Classes

- class [BBuffer](#)
- class [BBufferStore](#)

Macros

- `#define BBigEndian 0`

8.5.1 Macro Definition Documentation

8.5.1.1 `#define BBigEndian 0`

8.6 /src/cern/tms/beam/libBeam/BComms.cpp File Reference

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

8.7 /src/cern/tms/beam/libBeam/BComms.h File Reference

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

Classes

- class [BComms](#)

8.8 /src/cern/tms/beam/libBeam/BComplex.h File Reference

```
#include <BTypes.h>
#include <complex>
#include <algorithm>
```

Typedefs

- `typedef std::complex< double > BComplex`
- `typedef std::complex< float > BComplex32`
- `typedef std::complex< double > BComplex64`

8.8.1 Typedef Documentation

8.8.1.1 `typedef std::complex<double> BComplex`

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

8.8.1.2 `typedef std::complex<float> BComplex32`

8.8.1.3 `typedef std::complex<double> BComplex64`

8.9 /src/cern/tms/beam/libBeam/BCond.cpp File Reference

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

8.10 /src/cern/tms/beam/libBeam/BCond.h File Reference

```
#include <pthread.h>
```

Classes

- class [BCond](#)

8.11 /src/cern/tms/beam/libBeam/BCondInt.cpp File Reference

```
#include <BCondInt.h>
#include <sys/time.h>
#include <stdio.h>
#include <errno.h>
```

Functions

- static struct timespec [getTimeout](#) (uint32_t timeOutUs)

8.11.1 Function Documentation

8.11.1.1 static struct timespec [getTimeout](#) (uint32_t *timeOutUs*) [static]

8.12 /src/cern/tms/beam/libBeam/BCondInt.h File Reference

```
#include <BTypes.h>
#include <pthread.h>
```

Classes

- class [BCondInt](#)
Thread conditional value.
- class [BCondValue](#)
Thread conditional value.
- class [BCondBool](#)
Thread conditional boolean.
- class [BCondWrap](#)
- class [BCondResource](#)
Resource lock.

8.13 /src/cern/tms/beam/libBeam/BConfig.cpp File Reference

```
#include <BConfig.h>
#include <string.h>
```

8.14 /src/cern/tms/beam/libBeam/BConfig.h File Reference

```
#include <BDict.h>
#include <BFile.h>
#include <BMutex.h>
```

Classes

- class [BConfig](#)

This class implements the configuration file access.

8.15 /src/cern/tms/beam/libBeam/BCrc16.cpp File Reference

```
#include <BCrc16.h>
```

Functions

- [BUInt16 bcrc16](#) (void *buf, [BUInt16 len](#))

Variables

- static const [BUInt8 table_crc_hi](#) []
- static const [BUInt8 table_crc_lo](#) []

8.15.1 Function Documentation

8.15.1.1 [BUInt16 bcrc16](#) (void * buf, [BUInt16 len](#))

8.15.2 Variable Documentation

8.15.2.1 [const BUInt8 table_crc_hi](#)[] [static]

Initial value:

```
= {
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
  0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
  0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
  0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
  0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
  0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
  0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
  0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
  0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
```

```

0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
}

```

8.15.2.2 const BUInt8 table_crc_lo[] [static]

Initial value:

```

= {
0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
0x43, 0x83, 0x41, 0x81, 0x80, 0x40
}

```

8.16 /src/cern/tms/beam/libBeam/BCrc16.h File Reference

```
#include <BTypes.h>
```

Functions

- [BUInt16 bcrc16](#) (void *buf, BUInt16 len)

8.16.1 Function Documentation

8.16.1.1 BUInt16 bcrc16 (void * buf, BUInt16 len)

8.17 /src/cern/tms/beam/libBeam/BDate.cpp File Reference

```
#include <BDate.h>
#include <sys/time.h>
```

Functions

- void [toBString](#) (BDate &v, BString &s)
- void [fromBString](#) (BString &s, BDate &v)

Variables

- static int [mon_yday](#) [2][13]

8.17.1 Function Documentation

8.17.1.1 void [fromBString](#) (BString & s, BDate & v)

8.17.1.2 void [toBString](#) (BDate & v, BString & s)

8.17.2 Variable Documentation

8.17.2.1 int [mon_yday](#)[2][13] [static]

Initial value:

```
= {
    { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 },
    { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366 }
}
```

8.18 /src/cern/tms/beam/libBeam/BDate.h File Reference

```
#include <stdint.h>
#include <BError.h>
```

Classes

- class [BDate](#)

Functions

- void [toBString](#) (BDate &v, BString &s)
- void [fromBString](#) (BString &s, BDate &v)

8.18.1 Function Documentation

8.18.1.1 void [fromBString](#) (BString & s, BDate & v)

8.18.1.2 void [toBString](#) (BDate & v, BString & s)

8.19 /src/cern/tms/beam/libBeam/BDebug.cpp File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <unistd.h>
#include <syslog.h>
#include <sys/time.h>
#include <stdarg.h>
#include <fcntl.h>
#include <execinfo.h>
#include <ctype.h>
#include <BDebug.h>
#include <errno.h>
#include <linux/unistd.h>
```

Macros

- `#define BTRACE_SIZE 100`

Functions

- void `hd8` (void *data, unsigned int n)
- void `hd8a` (void *data, unsigned int n)
- void `hda8` (void *data, unsigned int n)
- void `hd32` (void *data, unsigned int n)
- void `hda32` (void *data, unsigned int n)
- double `getTime` ()
- void `setDebug` (int d)
- void `tprintf` (int log, const char *fmt,...)
- pid_t `gettid` ()

Variables

- int `bdebug`
- const unsigned int `STRBUF_SIZE` = (64 * 1024)

8.19.1 Macro Definition Documentation

8.19.1.1 `#define BTRACE_SIZE 100`

8.19.2 Function Documentation

8.19.2.1 pid_t `gettid` ()

8.19.2.2 double `getTime` ()

8.19.2.3 void `hd32` (void * *data*, unsigned int *n*)

8.19.2.4 void `hd8` (void * *data*, unsigned int *n*)

8.19.2.5 void `hd8a` (void * *data*, unsigned int *n*)

8.19.2.6 void `hda32` (void * *data*, unsigned int *n*)

8.19.2.7 void `hda8` (void * *data*, unsigned int *n*)

8.19.2.8 void `setDebug` (int *d*)

8.19.2.9 void `tprintf` (int *log*, const char * *fmt*, ...)

8.19.3 Variable Documentation

8.19.3.1 int `bdebug`

8.19.3.2 const unsigned int `STRBUF_SIZE` = (64 * 1024)

8.20 /src/cern/tms/beam/libBeam/BDebug.h File Reference

```
#include <stdio.h>
#include <syslog.h>
#include <time.h>
```

Classes

- class [BDebugBacktrace](#)

Macros

- #define `BDebug_STD` 0x000001
- #define `dprintf`(level, fmt, a...)
General debug functions.
- #define `nprintf`(fmt, a...) syslog(LOG_NOTICE, fmt, ##a)
Warnings and errors logging.
- #define `wprintf`(fmt, a...) syslog(LOG_WARNING, fmt, ##a)
- #define `eprintf`(fmt, a...) syslog(LOG_ERR, fmt, ##a)

Functions

- void `hd8` (void *data, unsigned int n)
- void `hd8a` (void *data, unsigned int n)
- void `hda8` (void *data, unsigned int n)
- void `hd32` (void *data, unsigned int n)
- void `hds32` (void *data, unsigned int n)
- double `getTime` ()
- void `setDebug` (int debug)
- void `tprintf` (int log, const char *fmt,...)
- pid_t `gettid` ()

Variables

- int `bdebug`

8.20.1 Macro Definition Documentation

8.20.1.1 `#define BDebug_STD 0x000001`

8.20.1.2 `#define dprintf(level, fmt, a...)`

General debug functions.

8.20.1.3 `#define eprintf(fmt, a...) syslog(LOG_ERR, fmt, ##a)`

8.20.1.4 `#define nprintf(fmt, a...) syslog(LOG_NOTICE, fmt, ##a)`

Warnings and errors logging.

8.20.1.5 `#define wprintf(fmt, a...) syslog(LOG_WARNING, fmt, ##a)`

8.20.2 Function Documentation

8.20.2.1 `pid_t gettid ()`

8.20.2.2 `double getTime ()`

8.20.2.3 `void hd32 (void * data, unsigned int n)`

8.20.2.4 `void hd8 (void * data, unsigned int n)`

8.20.2.5 `void hd8a (void * data, unsigned int n)`

8.20.2.6 `void hda8 (void * data, unsigned int n)`

8.20.2.7 `void hds32 (void * data, unsigned int n)`

8.20.2.8 `void setDebug (int debug)`

8.20.2.9 `void tprintf (int log, const char * fmt, ...)`

8.20.3 Variable Documentation

8.20.3.1 `int bdebug`

8.21 /src/cern/tms/beam/libBeam/BDict.cpp File Reference

```
#include <BDict.h>
```

Functions

- void [toBString](#) (const [BDictString](#) &v, [BString](#) &s)
- void [fromBString](#) (const [BString](#) &str, [BDictString](#) &v)
- [BString](#) [bdictStringToString](#) (const [BDictString](#) &dict)

8.21.1 Function Documentation

8.21.1.1 `BString` `bdictStringToString` (`const BDictString & dict`)

8.21.1.2 `void` `fromBString` (`const BString & str`, `BDictString & v`)

8.21.1.3 `void` `toBString` (`const BDictString & v`, `BString & s`)

8.22 /src/cern/tms/beam/libBeam/BDict.h File Reference

```
#include <BNameValue.h>
```

Classes

- class `BDictItem< Type >`
Template based Dictionary class.
- class `BDict< Type >`

Typedefs

- typedef `BDict< BString > BDictString`

Functions

- `void` `toBString` (`const BDictString &v`, `BString &s`)
- `void` `fromBString` (`const BString &s`, `BDictString &v`)
- `BString` `bdictStringToString` (`const BDictString &dict`)

8.22.1 Typedef Documentation

8.22.1.1 typedef `BDict<BString> BDictString`

8.22.2 Function Documentation

8.22.2.1 `BString` `bdictStringToString` (`const BDictString & dict`)

8.22.2.2 `void` `fromBString` (`const BString & s`, `BDictString & v`)

8.22.2.3 `void` `toBString` (`const BDictString & v`, `BString & s`)

8.23 /src/cern/tms/beam/libBeam/BDictMap.h File Reference

```
#include <BString.h>
#include <map>
```

Classes

- class `BDictMap< Value >`

Typedefs

- typedef [BDictMap< BString > BDictMapString](#)

8.23.1 Typedef Documentation

8.23.1.1 typedef [BDictMap<BString> BDictMapString](#)

8.24 /src/cern/tms/beam/libBeam/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](#)

8.24.1 Function Documentation

8.24.1.1 static int [wild](#) (const dirent * e) [static]

8.24.2 Variable Documentation

8.24.2.1 [BString wildString](#) [static]

8.25 /src/cern/tms/beam/libBeam/BDir.h File Reference

```
#include <BList.h>
#include <BString.h>
#include <BError.h>
#include <sys/stat.h>
```

Classes

- class [BDir](#)
File system directory class.

8.26 /src/cern/tms/beam/libBeam/BDuration.cpp File Reference

```
#include <BDuration.h>
#include <sys/time.h>
```

8.27 /src/cern/tms/beam/libBeam/BDuration.h File Reference

```
#include <stdint.h>
#include <BError.h>
```

Classes

- class [BDuration](#)

8.28 /src/cern/tms/beam/libBeam/BEndian.cpp File Reference

```
#include <BEndian.h>
#include <memory.h>
```

Functions

- void [bswap_copy](#) (int swap, const void *src, void *dst, [BUInt32](#) nBytes, const char *swapType)

8.28.1 Function Documentation

8.28.1.1 void [bswap_copy](#) (int *swap*, const void * *src*, void * *dst*, [BUInt32](#) *nBytes*, const char * *swapType*)

8.29 /src/cern/tms/beam/libBeam/BEndian.h File Reference

```
#include <BTypes.h>
#include <byteswap.h>
```

Macros

- #define [htobe16](#)(x) __bswap_16 (x)
- #define [htole16](#)(x) (x)
- #define [be16toh](#)(x) __bswap_16 (x)
- #define [le16toh](#)(x) (x)
- #define [htobe32](#)(x) __bswap_32 (x)
- #define [htole32](#)(x) (x)
- #define [be32toh](#)(x) __bswap_32 (x)
- #define [le32toh](#)(x) (x)
- #define [htobe64](#)(x) __bswap_64 (x)
- #define [htole64](#)(x) (x)
- #define [be64toh](#)(x) __bswap_64 (x)
- #define [le64toh](#)(x) (x)

Functions

- void [bswap_p8](#) (const void *s, void *d)
- void [bswap_p16](#) (const void *s, void *d)
- void [bswap_p32](#) (const void *s, void *d)

- void [bswap_p64](#) (const void *s, void *d)
- void [bswap_copy](#) (int swap, const void *src, void *dst, [BUInt32](#) nBytes, const char *swapType)
- [uint16_t htogle](#) ([uint16_t](#) v)
- [int16_t htogle](#) ([int16_t](#) v)
- [uint32_t htogle](#) ([uint32_t](#) v)
- [int32_t htogle](#) ([int32_t](#) v)
- [uint64_t htogle](#) ([uint64_t](#) v)
- [int64_t htogle](#) ([int64_t](#) v)
- [double htogle](#) ([double](#) v)
- [float htogle](#) ([float](#) v)
- [uint16_t htobe](#) ([uint16_t](#) v)
- [int16_t htobe](#) ([int16_t](#) v)
- [uint32_t htobe](#) ([uint32_t](#) v)
- [int32_t htobe](#) ([int32_t](#) v)
- [uint64_t htobe](#) ([uint64_t](#) v)
- [int64_t htobe](#) ([int64_t](#) v)
- [double htobe](#) ([double](#) v)
- [float htobe](#) ([float](#) v)
- [uint16_t letoh](#) ([uint16_t](#) v)
- [int16_t letoh](#) ([int16_t](#) v)
- [uint32_t letoh](#) ([uint32_t](#) v)
- [int32_t letoh](#) ([int32_t](#) v)
- [uint64_t letoh](#) ([uint64_t](#) v)
- [int64_t letoh](#) ([int64_t](#) v)
- [double letoh](#) ([double](#) v)
- [float letoh](#) ([float](#) v)
- [uint16_t betoh](#) ([uint16_t](#) v)
- [int16_t betoh](#) ([int16_t](#) v)
- [uint32_t betoh](#) ([uint32_t](#) v)
- [int32_t betoh](#) ([int32_t](#) v)
- [uint64_t betoh](#) ([uint64_t](#) v)
- [int64_t betoh](#) ([int64_t](#) v)
- [double betoh](#) ([double](#) v)
- [float betoh](#) ([float](#) v)

8.29.1 Macro Definition Documentation

8.29.1.1 `#define be16toh(x) __bswap_16(x)`

8.29.1.2 `#define be32toh(x) __bswap_32(x)`

8.29.1.3 `#define be64toh(x) __bswap_64(x)`

8.29.1.4 `#define htobe16(x) __bswap_16(x)`

8.29.1.5 `#define htobe32(x) __bswap_32(x)`

8.29.1.6 `#define htobe64(x) __bswap_64(x)`

8.29.1.7 `#define htogle16(x)(x)`

8.29.1.8 `#define htogle32(x)(x)`

8.29.1.9 `#define htogle64(x)(x)`

8.29.1.10 `#define le16toh(x)(x)`

8.29.1.11 `#define le32toh(x)(x)`

8.29.1.12 `#define le64toh(x)(x)`

8.29.2 Function Documentation

8.29.2.1 `uint16_t betoh (uint16_t v) [inline]`

8.29.2.2 `int16_t betoh (int16_t v) [inline]`

8.29.2.3 `uint32_t betoh (uint32_t v) [inline]`

8.29.2.4 `int32_t betoh (int32_t v) [inline]`

8.29.2.5 `uint64_t betoh (uint64_t v) [inline]`

8.29.2.6 `int64_t betoh (int64_t v) [inline]`

8.29.2.7 `double betoh (double v) [inline]`

8.29.2.8 `float betoh (float v) [inline]`

8.29.2.9 `void bswap_copy (int swap, const void * src, void * dst, BUInt32 nBytes, const char * swapType)`

8.29.2.10 `void bswap_p16 (const void * s, void * d) [inline]`

8.29.2.11 `void bswap_p32 (const void * s, void * d) [inline]`

8.29.2.12 `void bswap_p64 (const void * s, void * d) [inline]`

8.29.2.13 `void bswap_p8 (const void * s, void * d) [inline]`

8.29.2.14 `uint16_t htobe (uint16_t v) [inline]`

8.29.2.15 `int16_t htobe (int16_t v) [inline]`

8.29.2.16 `uint32_t htobe (uint32_t v) [inline]`

8.29.2.17 `int32_t htobe (int32_t v) [inline]`

8.29.2.18 `uint64_t htobe (uint64_t v) [inline]`

8.29.2.19 `int64_t htobe (int64_t v) [inline]`

8.29.2.20 `double htobe (double v) [inline]`

8.29.2.21 `float htobe (float v) [inline]`

8.29.2.22 `uint16_t htobe (uint16_t v) [inline]`

8.29.2.23 `int16_t htobe (int16_t v) [inline]`

8.29.2.24 `uint32_t htobe (uint32_t v) [inline]`

- 8.29.2.25 `int32_t htogle (int32_t v) [inline]`
- 8.29.2.26 `uint64_t htogle (uint64_t v) [inline]`
- 8.29.2.27 `int64_t htogle (int64_t v) [inline]`
- 8.29.2.28 `double htogle (double v) [inline]`
- 8.29.2.29 `float htogle (float v) [inline]`
- 8.29.2.30 `uint16_t leth (uint16_t v) [inline]`
- 8.29.2.31 `int16_t leth (int16_t v) [inline]`
- 8.29.2.32 `uint32_t leth (uint32_t v) [inline]`
- 8.29.2.33 `int32_t leth (int32_t v) [inline]`
- 8.29.2.34 `uint64_t leth (uint64_t v) [inline]`
- 8.29.2.35 `int64_t leth (int64_t v) [inline]`
- 8.29.2.36 `double leth (double v) [inline]`
- 8.29.2.37 `float leth (float v) [inline]`

8.30 /src/cern/tms/beam/libBeam/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>
```

8.31 /src/cern/tms/beam/libBeam/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.

8.32 /src/cern/tms/beam/libBeam/BError.cpp File Reference

```
#include <BError.h>
```

8.33 /src/cern/tms/beam/libBeam/BError.h File Reference

```
#include <BString.h>
```

Classes

- class [BError](#)
Error return class.

Enumerations

- enum [BErrorNum](#) {
 [ErrorOk](#) = 0, [ErrorMisc](#) = 1, [ErrorWarning](#) = 2, [ErrorParam](#) = 3,
 [ErrorTimeout](#) = 4, [ErrorNotAvailable](#) = 5, [ErrorData](#) = 6, [ErrorChecksum](#) = 7,
 [ErrorOverrun](#) = 8, [ErrorUnderrun](#) = 9, [ErrorInit](#) = 10, [ErrorConfig](#) = 11,
 [ErrorNotImplemented](#) = 12, [ErrorResourceLimit](#) = 13, [ErrorEndOfFile](#) = 14, [ErrorFile](#) = 15,
 [ErrorFormat](#) = 16, [ErrorComms](#) = 17, [ErrorAccessDenied](#) = 18, [ErrorNoData](#) = 19,
 [ErrorEndOfData](#) = 20, [ErrorDataPresent](#) = 21, [ErrorAppBase](#) = 64 }

8.33.1 Enumeration Type Documentation

8.33.1.1 enum BErrorNum

Enumerator

ErrorOk

ErrorMisc

ErrorWarning

ErrorParam

ErrorTimeout

ErrorNotAvailable

ErrorData

ErrorChecksum

ErrorOverrun

ErrorUnderrun

ErrorInit

ErrorConfig

ErrorNotImplemented

ErrorResourceLimit

ErrorEndOfFile

ErrorFile

ErrorFormat

ErrorComms

ErrorAccessDenied
ErrorNoData
ErrorEndOfData
ErrorDataPresent
ErrorAppBase

8.34 /src/cern/tms/beam/libBeam/BErrorTime.cpp File Reference

```
#include <BErrorTime.h>
```

8.35 /src/cern/tms/beam/libBeam/BErrorTime.h File Reference

```
#include <BString.h>  
#include <BTimeStamp.h>
```

Classes

- class [BErrorTime](#)
Error return class.

8.36 /src/cern/tms/beam/libBeam/BEvent.cpp File Reference

```
#include <BEvent.h>  
#include <BPoll.h>  
#include <stdlib.h>  
#include <unistd.h>  
#include <sys/ioctl.h>
```

8.37 /src/cern/tms/beam/libBeam/BEvent.h File Reference

```
#include <BTypes.h>  
#include <BQueue.h>
```

Classes

- class [BEvent](#)
- class [BEventPipe](#)
This class provides an interface for sending simple integer events via a pipe file descriptor.

Typedefs

- typedef [BQueue< BEvent > BEventQueue](#)
This class provides an interface for sending simple integer events via a [BQueue](#).

Enumerations

- enum [BEventType](#) { [BEventTypeNone](#) = 0 }

8.37.1 Typedef Documentation

8.37.1.1 typedef [BQueue](#)<[BEvent](#)> [BEventQueue](#)

This class provides an interface for sending simple integer events via a [BQueue](#).

8.37.2 Enumeration Type Documentation

8.37.2.1 enum [BEventType](#)

Enumerator

[BEventTypeNone](#)

8.38 /src/cern/tms/beam/libBeam/BEvent1.cpp File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <BEvent1.h>
#include <BPoll.h>
```

8.39 /src/cern/tms/beam/libBeam/BEvent1.h File Reference

```
#include <stdint.h>
#include <BError.h>
```

Classes

- class [BEvent1](#)

This class provides a base class for all event objects that can be sent over the events interface.

- class [BEvent1Error](#)
- class [BEvent1Pipe](#)

This class provides a base interface for sending events via a pipe. This allows threads to send events that can be picked up by the poll system call.

- class [BEvent1Int](#)

This class provides an interface for sending simple integer events via a file descriptor. This allows threads to send events that can be picked up by the poll system call.

Enumerations

- enum [BEvent1Type](#) { [BEvent1TypeNone](#), [BEvent1TypeInt](#), [BEvent1TypeError](#) }

8.39.1 Enumeration Type Documentation

8.39.1.1 enum BEvent1Type

Enumerator

BEvent1TypeNone

BEvent1TypeInt

BEvent1TypeError

8.40 /src/cern/tms/beam/libBeam/BFifo.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BMutex.h>
#include <BFifo.inc>
```

Classes

- class [BFifo< Type >](#)

8.41 /src/cern/tms/beam/libBeam/BFifo.inc File Reference

8.42 /src/cern/tms/beam/libBeam/BFifoCirc.cpp File Reference

```
#include <BFifoCirc.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/mman.h>
```

Macros

- #define [dprintf\(fmt, a...\)](#)

8.42.1 Macro Definition Documentation

8.42.1.1 #define dprintf(*fmt*, *a...*)

8.43 /src/cern/tms/beam/libBeam/BFifoCirc.h File Reference

```
#include <stdint.h>
#include <BError.h>
#include <BCondInt.h>
#include <BMutex.h>
#include <BFifoCirc.inc>
```

Classes

- class [BFifoCircPos](#)
This class implements a pointer into the Fifo's circular buffer.
- class [BFifoCirc< Type >](#)
This class implements a thread safe FIFO buffer.

8.44 /src/cern/tms/beam/libBeam/BFifoCirc.inc File Reference

8.45 /src/cern/tms/beam/libBeam/BFile.cpp File Reference

```
#include <stdarg.h>
#include <BFile.h>
#include <sys/stat.h>
#include <string.h>
#include <errno.h>
#include <unistd.h>
```

Macros

- #define [STRBUF](#) 10240

8.45.1 Macro Definition Documentation

8.45.1.1 #define STRBUF 10240

8.46 /src/cern/tms/beam/libBeam/BFile.h File Reference

```
#include <stdio.h>
#include <BTypes.h>
#include <BString.h>
#include <BError.h>
```

Classes

- class [BFile](#)
File operations class.

8.47 /src/cern/tms/beam/libBeam/BFileCsv.cpp File Reference

```
#include <BFileCsv.h>
#include <errno.h>
```

8.48 /src/cern/tms/beam/libBeam/BFileCsv.h File Reference

```
#include <BFile.h>
```

Classes

- class [BFileCsv](#)

8.49 /src/cern/tms/beam/libBeam/BFileData.cpp File Reference

```
#include <BFileCsv.h>  
#include <BFileData.h>  
#include <errno.h>
```

8.50 /src/cern/tms/beam/libBeam/BFileData.h File Reference

```
#include <BError.h>
```

Classes

- class [BFileData](#)

8.51 /src/cern/tms/beam/libBeam/BList.h File Reference

```
#include <BList_func.h>
```

Classes

- class [BNode](#)
- class [BIter](#)
Iterator for [BList](#).
- class [BList< T >](#)
Template based list class.
- class [BList< T >::Node](#)

Macros

- `#define BListLoop(list, i) for(BIter i = list.begin(); !list.isEnd(i); list.next(i))`

8.51.1 Macro Definition Documentation

8.51.1.1 `#define BListLoop(list, i) for(BIter i = list.begin(); !list.isEnd(i); list.next(i))`

8.52 /src/cern/tms/beam/libBeam/BList_func.h File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <memory.h>
```

8.53 /src/cern/tms/beam/libBeam/BMutex.cpp File Reference

```
#include <BMutex.h>
```

Macros

- `#define MDEBUG 0`

8.53.1 Macro Definition Documentation

8.53.1.1 `#define MDEBUG 0`

8.54 /src/cern/tms/beam/libBeam/BMutex.h File Reference

```
#include <pthread.h>
```

Classes

- class [BMutex](#)
Mutex class.
- class [BMutexLock](#)

8.55 /src/cern/tms/beam/libBeam/BMysql.cpp File Reference

```
#include <stdlib.h>
#include <string.h>
#include <Bmysql.h>
```

8.56 /src/cern/tms/beam/libBeam/BMysql.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BDict.h>
#include <BMutex.h>
#include <mysql/mysql.h>
```

Classes

- class [BMySQL](#)

8.57 /src/cern/tms/beam/libBeam/BNameValue.h File Reference

```
#include <BList.h>
#include <BString.h>
```

Classes

- class [BNameValue< T >](#)
- class [BNameValueList< T >](#)

8.58 /src/cern/tms/beam/libBeam/Boap.cpp File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <unistd.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
#include <Boap.h>
#include <byteswap.h>
#include <BoapnsD.h>
#include <BoapnsC.h>
```

Macros

- `#define DEBUG 0`
- `#define APIVERSION_TEST 1`
- `#define dprintf(fmt, a...)`
- `#define IS_BIG_ENDIAN 1`

Variables

- const int [boapPort](#) = 12000
The default BOAP connection port.

8.58.1 Macro Definition Documentation

8.58.1.1 `#define APIVERSION_TEST 1`

8.58.1.2 `#define DEBUG 0`

8.58.1.3 `#define dprintf(fmt, a...)`

8.58.1.4 `#define IS_BIG_ENDIAN 1`

8.58.2 Variable Documentation

8.58.2.1 `const int boapPort = 12000`

The default BOAP connection port.

8.59 /src/cern/tms/beam/libBeam/Boap.h File Reference

```
#include <stdint.h>
#include <BTypes.h>
#include <BPoll.h>
#include <BSocket.h>
#include <BThread.h>
#include <BError.h>
#include <BEvent1.h>
#include <BMutex.h>
#include <BTimeStamp.h>
#include <BBuffer.h>
```

Classes

- struct [BoapPacketHead](#)
- class [BoapPacket](#)
- class [BoapClientObject](#)
- class [BoapSignalObject](#)
- class [BoapServiceEntry](#)
- class [BoapServerConnection](#)
- class [BoapServer](#)
- class [BoapFuncEntry](#)
- class [BoapServiceObject](#)

Namespaces

- [Boapns](#)

Typedefs

- typedef [BUInt32](#) [BoapService](#)
- typedef [BError](#)([BoapServiceObject](#)::* [BoapFunc](#))([BoapServerConnection](#) *conn, [BoapPacket](#) &rx, [BoapPacket](#) &tx)

Enumerations

- enum [BoapType](#) {
 [BoapTypeRpc](#), [BoapTypeRpcReply](#), [BoapTypeSignal](#), [BoapTypeRpcError](#),
 [BoapTypeRpc](#), [BoapTypeSignal](#) }
- enum [BoapPriority](#) { [BoapPriorityLow](#), [BoapPriorityNormal](#), [BoapPriorityHigh](#) }

Variables

- const `BUInt32 BoapMagic` = 0x424F4100

8.59.1 Typedef Documentation

8.59.1.1 typedef `BError(BoapServiceObject::* BoapFunc)(BoapServerConnection *conn, BoapPacket &rx, BoapPacket &tx)`

8.59.1.2 typedef `BUInt32 BoapService`

8.59.2 Enumeration Type Documentation

8.59.2.1 enum `BoapPriority`

Enumerator

BoapPriorityLow
BoapPriorityNormal
BoapPriorityHigh

8.59.2.2 enum `BoapType`

Enumerator

BoapTypeRpc
BoapTypeRpcReply
BoapTypeSignal
BoapTypeRpcError
BoapTypeRpc
BoapTypeSignal

8.59.3 Variable Documentation

8.59.3.1 const `BUInt32 BoapMagic` = 0x424F4100

8.60 /src/cern/tms/beam/libBeam/BoapMc.cpp File Reference

```
#include <BoapMc.h>
#include <BCrc16.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
```

Macros

- #define `DEBUG_LOCAL` 0
- #define `DEBUG_LOCAL1` 0
- #define `dlprintf`(fmt, a...)
- #define `dl1printf`(fmt, a...)

8.60.1 Macro Definition Documentation

8.60.1.1 `#define DEBUG_LOCAL 0`

8.60.1.2 `#define DEBUG_LOCAL1 0`

8.60.1.3 `#define dl1printf(fmt, a...)`

8.60.1.4 `#define dlprintf(fmt, a...)`

8.61 /src/cern/tms/beam/libBeam/BoapMc.h File Reference

```
#include <BTypes.h>
#include <BMutex.h>
#include <BSemaphore.h>
#include <BQueue.h>
#include <BFifo.h>
#include <BComms.h>
```

Classes

- struct [BoapMcPacketHead](#)
- class [BoapMcPacket](#)
- class [BoapMcClientObject](#)
- class [BoapMcSignalObject](#)
- class [BoapMcServiceObject](#)
- class [BoapMcComms](#)

Enumerations

- enum [BoapMcType](#) { [BoapMcTypeRequest](#) = 0x00, [BoapMcTypeReply](#) = 0x80 }

Functions

- struct [BoapMcPacketHead](#) `__attribute__((aligned(8), packed))`

Variables

- [BUInt8](#) `length`
- [BUInt8](#) `addressTo`
- [BUInt8](#) `addressFrom`
- [BUInt8](#) `cmd`
- [BUInt16](#) `error`
- [BUInt16](#) `checksum`
- class [BoapMcPacket](#) `__attribute__((aligned(8), packed))`

8.61.1 Enumeration Type Documentation

8.61.1.1 enum [BoapMcType](#)

Enumerator

BoapMcTypeRequest

BoapMcTypeReply

8.61.2 Function Documentation

8.61.2.1 struct BoapMcPacketHead __attribute__ ((aligned(8), packed))

8.61.3 Variable Documentation

8.61.3.1 class BoapMcPacket __attribute__

8.61.3.2 BUInt8 addressFrom

8.61.3.3 BUInt8 addressTo

8.61.3.4 BUInt16 checksum

8.61.3.5 BUInt8 cmd

8.61.3.6 BUInt16 error

8.61.3.7 BUInt8 length

8.62 /src/cern/tms/beam/libBeam/BoapnsC.cpp File Reference

```
#include <BoapnsC.h>
```

Namespaces

- [Boapns](#)

8.63 /src/cern/tms/beam/libBeam/BoapnsC.h File Reference

```
#include <stdlib.h>  
#include <stdint.h>  
#include <Boap.h>  
#include <BString.h>  
#include <BList.h>  
#include <BArray.h>  
#include <BoapnsD.h>
```

Classes

- class [Boapns::Boapns](#)

Namespaces

- [Boapns](#)

Variables

- const [BUInt32 Boapns::apiVersion](#) = 0

8.64 /src/cern/tms/beam/libBeam/BoapnsD.cpp File Reference

```
#include <BoapnsD.h>
```

Namespaces

- [Boapns](#)

8.65 /src/cern/tms/beam/libBeam/BoapnsD.h File Reference

```
#include <Boap.h>
#include <BObj.h>
#include <BDate.h>
#include <BTimeStamp.h>
#include <BComplex.h>
#include <BList.h>
#include <BArray.h>
```

Classes

- class [Boapns::BoapEntry](#)

Namespaces

- [Boapns](#)

8.66 /src/cern/tms/beam/libBeam/BoapSimple.cc File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <errno.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <Boap.h>
#include <BoapnsD.h>
#include <BoapnsC.h>
```

Macros

- #define [DEBUG](#) 0
- #define [dprintf](#)(fmt, a...)

Variables

- const int [roundSize](#) = 256

8.66.1 Macro Definition Documentation

8.66.1.1 `#define DEBUG 0`

8.66.1.2 `#define dprintf(fmt, a...)`

8.66.2 Variable Documentation

8.66.2.1 `const int roundSize = 256`

8.67 /src/cern/tms/beam/libBeam/BoapSimple.h File Reference

```
#include <stdint.h>
#include <BPoll.h>
#include <BSocket.h>
#include <BError.h>
```

Classes

- struct [BoapPacketHead](#)
- class [BoapPacket](#)
- class [BoapClientObject](#)
- class [BoapSignalObject](#)
- class [BoapServiceEntry](#)
- class [BoapServer](#)
- class [BoapFuncEntry](#)
- class [BoapServiceObject](#)

Typedefs

- typedef int8_t [Int8](#)
- typedef uint8_t [UInt8](#)
- typedef int16_t [Int16](#)
- typedef uint16_t [UInt16](#)
- typedef int32_t [Int32](#)
- typedef uint32_t [UInt32](#)
- typedef double [Double](#)
- typedef uint32_t [BoapService](#)
- typedef [BError](#)([BoapServiceObject](#)::* [BoapFunc](#))([BoapPacket](#) &rx, [BoapPacket](#) &tx)

Enumerations

- enum [BoapType](#) {
[BoapTypeRpc](#), [BoapTypeRpcReply](#), [BoapTypeSignal](#), [BoapTypeRpcError](#),
[BoapTypeRpc](#), [BoapTypeSignal](#) }

8.67.1 Typedef Documentation

8.67.1.1 typedef BError(BoapServiceObject::* BoapFunc)(BoapPacket &rx, BoapPacket &tx)

8.67.1.2 typedef uint32_t BoapService

8.67.1.3 typedef double Double

8.67.1.4 typedef int16_t Int16

8.67.1.5 typedef int32_t Int32

8.67.1.6 typedef int8_t Int8

8.67.1.7 typedef uint16_t UInt16

8.67.1.8 typedef uint32_t UInt32

8.67.1.9 typedef uint8_t UInt8

8.67.2 Enumeration Type Documentation

8.67.2.1 enum BoapType

Enumerator

BoapTypeRpc

BoapTypeRpcReply

BoapTypeSignal

BoapTypeRpcError

BoapTypeRpc

BoapTypeSignal

8.68 /src/cern/tms/beam/libBeam/BObj.cpp File Reference

```
#include <BObj.h>
```

8.69 /src/cern/tms/beam/libBeam/BObj.h File Reference

```
#include <BTypes.h>  
#include <BDict.h>  
#include <BString.h>  
#include <BError.h>
```

Classes

- class [BObj](#)

8.70 /src/cern/tms/beam/libBeam/BObjStringFormat.cpp File Reference

```
#include <BObjStringFormat.h>
#include <BTime.h>
#include <math.h>
```

Functions

- [BString toBString \(BString n, Bool v\)](#)
- [BString toBString \(BString n, BInt8 v\)](#)
- [BString toBString \(BString n, BUInt8 v\)](#)
- [BString toBString \(BString n, BInt16 v\)](#)
- [BString toBString \(BString n, BUInt16 v\)](#)
- [BString toBString \(BString n, BInt32 v\)](#)
- [BString toBString \(BString n, BUInt32 v\)](#)
- [BString toBString \(BString n, BInt64 v\)](#)
- [BString toBString \(BString n, BUInt64 v\)](#)
- [BString toBString \(BString n, BFloat32 v\)](#)
- [BString toBString \(BString n, BFloat64 v\)](#)
- [BString toBString \(BString n, BChar v\)](#)
- [BString toBString \(BString n, const BChar *v\)](#)
- [BString toBString \(BString n, BString v\)](#)
- [BString toBString \(BString n, BError v\)](#)
- [BString toBString \(BString n, BTime v\)](#)
- [BString toBString \(BString name, const BObjMember *m, const void *obj, BStringList ignoreFields\)](#)
- [BString toBString \(BString n, BObj &obj\)](#)
- [BString toBStringJson \(BString n, Bool v\)](#)
- [BString toBStringJson \(BString n, BInt8 v\)](#)
- [BString toBStringJson \(BString n, BUInt8 v\)](#)
- [BString toBStringJson \(BString n, BInt16 v\)](#)
- [BString toBStringJson \(BString n, BUInt16 v\)](#)
- [BString toBStringJson \(BString n, BInt32 v\)](#)
- [BString toBStringJson \(BString n, BUInt32 v\)](#)
- [BString toBStringJson \(BString n, BInt64 v\)](#)
- [BString toBStringJson \(BString n, BUInt64 v\)](#)
- [BString toBStringJson \(BString n, BFloat32 v\)](#)
- [BString toBStringJson \(BString n, BFloat64 v\)](#)
- [BString toBStringJson \(BString n, BChar v\)](#)
- [BString toBStringJson \(BString n, const BChar *v\)](#)
- [BString toBStringJson \(BString n, BString v\)](#)
- [BString toBStringJson \(BString n, BError v\)](#)
- [BString toBStringJson \(BString n, BTime v\)](#)
- [BString toBStringJson \(BString n, const BObjMember *m, const void *obj, BStringList ignoreFields\)](#)
- [BString toBStringJson \(BString n, BObj &obj\)](#)
- [BError toBDictStringFromJson \(BString json, BDictString &ds\)](#)

8.70.1 Function Documentation

- 8.70.1.1 `BError toBDictStringFromJson (BString json, BDictString & ds)`
- 8.70.1.2 `BString toBString (BString n, Bool v)`
- 8.70.1.3 `BString toBString (BString n, BInt8 v)`
- 8.70.1.4 `BString toBString (BString n, BUInt8 v)`
- 8.70.1.5 `BString toBString (BString n, BInt16 v)`
- 8.70.1.6 `BString toBString (BString n, BUInt16 v)`
- 8.70.1.7 `BString toBString (BString n, BInt32 v)`
- 8.70.1.8 `BString toBString (BString n, BUInt32 v)`
- 8.70.1.9 `BString toBString (BString n, BInt64 v)`
- 8.70.1.10 `BString toBString (BString n, BUInt64 v)`
- 8.70.1.11 `BString toBString (BString n, BFloat32 v)`
- 8.70.1.12 `BString toBString (BString n, BFloat64 v)`
- 8.70.1.13 `BString toBString (BString n, BChar v)`
- 8.70.1.14 `BString toBString (BString n, const BChar * v)`
- 8.70.1.15 `BString toBString (BString n, BString v)`
- 8.70.1.16 `BString toBString (BString n, BError v)`
- 8.70.1.17 `BString toBString (BString n, BTime v)`
- 8.70.1.18 `BString toBString (BString name, const BObjMember * m, const void * obj, BStringList ignoreFields)`
- 8.70.1.19 `BString toBString (BString n, BObj & obj)`
- 8.70.1.20 `BString toBStringJson (BString n, Bool v)`
- 8.70.1.21 `BString toBStringJson (BString n, BInt8 v)`
- 8.70.1.22 `BString toBStringJson (BString n, BUInt8 v)`
- 8.70.1.23 `BString toBStringJson (BString n, BInt16 v)`
- 8.70.1.24 `BString toBStringJson (BString n, BUInt16 v)`
- 8.70.1.25 `BString toBStringJson (BString n, BInt32 v)`
- 8.70.1.26 `BString toBStringJson (BString n, BUInt32 v)`
- 8.70.1.27 `BString toBStringJson (BString n, BInt64 v)`

- 8.70.1.28 `BString toBStringJson (BString n, BUInt64 v)`
- 8.70.1.29 `BString toBStringJson (BString n, BFloat32 v)`
- 8.70.1.30 `BString toBStringJson (BString n, BFloat64 v)`
- 8.70.1.31 `BString toBStringJson (BString n, BChar v)`
- 8.70.1.32 `BString toBStringJson (BString n, const BChar * v)`
- 8.70.1.33 `BString toBStringJson (BString n, BString v)`
- 8.70.1.34 `BString toBStringJson (BString n, BError v)`
- 8.70.1.35 `BString toBStringJson (BString n, BTime v)`
- 8.70.1.36 `BString toBStringJson (BString n, const BObjMember * m, const void * obj, BStringList ignoreFields)`
- 8.70.1.37 `BString toBStringJson (BString n, BObj & obj)`

8.71 /src/cern/tms/beam/libBeam/BObjStringFormat.h File Reference

```
#include <BObj.h>
#include <BString.h>
#include <BTime.h>
```

Functions

- `BString toBString (BString name, Bool value)`
- `BString toBString (BString name, BInt8 value)`
- `BString toBString (BString name, BUInt8 value)`
- `BString toBString (BString name, BInt16 value)`
- `BString toBString (BString name, BUInt16 value)`
- `BString toBString (BString name, BInt32 value)`
- `BString toBString (BString name, BUInt32 value)`
- `BString toBString (BString name, BFloat32 value)`
- `BString toBString (BString name, BFloat64 value)`
- `BString toBString (BString name, BChar value)`
- `BString toBString (BString name, const BChar *value)`
- `BString toBString (BString name, BString value)`
- `BString toBString (BString name, BError value)`
- `BString toBString (BString name, BTime time)`
- `BString toBString (BString name, const BObjMember *members, const void *obj, BStringList ignoreFields=BStringList())`
- `BString toBString (BString name, BObj &obj)`
- `BString toBStringJson (BString name, Bool value)`
- `BString toBStringJson (BString name, BInt8 value)`
- `BString toBStringJson (BString name, BUInt8 value)`
- `BString toBStringJson (BString name, BInt16 value)`
- `BString toBStringJson (BString name, BUInt16 value)`
- `BString toBStringJson (BString name, BInt32 value)`
- `BString toBStringJson (BString name, BUInt32 value)`
- `BString toBStringJson (BString name, BFloat32 value)`

- [BString toBStringJson](#) (BString name, BFloat64 value)
- [BString toBStringJson](#) (BString name, BChar value)
- [BString toBStringJson](#) (BString name, const BChar *value)
- [BString toBStringJson](#) (BString name, BString value)
- [BString toBStringJson](#) (BString name, BError value)
- [BString toBStringJson](#) (BString name, BTime time)
- [BString toBStringJson](#) (BString name, const BObjMember *members, const void *obj, BStringList ignoreFields=BStringList())
- [BString toBStringJson](#) (BString name, BObj &obj)
- [BError toBDictStringFromJson](#) (BString json, BDictString &ds)
- [BString base64_encode](#) (void *data, BUInt len)
- [BError base64_decode](#) (BString strIn, BString &strOut)

8.71.1 Function Documentation

8.71.1.1 [BError base64_decode](#) (BString *strIn*, BString & *strOut*)

8.71.1.2 [BString base64_encode](#) (void * *data*, BUInt *len*)

8.71.1.3 [BError toBDictStringFromJson](#) (BString *json*, BDictString & *ds*)

8.71.1.4 [BString toBString](#) (BString *name*, Bool *value*)

8.71.1.5 [BString toBString](#) (BString *name*, BInt8 *value*)

8.71.1.6 [BString toBString](#) (BString *name*, BUInt8 *value*)

8.71.1.7 [BString toBString](#) (BString *name*, BInt16 *value*)

8.71.1.8 [BString toBString](#) (BString *name*, BUInt16 *value*)

8.71.1.9 [BString toBString](#) (BString *name*, BInt32 *value*)

8.71.1.10 [BString toBString](#) (BString *name*, BUInt32 *value*)

8.71.1.11 [BString toBString](#) (BString *name*, BFloat32 *value*)

8.71.1.12 [BString toBString](#) (BString *name*, BFloat64 *value*)

8.71.1.13 [BString toBString](#) (BString *name*, BChar *value*)

8.71.1.14 [BString toBString](#) (BString *name*, const BChar * *value*)

8.71.1.15 [BString toBString](#) (BString *name*, BString *value*)

8.71.1.16 [BString toBString](#) (BString *name*, BError *value*)

8.71.1.17 [BString toBString](#) (BString *name*, BTime *time*)

8.71.1.18 [BString toBString](#) (BString *name*, const BObjMember * *members*, const void * *obj*, BStringList *ignoreFields* =BStringList())

8.71.1.19 [BString toBString](#) (BString *name*, BObj & *obj*)

8.71.1.20 [BString toBStringJson](#) (BString *name*, Bool *value*)

- 8.71.1.21 BString toBStringJson (BString name, BInt8 value)
- 8.71.1.22 BString toBStringJson (BString name, BUInt8 value)
- 8.71.1.23 BString toBStringJson (BString name, BInt16 value)
- 8.71.1.24 BString toBStringJson (BString name, BUInt16 value)
- 8.71.1.25 BString toBStringJson (BString name, BInt32 value)
- 8.71.1.26 BString toBStringJson (BString name, BUInt32 value)
- 8.71.1.27 BString toBStringJson (BString name, BFloat32 value)
- 8.71.1.28 BString toBStringJson (BString name, BFloat64 value)
- 8.71.1.29 BString toBStringJson (BString name, BChar value)
- 8.71.1.30 BString toBStringJson (BString name, const BChar * value)
- 8.71.1.31 BString toBStringJson (BString name, BString value)
- 8.71.1.32 BString toBStringJson (BString name, BError value)
- 8.71.1.33 BString toBStringJson (BString name, BTime time)
- 8.71.1.34 BString toBStringJson (BString name, const BObjMember * members, const void * obj, BStringList ignoreFields = BStringList())
- 8.71.1.35 BString toBStringJson (BString name, BObj & obj)

8.72 /src/cern/tms/beam/libBeam/BPoll.cpp File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <BPoll.h>
```

8.73 /src/cern/tms/beam/libBeam/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.

8.74 /src/cern/tms/beam/libBeam/BQueue.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BList.h>
#include <BMutex.h>
#include <BCondInt.h>
```

Classes

- class [BQueue< T >](#)
Queue class.

Typedefs

- typedef [BQueue< BInt32 >](#) [BQueueInt](#)

8.74.1 Typedef Documentation

8.74.1.1 typedef [BQueue<BInt32>](#) [BQueueInt](#)

8.75 /src/cern/tms/beam/libBeam/BRefData.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <BRefData.h>
```

Macros

- #define [CHUNK](#) 16

8.75.1 Macro Definition Documentation

8.75.1.1 #define [CHUNK](#) 16

8.76 /src/cern/tms/beam/libBeam/BRefData.h File Reference

```
#include <BAAtomicCount.h>
```

Classes

- class [BRefData](#)

8.77 /src/cern/tms/beam/libBeam/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>
```

8.78 /src/cern/tms/beam/libBeam/BRtc.h File Reference

```
#include <BError.h>
#include <BThread.h>
#include <BCond.h>
```

Classes

- class [BRtc](#)
Realtime clock.
- class [BRtcThreaded](#)
Threaded real time clock.

8.79 /src/cern/tms/beam/libBeam/BRWLock.cpp File Reference

```
#include <BRWLock.h>
```

8.80 /src/cern/tms/beam/libBeam/BRWLock.h File Reference

```
#include <pthread.h>
```

Classes

- class [BRWLock](#)
thread read-write locks

8.81 /src/cern/tms/beam/libBeam/BSema.cpp File Reference

```
#include <BSema.h>
#include <errno.h>
#include <sys/time.h>
```

8.82 /src/cern/tms/beam/libBeam/BSema.h File Reference

```
#include <sys/types.h>
#include <semaphore.h>
```

Classes

- class [BSema](#)
Sempahore class.

8.83 /src/cern/tms/beam/libBeam/BSemaphore.cpp File Reference

```
#include <BSemaphore.h>
#include <sys/time.h>
```

8.84 /src/cern/tms/beam/libBeam/BSemaphore.h File Reference

```
#include <BTypes.h>
#include <BMutex.h>
#include <semaphore.h>
```

Classes

- class [BSemaphore](#)
Semaphore class.
- class [BSemaphoreCount](#)

8.85 /src/cern/tms/beam/libBeam/BSocket.cpp File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <net/if.h>
#include "BSocket.h"
```

Macros

- #define [IP_MTU](#) 14

8.85.1 Macro Definition Documentation

8.85.1.1 #define IP_MTU 14

8.86 /src/cern/tms/beam/libBeam/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](#)

8.87 /src/cern/tms/beam/libBeam/BSpi.cpp File Reference

```
#include <BSpi.h>
#include <fcntl.h>
#include <errno.h>
#include <sys/ioctl.h>
#include <linux/spi/spidev.h>
```

8.88 /src/cern/tms/beam/libBeam/BSpi.h File Reference

```
#include <BTypes.h>
#include <BError.h>
```

Classes

- class [BSpi](#)
BSpi class.

8.89 /src/cern/tms/beam/libBeam/BString.cpp File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdarg.h>
#include <ctype.h>
#include <BString.h>
#include <BError.h>
#include <regex.h>
```

Macros

- `#define STRIP 0x7f`
- `#define MINUS '-'`

Functions

- static int `gmatch` (const char *s, const char *p)
- std::ostream & `operator<<` (std::ostream &o, BString &s)
- std::istream & `operator>>` (std::istream &i, BString &s)
- int `bstringListinList` (BStringList &list, BString s)
- BString `blistToString` (const BStringList &list)
- BStringList `bstringToList` (BString str, int stripSpaces)
- BStringList `charToList` (const char **str)
- BString `barrayToString` (const BStringArray &list)
- BStringArray `bstringToArray` (BString str, int stripSpaces)
- BStringArray `charToArray` (const char **str)
- void `toBString` (BString &v, BString &s)
- void `toBString` (BStringList &v, BString &s)
- void `toBString` (BInt32 &v, BString &s)
- void `toBString` (BUInt32 &v, BString &s)
- void `toBString` (BUInt64 &v, BString &s)
- void `toBString` (BFloat64 &v, BString &s)
- void `fromBString` (BString &s, BString &v)
- void `fromBString` (BString &s, BStringList &v)
- void `fromBString` (BString &s, BInt32 &v)
- void `fromBString` (BString &s, BUInt32 &v)
- void `fromBString` (BString &s, BUInt64 &v)
- void `fromBString` (BString &s, BFloat64 &v)

Variables

- static const BUInt8 `base64_decode_table` []

8.89.1 Macro Definition Documentation

8.89.1.1 `#define MINUS '-'`

8.89.1.2 `#define STRIP 0x7f`

8.90.1.9 void toBString (BString & v, BString & s)

8.90.1.10 void toBString (BStringList & v, BString & s)

8.90.1.11 void toBString (BInt32 & v, BString & s)

8.90.1.12 void toBString (BUInt32 & v, BString & s)

8.90.1.13 void toBString (BUInt64 & v, BString & s)

8.90.1.14 void toBString (BFloat64 & v, BString & s)

8.91 /src/cern/tms/beam/libBeam/BStringLocked.h File Reference

```
#include <BString.h>
#include <BMutex.h>
```

Classes

- class [BStringMutex](#)
- class [BStringLocked](#)

8.92 /src/cern/tms/beam/libBeam/BTable.cpp File Reference

```
#include <BTable.h>
```

8.93 /src/cern/tms/beam/libBeam/BTable.h File Reference

```
#include <BArray.h>
#include <BString.h>
```

Classes

- class [BTable](#)

8.94 /src/cern/tms/beam/libBeam/BThread.cpp File Reference

```
#include <BThread.h>
#include <unistd.h>
#include <errno.h>
#include <sys/types.h>
```

8.95 /src/cern/tms/beam/libBeam/BThread.h File Reference

```
#include <pthread.h>
```

Classes

- class [BThread](#)

8.96 /src/cern/tms/beam/libBeam/BTime.cpp File Reference

```
#include <BTime.h>
```

Functions

- static bool [yearIsLeap](#) ([BUInt16](#) year)
- static [BUInt16](#) [yearDays](#) ([BUInt16](#) year)

Variables

- static [BUInt16](#) [monDays](#) [2][13]

8.96.1 Function Documentation

8.96.1.1 static [BUInt16](#) [yearDays](#) ([BUInt16](#) year) [inline],[static]

8.96.1.2 static bool [yearIsLeap](#) ([BUInt16](#) year) [inline],[static]

8.96.2 Variable Documentation

8.96.2.1 [BUInt16](#) [monDays](#)[2][13] [static]

Initial value:

```
= {  
    { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 },  
    { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366 }  
}
```

8.97 /src/cern/tms/beam/libBeam/BTime.h File Reference

```
#include <BTypes.h>  
#include <BError.h>  
#include <BString.h>
```

Classes

- class [BTime](#)

8.98 /src/cern/tms/beam/libBeam/BTimer.cpp File Reference

```
#include <BTimer.h>
#include <sys/time.h>
```

8.99 /src/cern/tms/beam/libBeam/BTimer.h File Reference

```
#include <BMutex.h>
```

Classes

- class [BTimer](#)
Stopwatch style timer.

8.100 /src/cern/tms/beam/libBeam/BTimeStamp.cpp File Reference

```
#include <BTimeStamp.h>
#include <BTimeStampMs.h>
#include <sys/time.h>
```

Functions

- void [toBString](#) ([BTimeStamp](#) &*v*, [BString](#) &*s*)
- void [fromBString](#) ([BString](#) &*s*, [BTimeStamp](#) &*v*)

Variables

- static int [mon_yday](#) [2][13]

8.100.1 Function Documentation

8.100.1.1 void [fromBString](#) ([BString](#) & *s*, [BTimeStamp](#) & *v*)

8.100.1.2 void [toBString](#) ([BTimeStamp](#) & *v*, [BString](#) & *s*)

8.100.2 Variable Documentation

8.100.2.1 int [mon_yday](#)[2][13] [*static*]

Initial value:

```
= {
    { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 },
    { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366 }
}
```

8.101 /src/cern/tms/beam/libBeam/BTimeStamp.h File Reference

```
#include <stdint.h>
#include <BError.h>
```

Classes

- class [BTimeStamp](#)

Functions

- void [toBString](#) ([BTimeStamp](#) &v, [BString](#) &s)
- void [fromBString](#) ([BString](#) &s, [BTimeStamp](#) &v)

8.101.1 Function Documentation

8.101.1.1 void [fromBString](#) ([BString](#) & s, [BTimeStamp](#) & v)

8.101.1.2 void [toBString](#) ([BTimeStamp](#) & v, [BString](#) & s)

8.102 /src/cern/tms/beam/libBeam/BTimeStampMs.cpp File Reference

```
#include <BTimeStampMs.h>
#include <sys/time.h>
```

Variables

- static int [mon_yday](#) [2][13]

8.102.1 Variable Documentation

8.102.1.1 int [mon_yday](#)[2][13] [[static](#)]

Initial value:

```
= {
    { 0, 31, 59, 90, 120, 151, 181, 212, 243, 273, 304, 334, 365 },
    { 0, 31, 60, 91, 121, 152, 182, 213, 244, 274, 305, 335, 366 }
}
```

8.103 /src/cern/tms/beam/libBeam/BTimeStampMs.h File Reference

```
#include <stdint.h>
#include <BError.h>
```

Classes

- class [BTimeStampMs](#)

8.104 /src/cern/tms/beam/libBeam/BTypes.h File Reference

```
#include <stdint.h>
#include <sys/types.h>
#include <vector>
```

Classes

- struct [BObjMember](#)

Typedefs

- typedef bool [Bool](#)
- typedef int8_t [BInt8](#)
- typedef uint8_t [BUInt8](#)
- typedef int16_t [BInt16](#)
- typedef uint16_t [BUInt16](#)
- typedef int32_t [BInt32](#)
- typedef uint32_t [BUInt32](#)
- typedef int64_t [BInt64](#)
- typedef uint64_t [BUInt64](#)
- typedef float [BFloat32](#)
- typedef double [BFloat64](#)
- typedef char [BChar](#)
- typedef [BInt32](#) [BInt](#)
- typedef [BUInt32](#) [BUInt](#)
- typedef [BFloat32](#) [BFloat](#)
- typedef [BFloat64](#) [BDouble](#)
- typedef size_t [BSize](#)
- typedef std::vector< [BFloat32](#) > [BArrayFloat](#)
- typedef std::vector< [BFloat64](#) > [BArrayDouble](#)
- typedef [BUInt32](#) [BTimeout](#)

Enumerations

- enum [BType](#) {
[BTypeNone](#), [BTypeBool](#), [BTypeInt8](#), [BTypeUInt8](#),
[BTypeInt16](#), [BTypeUInt16](#), [BTypeInt32](#), [BTypeUInt32](#),
[BTypeInt64](#), [BTypeUInt64](#), [BTypeFloat32](#), [BTypeFloat64](#),
[BTypeChar](#), [BTypeString](#), [BTypeError](#), [BTypeTime](#),
[BTypeObj](#) = 100 }
- enum [BTypeComp](#) {
[BTypeCompSingle](#), [BTypeCompArray](#), [BTypeCompArrayFixed](#), [BTypeCompList](#),
[BTypeCompDict](#) }

Functions

- [BTimeout timeoutTicks](#) ([BTimeout timeoutUs](#))
- void [byteSwap8](#) (void *d, void *s)
- void [byteSwap16](#) (void *d, void *s)
- void [byteSwap32](#) (void *d, void *s)
- void [byteSwap64](#) (void *d, void *s)

Variables

- const [BTimeout BTimeoutForever](#) = 0xFFFFFFFF

8.104.1 Typedef Documentation

8.104.1.1 typedef std::vector<BFloat64> **BArrayDouble**

8.104.1.2 typedef std::vector<BFloat32> **BArrayFloat**

8.104.1.3 typedef char **BChar**

8.104.1.4 typedef BFloat64 **BDouble**

8.104.1.5 typedef BFloat32 **BFloat**

8.104.1.6 typedef float **BFloat32**

8.104.1.7 typedef double **BFloat64**

8.104.1.8 typedef BInt32 **BInt**

8.104.1.9 typedef int16_t **BInt16**

8.104.1.10 typedef int32_t **BInt32**

8.104.1.11 typedef int64_t **BInt64**

8.104.1.12 typedef int8_t **BInt8**

8.104.1.13 typedef bool **Bool**

8.104.1.14 typedef size_t **BSize**

8.104.1.15 typedef BUInt32 **BTimeout**

8.104.1.16 typedef BUInt32 **BUInt**

8.104.1.17 typedef uint16_t **BUInt16**

8.104.1.18 typedef uint32_t **BUInt32**

8.104.1.19 typedef uint64_t **BUInt64**

8.104.1.20 typedef uint8_t **BUInt8**

8.104.2 Enumeration Type Documentation

8.104.2.1 enum BType

Enumerator

BTypeNone
BTypeBool
BTypeInt8
BTypeUInt8
BTypeInt16
BTypeUInt16
BTypeInt32
BTypeUInt32
BTypeInt64
BTypeUInt64
BTypeFloat32
BTypeFloat64
BTypeChar
BTypeString
BTypeError
BTypeTime
BTypeObj

8.104.2.2 enum BTypeComp

Enumerator

BTypeCompSingle
BTypeCompArray
BTypeCompArrayFixed
BTypeCompList
BTypeCompDict

8.104.3 Function Documentation

- 8.104.3.1 void byteSwap16 (void * *d*, void * *s*) [inline]
- 8.104.3.2 void byteSwap32 (void * *d*, void * *s*) [inline]
- 8.104.3.3 void byteSwap64 (void * *d*, void * *s*) [inline]
- 8.104.3.4 void byteSwap8 (void * *d*, void * *s*) [inline]
- 8.104.3.5 BTimeout timeoutTicks (BTimeout *timeoutUs*) [inline]

8.104.4 Variable Documentation

- 8.104.4.1 const BTimeout BTimeoutForever = 0xFFFFFFFF

8.105 /src/cern/tms/beam/libBeam/BUrl.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <memory.h>
#include <BUrl.h>
#include <curl/curl.h>
```

8.106 /src/cern/tms/beam/libBeam/BUrl.h File Reference

```
#include <stdio.h>
#include <BString.h>
#include <BError.h>
```

Classes

- class [BUrl](#)
Basic access to a Url.

8.107 overview.dox File Reference

8.108 SigGen.cpp File Reference

```
#include <SigGen.h>
#include <math.h>
#include <time.h>
```

Macros

- #define [DEBUG](#) 0
- #define [dprintf](#)(fmt, a...)

8.108.1 Macro Definition Documentation

8.108.1.1 #define [DEBUG](#) 0

8.108.1.2 #define [dprintf](#)(*fmt*, *a...*)

8.109 SigGen.h File Reference

```
#include <BList.h>
#include <BError.h>
```

Classes

- class [BSignal](#)
- class [SigGen](#)
- class [SigGenSine](#)
- class [SigGenSquare](#)
- class [SigGenNoise](#)
- class [SigGenPulse](#)
- class [SigGenBeam](#)

Typedefs

- typedef float [Sample](#)
- typedef [BList](#)< [BSignal](#) > [BSignalList](#)

8.109.1 Typedef Documentation

8.109.1.1 typedef [BList](#)<[BSignal](#)> [BSignalList](#)

8.109.1.2 typedef float [Sample](#)

8.110 test1.cpp File Reference

```
#include <stdio.h>
#include <TmsLib.h>
```

Functions

- void [printCycleParams](#) ([Tms::CycleParam](#) p)
- int [main](#) ()

8.110.1 Function Documentation

8.110.1.1 int [main](#) ()

8.110.1.2 void [printCycleParams](#) ([Tms::CycleParam](#) p)

8.111 TmsC.cc File Reference

```
#include <TmsC.h>
```

Namespaces

- [Tms](#)

8.112 TmsC.h File Reference

This file contains the TmsAPi class definitions.

```
#include <stdlib.h>
#include <stdint.h>
#include <Boap.h>
#include <BString.h>
#include <BList.h>
#include <BArray.h>
#include <TmsD.h>
```

Classes

- class [Tms::PuControl](#)
This class defines the parameters for a test data capture.
- class [Tms::PuProcess](#)
This interface provides functions to configure and capture data from individual pick-up.
- class [Tms::TmsControl](#)
This interface provides functions to control, test and get statistics from the TMS as a whole.
- class [Tms::TmsProcess](#)
This interface provides functions to capture data from the TMS as a whole.
- class [Tms::TmsEvent](#)
This interface provides functions for events to be sent to clients from the TMS as a whole.

Namespaces

- [Tms](#)

Variables

- const [BUInt32 Tms::apiVersion](#) = 0

8.112.1 Detailed Description

This file contains the TmsAPi class definitions.

8.113 TmsCycleParam.cc File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <unistd.h>
#include <errno.h>
#include <math.h>
#include <TmsCycleParam.h>
#include <BFile.h>
#include <BEntry.h>
```

Namespaces

- [Tms](#)

8.114 TmsCycleParam.h File Reference

```
#include <TmsLib.h>
```

Classes

- class [Tms::CycleParamState](#)
- class [Tms::CycleParamEdit](#)
Cycle Parameter management class.

Namespaces

- [Tms](#)

8.115 TmsD.cc File Reference

```
#include <TmsD.h>
```

Namespaces

- [Tms](#)

8.116 TmsD.h File Reference

```
#include <Boap.h>  
#include <BObj.h>  
#include <BDate.h>  
#include <BTimeStamp.h>  
#include <BComplex.h>  
#include <BList.h>  
#include <BArray.h>
```

Classes

- class [Tms::NameValue](#)
- class [Tms::PuChannel](#)
This class stores a Physical Pick-Up channel id.
- class [Tms::PuStatus](#)
This class stores the status of an individual Pick-Up.
- class [Tms::ConfigInfo](#)
This class describes the configuration of the TMS.
- class [Tms::DataInfo](#)

This class defines the data to be acquired and/or fetched.

- class [Tms::DataValue](#)

This is the definition of a single data value.

- class [Tms::Data](#)

This class stores the raw data.

- class [Tms::PuStateTable](#)

This class defines the Pick-Up state table.

- class [Tms::CycleParam](#)

This class defines the parameters for a PS processing cycle.

- class [Tms::CycleParamItem](#)

- class [Tms::TestCaptureInfo](#)

This class defines the parameters for a test data capture.

- class [Tms::PupeConfig](#)

- class [Tms::CycleInformationPeriod](#)

Cycle information.

- class [Tms::CycleInformation](#)

- class [Tms::CycleTypeInfoPeriod](#)

Cycle Type information.

- class [Tms::CycleTypeInfo](#)

- class [Tms::Simulation](#)

Namespaces

- [Tms](#)

Enumerations

- enum [Tms::CyclePeriod](#) { [Tms::CyclePeriodAll](#), [Tms::CyclePeriodCalibration](#), [Tms::CyclePeriodEvent0](#), [Tms::CyclePeriodEvent1](#), [Tms::CyclePeriodEvent2](#), [Tms::CyclePeriodEvent3](#), [Tms::CyclePeriodEvent4](#), [Tms::CyclePeriodEvent5](#), [Tms::CyclePeriodEvent6](#), [Tms::CyclePeriodEvent7](#), [Tms::CyclePeriodEvent8](#), [Tms::CyclePeriodEvent9](#) }
- enum [Tms::DataType](#) { [Tms::DataTypeRaw](#) }
- enum [Tms::DataFunction](#) { [Tms::DataFunctionRaw](#), [Tms::DataFunctionMean](#), [Tms::DataFunctionMeanAll](#), [Tms::DataFunctionMean0](#), [Tms::DataFunctionMean1](#) }
- enum [Tms::TestOutput](#) { [Tms::TestOutputFrefLocal](#), [Tms::TestOutputPIIL1](#), [Tms::TestOutputPIIL2](#) }
- enum [Tms::Priority](#) { [Tms::PriorityLow](#), [Tms::PriorityNormal](#), [Tms::PriorityHigh](#) }

8.117 TmsEventServerList.cc File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <TmsEventServerList.h>
```

Namespaces

- [Tms](#)

8.118 TmsEventServerList.h File Reference

```
#include <TmsD.h>
#include <TmsC.h>
```

Classes

- class [Tms::TmsEventServerList](#)

Namespaces

- [Tms](#)

8.119 tmsFunctions.dox File Reference

Namespaces

- [Tms](#)

8.120 TmsLib.cc File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <unistd.h>
#include <math.h>
#include <TmsLib.h>
#include <BDir.h>
#include <BEntry.h>
#include <BFile.h>
#include <TmsCycleParam.h>
```

Namespaces

- [Tms](#)

8.121 TmsLib.h File Reference

```
#include <TmsD.h>
#include <TmsC.h>
```

Classes

- union [Tms::TmsState](#)
The [Tms](#) State entry.
- union [Tms::TmsPhase](#)
The [Tms](#) Phase Table Entry.

- class [Tms::CycleParamDb](#)
Internal CycleParameter management class.

Namespaces

- [Tms](#)

Enumerations

- enum [Tms::TimingSig](#) {
[Tms::TimingSigClock](#) = 0x01, [Tms::TimingSigCycleStart](#) = 0x02, [Tms::TimingSigCycleStop](#) = 0x04, [Tms::TimingSigCalStart](#) = 0x08,
[Tms::TimingSigCalStop](#) = 0x10, [Tms::TimingSigInjection](#) = 0x20, [Tms::TimingSigHChange](#) = 0x40, [Tms::TimingSigFRef](#) = 0x80 }
The timing signal bits.
- enum [Tms::CaptureClock](#) {
[Tms::ClkAdcDiv_1](#) = 0x00, [Tms::ClkAdcDiv_2](#) = 0x01, [Tms::ClkAdcDiv_5](#) = 0x02, [Tms::ClkAdcDiv_10](#) = 0x03,
[Tms::ClkAdcDiv_20](#) = 0x04, [Tms::ClkAdcDiv_50](#) = 0x05, [Tms::ClkAdcDiv_100](#) = 0x06, [Tms::ClkAdcDiv_200](#) = 0x07,
[Tms::ClkAdcDiv_500](#) = 0x08, [Tms::ClkAdcDiv_1000](#) = 0x09, [Tms::ClkAdcDiv_2000](#) = 0x0A, [Tms::ClkAdcDiv_5000](#) = 0x0B,
[Tms::ClkAdcDiv_10000](#) = 0x0C, [Tms::ClkAdcDiv_20000](#) = 0x0D, [Tms::ClkAdcDiv_50000](#) = 0x0E, [Tms::ClkAdcDiv_100000](#) = 0x0F,
[Tms::ClkMs](#) = 0x10, [Tms::ClkFref](#) = 0x11 }
The Diagnostics Capture Clock settings.

Variables

- const unsigned int [Tms::tmsNumPickups](#) = 40
The default number of pick ups.
- const unsigned int [Tms::tmsPhaseTableSize](#) = 512
The size of the Phase Table.

8.122 TmsS.cc File Reference

```
#include <TmsC.h>
#include <TmsS.h>
```

Namespaces

- [Tms](#)

8.123 TmsT.cc File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <TmsT.h>
```

Index

- ~BBuffer
 - BBuffer, [31](#)
- ~BBufferStore
 - BBufferStore, [33](#)
- ~BComms
 - BComms, [36](#)
- ~BCond
 - BCond, [37](#)
- ~BCondBool
 - BCondBool, [38](#)
- ~BCondInt
 - BCondInt, [40](#)
- ~BCondResource
 - BCondResource, [42](#)
- ~BCondValue
 - BCondValue, [43](#)
- ~BCondWrap
 - BCondWrap, [45](#)
- ~BDate
 - BDate, [49](#)
- ~BDebugBacktrace
 - BDebugBacktrace, [51](#)
- ~BDir
 - BDir, [57](#)
- ~BDuration
 - BDuration, [59](#)
- ~BEntryFile
 - BEntryFile, [63](#)
- ~BEvent1
 - BEvent1, [72](#)
- ~BEvent1Int
 - BEvent1Int, [74](#)
- ~BEvent1Pipe
 - BEvent1Pipe, [75](#)
- ~BEventPipe
 - BEventPipe, [77](#)
- ~BFifo
 - BFifo, [79](#)
- ~BFifoCirc
 - BFifoCirc, [82](#)
- ~BFile
 - BFile, [87](#)
- ~BList
 - BList, [94](#)
- ~BMutex
 - BMutex, [98](#)
- ~BMutexLock
 - BMutexLock, [99](#)
- ~BMySQL
 - BMySQL, [100](#)
- ~BObj
 - BObj, [130](#)
- ~BPoll
 - BPoll, [132](#)
- ~BQueue
 - BQueue, [134](#)
- ~BRWLock
 - BRWLock, [140](#)
- ~BRefData
 - BRefData, [136](#)
- ~BRtc
 - BRtc, [137](#)
- ~BRtcThreaded
 - BRtcThreaded, [139](#)
- ~BSema
 - BSema, [141](#)
- ~BSemaphore
 - BSemaphore, [143](#)
- ~BSemaphoreCount
 - BSemaphoreCount, [144](#)
- ~BSignal
 - BSignal, [145](#)
- ~BSocket
 - BSocket, [147](#)
- ~BSocketAddress
 - BSocketAddress, [149](#)
- ~BString
 - BString, [156](#)
- ~BTable
 - BTable, [164](#)
- ~BThread
 - BThread, [165](#)
- ~BTimeStamp
 - BTimeStamp, [172](#)
- ~BTimeStampMs
 - BTimeStampMs, [177](#)
- ~BTimer
 - BTimer, [170](#)
- ~BUrl
 - BUrl, [180](#)
- ~BoapClientObject
 - BoapClientObject, [105](#)
- ~BoapMcClientObject
 - BoapMcClientObject, [109](#)
- ~BoapMcComms
 - BoapMcComms, [111](#)
- ~BoapMcServiceObject
 - BoapMcServiceObject, [115](#)

- ~BoapPacket
 - BoapPacket, [119](#)
- ~BoapServer
 - BoapServer, [123](#)
- ~BoapServerConnection
 - BoapServerConnection, [125](#)
- ~BoapServiceObject
 - BoapServiceObject, [128](#)
- ~SigGen
 - SigGen, [210](#)
- ~SigGenBeam
 - SigGenBeam, [211](#)
- ~SigGenNoise
 - SigGenNoise, [212](#)
- ~SigGenPulse
 - SigGenPulse, [213](#)
- ~SigGenSine
 - SigGenSine, [214](#)
- ~SigGenSquare
 - SigGenSquare, [215](#)
- ~TmsEventServerList
 - Tms::TmsEventServerList, [227](#)
- /src/cern/tms/beam/libBeam/BArray.h, [235](#)
- /src/cern/tms/beam/libBeam/BAtomic.h, [235](#)
- /src/cern/tms/beam/libBeam/BAtomicCount.h, [236](#)
- /src/cern/tms/beam/libBeam/BBuffer.cpp, [236](#)
- /src/cern/tms/beam/libBeam/BBuffer.h, [236](#)
- /src/cern/tms/beam/libBeam/BComms.cpp, [237](#)
- /src/cern/tms/beam/libBeam/BComms.h, [237](#)
- /src/cern/tms/beam/libBeam/BComplex.h, [237](#)
- /src/cern/tms/beam/libBeam/BCond.cpp, [238](#)
- /src/cern/tms/beam/libBeam/BCond.h, [238](#)
- /src/cern/tms/beam/libBeam/BCondInt.cpp, [238](#)
- /src/cern/tms/beam/libBeam/BCondInt.h, [238](#)
- /src/cern/tms/beam/libBeam/BConfig.cpp, [239](#)
- /src/cern/tms/beam/libBeam/BConfig.h, [239](#)
- /src/cern/tms/beam/libBeam/BCrc16.cpp, [239](#)
- /src/cern/tms/beam/libBeam/BCrc16.h, [240](#)
- /src/cern/tms/beam/libBeam/BDate.cpp, [240](#)
- /src/cern/tms/beam/libBeam/BDate.h, [241](#)
- /src/cern/tms/beam/libBeam/BDebug.cpp, [242](#)
- /src/cern/tms/beam/libBeam/BDebug.h, [243](#)
- /src/cern/tms/beam/libBeam/BDict.cpp, [244](#)
- /src/cern/tms/beam/libBeam/BDict.h, [245](#)
- /src/cern/tms/beam/libBeam/BDictMap.h, [245](#)
- /src/cern/tms/beam/libBeam/BDir.cpp, [246](#)
- /src/cern/tms/beam/libBeam/BDir.h, [246](#)
- /src/cern/tms/beam/libBeam/BDuration.cpp, [246](#)
- /src/cern/tms/beam/libBeam/BDuration.h, [247](#)
- /src/cern/tms/beam/libBeam/BEndian.cpp, [247](#)
- /src/cern/tms/beam/libBeam/BEndian.h, [247](#)
- /src/cern/tms/beam/libBeam/BEntry.cpp, [250](#)
- /src/cern/tms/beam/libBeam/BEntry.h, [250](#)
- /src/cern/tms/beam/libBeam/BError.cpp, [251](#)
- /src/cern/tms/beam/libBeam/BError.h, [251](#)
- /src/cern/tms/beam/libBeam/BErrorTime.cpp, [252](#)
- /src/cern/tms/beam/libBeam/BErrorTime.h, [252](#)
- /src/cern/tms/beam/libBeam/BEvent.cpp, [252](#)
- /src/cern/tms/beam/libBeam/BEvent.h, [252](#)
- /src/cern/tms/beam/libBeam/BEvent1.cpp, [253](#)
- /src/cern/tms/beam/libBeam/BEvent1.h, [253](#)
- /src/cern/tms/beam/libBeam/BFifo.h, [254](#)
- /src/cern/tms/beam/libBeam/BFifo.inc, [254](#)
- /src/cern/tms/beam/libBeam/BFifoCirc.cpp, [254](#)
- /src/cern/tms/beam/libBeam/BFifoCirc.h, [254](#)
- /src/cern/tms/beam/libBeam/BFifoCirc.inc, [255](#)
- /src/cern/tms/beam/libBeam/BFile.cpp, [255](#)
- /src/cern/tms/beam/libBeam/BFile.h, [255](#)
- /src/cern/tms/beam/libBeam/BFileCsv.cpp, [255](#)
- /src/cern/tms/beam/libBeam/BFileCsv.h, [256](#)
- /src/cern/tms/beam/libBeam/BFileData.cpp, [256](#)
- /src/cern/tms/beam/libBeam/BFileData.h, [256](#)
- /src/cern/tms/beam/libBeam/BList.h, [256](#)
- /src/cern/tms/beam/libBeam/BList_func.h, [257](#)
- /src/cern/tms/beam/libBeam/BMutex.cpp, [257](#)
- /src/cern/tms/beam/libBeam/BMutex.h, [257](#)
- /src/cern/tms/beam/libBeam/BMySQL.cpp, [257](#)
- /src/cern/tms/beam/libBeam/BMySQL.h, [257](#)
- /src/cern/tms/beam/libBeam/BNameValue.h, [258](#)
- /src/cern/tms/beam/libBeam/BObj.cpp, [265](#)
- /src/cern/tms/beam/libBeam/BObj.h, [265](#)
- /src/cern/tms/beam/libBeam/BObjStringFormat.cpp, [266](#)
- /src/cern/tms/beam/libBeam/BObjStringFormat.h, [268](#)
- /src/cern/tms/beam/libBeam/BPoll.cpp, [270](#)
- /src/cern/tms/beam/libBeam/BPoll.h, [270](#)
- /src/cern/tms/beam/libBeam/BQueue.h, [271](#)
- /src/cern/tms/beam/libBeam/BRWLock.cpp, [272](#)
- /src/cern/tms/beam/libBeam/BRWLock.h, [272](#)
- /src/cern/tms/beam/libBeam/BRefData.cpp, [271](#)
- /src/cern/tms/beam/libBeam/BRefData.h, [271](#)
- /src/cern/tms/beam/libBeam/BRtc.cpp, [272](#)
- /src/cern/tms/beam/libBeam/BRtc.h, [272](#)
- /src/cern/tms/beam/libBeam/BSema.cpp, [272](#)
- /src/cern/tms/beam/libBeam/BSema.h, [273](#)
- /src/cern/tms/beam/libBeam/BSemaphore.cpp, [273](#)
- /src/cern/tms/beam/libBeam/BSemaphore.h, [273](#)
- /src/cern/tms/beam/libBeam/BSocket.cpp, [273](#)
- /src/cern/tms/beam/libBeam/BSocket.h, [274](#)
- /src/cern/tms/beam/libBeam/BSpi.cpp, [274](#)
- /src/cern/tms/beam/libBeam/BSpi.h, [274](#)
- /src/cern/tms/beam/libBeam/BString.cpp, [275](#)
- /src/cern/tms/beam/libBeam/BString.h, [277](#)
- /src/cern/tms/beam/libBeam/BStringLocked.h, [278](#)
- /src/cern/tms/beam/libBeam/BTable.cpp, [278](#)
- /src/cern/tms/beam/libBeam/BTable.h, [278](#)
- /src/cern/tms/beam/libBeam/BThread.cpp, [278](#)
- /src/cern/tms/beam/libBeam/BThread.h, [279](#)
- /src/cern/tms/beam/libBeam/BTime.cpp, [279](#)
- /src/cern/tms/beam/libBeam/BTime.h, [279](#)
- /src/cern/tms/beam/libBeam/BTimeStamp.cpp, [280](#)
- /src/cern/tms/beam/libBeam/BTimeStamp.h, [281](#)
- /src/cern/tms/beam/libBeam/BTimeStampMs.cpp, [281](#)
- /src/cern/tms/beam/libBeam/BTimeStampMs.h, [281](#)
- /src/cern/tms/beam/libBeam/BTimer.cpp, [280](#)
- /src/cern/tms/beam/libBeam/BTimer.h, [280](#)

- [/src/cern/tms/beam/libBeam/BTypes.h](#), 282
- [/src/cern/tms/beam/libBeam/BUrl.cpp](#), 285
- [/src/cern/tms/beam/libBeam/BUrl.h](#), 285
- [/src/cern/tms/beam/libBeam/Boap.cpp](#), 258
- [/src/cern/tms/beam/libBeam/Boap.h](#), 259
- [/src/cern/tms/beam/libBeam/BoapMc.cpp](#), 260
- [/src/cern/tms/beam/libBeam/BoapMc.h](#), 261
- [/src/cern/tms/beam/libBeam/BoapSimple.cc](#), 263
- [/src/cern/tms/beam/libBeam/BoapSimple.h](#), 264
- [/src/cern/tms/beam/libBeam/BoapnsC.cpp](#), 262
- [/src/cern/tms/beam/libBeam/BoapnsC.h](#), 262
- [/src/cern/tms/beam/libBeam/BoapnsD.cpp](#), 263
- [/src/cern/tms/beam/libBeam/BoapnsD.h](#), 263
- [__attribute__](#)
 - [BoapMc.h](#), 262
- APIVERSION_TEST
 - [Boap.cpp](#), 258
- accept
 - [BSocket](#), 147
- acquireData
 - [Tms::TmsState](#), 233
- adcSysclkSync
 - [Tms::PupeConfig](#), 205
- add
 - [BAtomic](#), 29
 - [BAtomicCount](#), 30
 - [BSemaphoreCount](#), 144
 - [BString](#), 156
 - [BTimer](#), 170
- addEntry
 - [Boapns::Boapns](#), 117
- addEventServer
 - [Tms::PuProcess](#), 207
 - [Tms::TmsProcess](#), 230
- addMicroSeconds
 - [BDuration](#), 59
 - [BTimeStamp](#), 173
- addMilliSeconds
 - [BDuration](#), 59
 - [BTimeStamp](#), 173
 - [BTimeStampMs](#), 177
- addObject
 - [BoapServer](#), 123
- addRef
 - [BRefData](#), 136
- addRow
 - [BTable](#), 164
- addSeconds
 - [BDuration](#), 59
 - [BTime](#), 167
 - [BTimeStamp](#), 173
 - [BTimeStampMs](#), 177
- address
 - [BSocketAddressINET](#), 151
- addressFrom
 - [BoapMc.h](#), 262
 - [BoapMcPacketHead](#), 115
- addressList
 - [Boapns::BoapEntry](#), 107
- addressTo
 - [BoapMc.h](#), 262
 - [BoapMcPacketHead](#), 115
- apiVersion
 - [Boapns](#), 21
 - [Tms](#), 25
- append
 - [BArray](#), 28
 - [BDict](#), 53
 - [BList](#), 94
 - [BPoll](#), 132
 - [BString](#), 156
 - [Tms::TmsEventServerList](#), 227
- arg
 - [BEvent](#), 71
- argument
 - [Tms::DataInfo](#), 197
- average
 - [BTimer](#), 170
- BComms
 - [WaitError](#), 35
 - [WaitNone](#), 35
 - [WaitRead](#), 35
 - [WaitWrite](#), 35
- BError.h
 - [ErrorAccessDenied](#), 251
 - [ErrorAppBase](#), 252
 - [ErrorChecksum](#), 251
 - [ErrorComms](#), 251
 - [ErrorConfig](#), 251
 - [ErrorData](#), 251
 - [ErrorDataPresent](#), 252
 - [ErrorEndOfData](#), 252
 - [ErrorEndOfFile](#), 251
 - [ErrorFile](#), 251
 - [ErrorFormat](#), 251
 - [ErrorInit](#), 251
 - [ErrorMisc](#), 251
 - [ErrorNoData](#), 252
 - [ErrorNotAvailable](#), 251
 - [ErrorNotImplemented](#), 251
 - [ErrorOk](#), 251
 - [ErrorOverrun](#), 251
 - [ErrorParam](#), 251
 - [ErrorResourceLimit](#), 251
 - [ErrorTimeout](#), 251
 - [ErrorUnderrun](#), 251
 - [ErrorWarning](#), 251
- BErrorTime
 - [Error](#), 70
 - [None](#), 70
- BEvent.h
 - [BEventTypeNone](#), 253
- BEvent1.h
 - [BEvent1TypeError](#), 254
 - [BEvent1TypeInt](#), 254
 - [BEvent1TypeNone](#), 254

- BEvent1 TypeError
 - BEvent1.h, 254
 - BEvent1 TypeInt
 - BEvent1.h, 254
 - BEvent1 TypeNone
 - BEvent1.h, 254
 - BEventTypeNone
 - BEvent.h, 253
 - BFifoCirc
 - defaultSize, 82
 - BMutex
 - Normal, 98
 - Recursive, 98
 - BSignal
 - NumChannels, 145
 - BSocket
 - DGRAM, 147
 - PriorityHigh, 147
 - PriorityLow, 147
 - PriorityNormal, 147
 - STREAM, 147
 - BSpi
 - Mode0, 152
 - Mode1, 152
 - Mode2, 152
 - Mode3, 152
 - BTypeBool
 - BTypes.h, 284
 - BTypeChar
 - BTypes.h, 284
 - BTypeCompArray
 - BTypes.h, 284
 - BTypeCompArrayFixed
 - BTypes.h, 284
 - BTypeCompDict
 - BTypes.h, 284
 - BTypeCompList
 - BTypes.h, 284
 - BTypeCompSingle
 - BTypes.h, 284
 - BTypeError
 - BTypes.h, 284
 - BTypeFloat32
 - BTypes.h, 284
 - BTypeFloat64
 - BTypes.h, 284
 - BTypeInt16
 - BTypes.h, 284
 - BTypeInt32
 - BTypes.h, 284
 - BTypeInt64
 - BTypes.h, 284
 - BTypeInt8
 - BTypes.h, 284
 - BTypeNone
 - BTypes.h, 284
 - BTypeObj
 - BTypes.h, 284
 - BTypeString
 - BTypes.h, 284
 - BTypeTime
 - BTypes.h, 284
 - BTypeUInt16
 - BTypes.h, 284
 - BTypeUInt32
 - BTypes.h, 284
 - BTypeUInt64
 - BTypes.h, 284
 - BTypeUInt8
 - BTypes.h, 284
- BTypes.h
- BTypeBool, 284
 - BTypeChar, 284
 - BTypeCompArray, 284
 - BTypeCompArrayFixed, 284
 - BTypeCompDict, 284
 - BTypeCompList, 284
 - BTypeCompSingle, 284
 - BTypeError, 284
 - BTypeFloat32, 284
 - BTypeFloat64, 284
 - BTypeInt16, 284
 - BTypeInt32, 284
 - BTypeInt64, 284
 - BTypeInt8, 284
 - BTypeNone, 284
 - BTypeObj, 284
 - BTypeString, 284
 - BTypeTime, 284
 - BTypeUInt16, 284
 - BTypeUInt32, 284
 - BTypeUInt64, 284
 - BTypeUInt8, 284
- BArray
- append, 28
 - BArray, 28
 - BArray, 28
 - del, 28
 - insert, 28
 - number, 28
 - rear, 28
 - sort, 28
 - SortFunc, 28
- BArray< T >, 27
- BArray.h
- BArrayLoop, 235
- BArrayDouble
- BTypes.h, 283
- BArrayFloat
- BTypes.h, 283
- BArrayLoop
- BArray.h, 235
- BAtomic
- add, 29
 - BAtomic, 29
 - BAtomic, 29

- getValue, 29
 - operator Type, 29
 - operator++, 29
 - operator--, 29
 - ovalue, 29
- BAtomic< Type >, 28
- BAtomic.h
 - BAtomicInt32, 236
 - BAtomicInt64, 236
 - BAtomicUInt32, 236
 - BAtomicUInt64, 236
- BAtomicCount, 29
 - add, 30
 - BAtomicCount, 30
 - BAtomicCount, 30
 - getValue, 30
 - operator long, 30
 - operator++, 30
 - operator--, 30
 - ovalue, 30
- BAtomicInt32
 - BAtomic.h, 236
- BAtomicInt64
 - BAtomic.h, 236
- BAtomicUInt32
 - BAtomic.h, 236
- BAtomicUInt64
 - BAtomic.h, 236
- BBigEndian
 - BBuffer.h, 237
- BBuffer, 30
 - ~BBuffer, 31
 - BBuffer, 31
 - BBuffer, 31
 - data, 31
 - odata, 32
 - odataSize, 32
 - osize, 32
 - resize, 31
 - setData, 31
 - setSize, 31
 - size, 31
 - writeData, 31
- BBuffer.cpp
 - roundSize, 236
- BBuffer.h
 - BBigEndian, 237
- BBufferStore, 32
 - ~BBufferStore, 33
 - BBufferStore, 33
 - BBufferStore, 33
 - getHexString, 33
 - getPos, 33
 - opos, 34
 - oswapBytes, 34
 - pop, 33, 34
 - push, 34
 - setHexString, 34
 - setPos, 34
- BChar
 - BTypes.h, 283
- BComms, 35
 - ~BComms, 36
 - BComms, 36
 - BComms, 36
 - eventQueue, 36
 - init, 36
 - oevent, 36
 - oeventNum, 36
 - oeventQueue, 36
 - opacketMode, 36
 - otimeout, 36
 - packetMode, 36
 - read, 36
 - readAvailable, 36
 - setPacketMode, 36
 - setTimeout, 36
 - Wait, 35
 - wait, 36
 - write, 36
 - writeAvailable, 36
- BComplex
 - BComplex.h, 237
- BComplex.h
 - BComplex, 237
 - BComplex32, 237
 - BComplex64, 237
- BComplex32
 - BComplex.h, 237
- BComplex64
 - BComplex.h, 237
- BCond, 37
 - ~BCond, 37
 - BCond, 37
 - BCond, 37
 - ocond, 37
 - omutex, 37
 - signal, 37
 - timedWait, 37
 - wait, 37
- BCondBool, 37
 - ~BCondBool, 38
 - BCondBool, 38
 - BCondBool, 38
 - clear, 38
 - ocond, 39
 - omutex, 39
 - operator int, 38
 - ovalue, 39
 - set, 38
 - timedWait, 38
 - value, 38
 - wait, 38
- BCondInt, 39
 - ~BCondInt, 40
 - BCondInt, 40

- BCondInt, 40
- decrement, 40
- increment, 40
- ocond, 41
- omutex, 41
- operator++, 40
- operator+=", 40
- operator--, 40
- operator-=, 40
- ovalue, 41
- setValue, 40
- value, 40
- waitLessThan, 40
- waitLessThanOrEqual, 41
- waitMoreThanOrEqual, 41
- BCondInt.cpp
 - getTimeout, 238
- BCondResource, 41
 - ~BCondResource, 42
 - BCondResource, 42
 - BCondResource, 42
 - end, 42
 - inUse, 42
 - lock, 42
 - locked, 42
 - ocond, 42
 - olock, 42
 - omutex, 42
 - ouse, 42
 - start, 42
 - unlock, 42
- BCondValue, 42
 - ~BCondValue, 43
 - BCondValue, 43
 - BCondValue, 43
 - decrement, 43
 - increment, 43
 - ocond, 44
 - omutex, 44
 - operator++, 43
 - operator+=", 44
 - operator--, 44
 - operator-=, 44
 - ovalue, 44
 - setValue, 44
 - value, 44
 - waitLessThan, 44
 - waitLessThanOrEqual, 44
 - waitMoreThanOrEqual, 44
- BCondWrap, 45
 - ~BCondWrap, 45
 - BCondWrap, 45
 - BCondWrap, 45
 - decrement, 45
 - diff, 45
 - increment, 46
 - ocond, 46
 - omutex, 46
 - operator++, 46
 - operator+=", 46
 - operator--, 46
 - operator-=, 46
 - ovalue, 46
 - setValue, 46
 - value, 46
 - waitLessThan, 46
 - waitLessThanOrEqual, 46
 - waitMoreThanOrEqual, 46
- BConfig, 47
 - close, 47
 - fileName, 47
 - findValue, 47
 - ofile, 48
 - ofileName, 48
 - olock, 48
 - open, 47
 - read, 48
 - write, 48
- BCrc16.cpp
 - bcrc16, 239
 - table_crc_hi, 239
 - table_crc_lo, 240
- BCrc16.h
 - bcrc16, 240
- BDate, 48
 - ~BDate, 49
 - BDate, 49
 - BDate, 49
 - clear, 49
 - compare, 49
 - day, 49
 - daysInMonth, 49
 - getDate, 49
 - getString, 49
 - getStringFormatted, 49
 - isLeap, 50
 - isSet, 50
 - month, 50
 - operator BString, 50
 - operator<, 50
 - operator<=, 50
 - operator>, 50
 - operator>=, 50
 - operator==, 50
 - oyday, 51
 - oyear, 51
 - set, 50
 - setFirst, 50
 - setLast, 50
 - setNow, 50
 - setString, 50
 - setYDay, 50
 - yday, 50
 - year, 50
- BDate.cpp
 - fromBString, 241

- mon_yday, [241](#)
 - toBString, [241](#)
- BDate.h
 - fromBString, [241](#)
 - toBString, [241](#)
- BDebug.cpp
 - BTRACE_SIZE, [242](#)
 - bdebug, [243](#)
 - getTime, [242](#)
 - gettid, [242](#)
 - hd32, [242](#)
 - hd8, [242](#)
 - hd8a, [242](#)
 - hda32, [242](#)
 - hda8, [243](#)
 - STRBUF_SIZE, [243](#)
 - setDebug, [243](#)
 - tprintf, [243](#)
- BDebug.h
 - BDebug_STD, [244](#)
 - bdebug, [244](#)
 - dprintf, [244](#)
 - eprintf, [244](#)
 - getTime, [244](#)
 - gettid, [244](#)
 - hd32, [244](#)
 - hd8, [244](#)
 - hd8a, [244](#)
 - hda8, [244](#)
 - hds32, [244](#)
 - nprintf, [244](#)
 - setDebug, [244](#)
 - tprintf, [244](#)
 - wprintf, [244](#)
- BDebug_STD
 - BDebug.h, [244](#)
- BDebugBacktrace, [51](#)
 - ~BDebugBacktrace, [51](#)
 - BDebugBacktrace, [51](#)
 - BDebugBacktrace, [51](#)
 - dumpBacktrace, [51](#)
 - dumpBacktraceFile, [51](#)
 - dumpBacktraceStdout, [51](#)
 - dumpBacktraceSyslog, [51](#)
- BDict
 - append, [53](#)
 - BDict, [53](#)
 - BDict, [53](#)
 - clear, [53](#)
 - del, [53](#)
 - find, [53](#)
 - hasKey, [53](#)
 - hashAdd, [53](#)
 - hashDelete, [53](#)
 - hashFind, [53](#)
 - hashPrint, [53](#)
 - insert, [53](#)
 - iterator, [53](#)
 - key, [53](#)
 - ohashLists, [54](#)
 - ohashSize, [54](#)
 - operator+, [53](#)
 - operator=, [53](#)
- BDict< Type >, [52](#)
- BDict.cpp
 - bdictStringToString, [245](#)
 - fromBString, [245](#)
 - toBString, [245](#)
- BDict.h
 - BDictString, [245](#)
 - bdictStringToString, [245](#)
 - fromBString, [245](#)
 - toBString, [245](#)
- BDictItem
 - BDictItem, [54](#)
 - BDictItem, [54](#)
 - key, [54](#)
 - value, [54](#)
- BDictItem< Type >, [54](#)
- BDictMap
 - clear, [55](#)
 - del, [55](#)
 - hasKey, [55](#)
 - isEnd, [55](#)
 - iterator, [55](#)
 - key, [55](#)
 - next, [56](#)
 - size, [56](#)
 - start, [56](#)
- BDictMap< Value >, [55](#)
- BDictMap.h
 - BDictMapString, [246](#)
- BDictMapString
 - BDictMap.h, [246](#)
- BDictString
 - BDict.h, [245](#)
- BDir, [56](#)
 - ~BDir, [57](#)
 - BDir, [57](#)
 - BDir, [57](#)
 - clear, [57](#)
 - entryName, [57](#)
 - entryStat, [57](#)
 - entryStat64, [57](#)
 - error, [57](#)
 - odirname, [58](#)
 - oerror, [58](#)
 - open, [57](#)
 - osort, [58](#)
 - owild, [58](#)
 - read, [57](#)
 - setSort, [57](#)
 - setWild, [58](#)
- BDir.cpp
 - wild, [246](#)
 - wildString, [246](#)

- BDouble
 - BTypes.h, 283
- BDuration, 58
 - ~BDuration, 59
 - addMicroSeconds, 59
 - addMilliSeconds, 59
 - addSeconds, 59
 - BDuration, 59
 - BDuration, 59
 - clear, 59
 - getMicroSeconds, 59
 - getSeconds, 59
 - getString, 59
 - hour, 59
 - microSecond, 60
 - minute, 60
 - ohour, 60
 - omicroSecond, 60
 - ominute, 60
 - osecond, 60
 - ospare, 60
 - second, 60
 - set, 60
 - setString, 60
- BEndian.cpp
 - bswap_copy, 247
- BEndian.h
 - be16toh, 248
 - be32toh, 248
 - be64toh, 248
 - betoh, 249
 - bswap_copy, 249
 - bswap_p16, 249
 - bswap_p32, 249
 - bswap_p64, 249
 - bswap_p8, 249
 - htobe, 249
 - htobe16, 248
 - htobe32, 248
 - htobe64, 248
 - htole, 249, 250
 - htole16, 248
 - htole32, 248
 - htole64, 248
 - le16toh, 248
 - le32toh, 249
 - le64toh, 249
 - letoh, 250
- BEntry, 60
 - BEntry, 61
 - BEntry, 61
 - getName, 61
 - getValue, 61
 - line, 61
 - oname, 62
 - ovalue, 62
 - print, 62
 - setLine, 62
 - setName, 62
 - setValue, 62
- BEntryFile, 62
 - ~BEntryFile, 63
 - BEntryFile, 63
 - BEntryFile, 63
 - clear, 63
 - filename, 63
 - ocomments, 64
 - ofilename, 64
 - open, 63
 - read, 63
 - write, 64
 - writeList, 64
- BEntryList, 64
 - BEntryList, 65
 - BEntryList, 65
 - clear, 65
 - del, 65
 - deleteEntry, 65
 - find, 65
 - findValue, 65
 - getString, 66
 - insert, 66
 - isSet, 66
 - olastPos, 66
 - operator=, 66
 - print, 66
 - setValue, 66
 - setValueRaw, 66
- BError, 66
 - BError, 67
 - BError, 67
 - clear, 68
 - copy, 68
 - getErrorNo, 68
 - getNumber, 68
 - getString, 68
 - num, 68
 - oerrNo, 69
 - oerrStr, 69
 - operator int, 68
 - set, 68
 - setError, 68
 - str, 68
- BError.h
 - BErrorNum, 251
- BErrorNum
 - BError.h, 251
- BErrorTime, 69
 - BErrorTime, 70
 - BErrorTime, 70
 - clear, 70
 - copy, 70
 - getErrorNo, 70
 - getString, 70
 - getTime, 70
 - oerrNo, 70

- oerrStr, 70
- oerrTime, 70
- operator int, 70
- set, 70
- Type, 70
- BEvent, 71
 - arg, 71
 - BEvent, 71
 - BEvent, 71
 - oarg, 71
 - otype, 71
 - type, 71
- BEvent.h
 - BEventQueue, 253
 - BEventType, 253
- BEvent1, 72
 - ~BEvent1, 72
 - BEvent1, 72
 - BEvent1, 72
 - getBinary, 72
 - getType, 72
 - otype, 73
 - setBinary, 72
- BEvent1.h
 - BEvent1Type, 254
- BEvent1Error, 73
 - BEvent1Error, 73
 - BEvent1Error, 73
 - getBinary, 73
 - setBinary, 73
- BEvent1Int, 73
 - ~BEvent1Int, 74
 - BEvent1Int, 74
 - BEvent1Int, 74
 - clear, 74
 - getEvent, 74
 - getFd, 74
 - ofds, 74
 - sendEvent, 74
- BEvent1Pipe, 75
 - ~BEvent1Pipe, 75
 - BEvent1Pipe, 75
 - BEvent1Pipe, 75
 - clear, 75
 - getEvent, 75
 - getReceiveFd, 75
 - ofds, 76
 - sendEvent, 76
- BEvent1Type
 - BEvent1.h, 254
- BEventPipe, 76
 - ~BEventPipe, 77
 - BEventPipe, 77
 - BEventPipe, 77
 - clear, 77
 - getFd, 77
 - ofds, 77
 - read, 77
 - readAvailable, 77
 - write, 77
 - writeAvailable, 77
- BEventQueue
 - BEvent.h, 253
- BEventType
 - BEvent.h, 253
- BFifo
 - ~BFifo, 79
 - BFifo, 79
 - BFifo, 79
 - clear, 79
 - odata, 80
 - olock, 80
 - oreadPos, 80
 - osize, 80
 - owritePos, 80
 - read, 79
 - readAvailable, 79
 - readAvailableChunk, 79
 - readData, 79
 - readDone, 79
 - readPos, 79
 - resize, 79
 - size, 79
 - write, 79, 80
 - writeAvailable, 80
 - writeAvailableChunk, 80
 - writeBackup, 80
 - writeData, 80
 - writeDone, 80
- BFifo< Type >, 77
- BFifoCirc
 - ~BFifoCirc, 82
 - BFifoCirc, 82
 - BFifoCirc, 82
 - clear, 82
 - mapCircularBuffer, 82
 - odata, 83
 - olock, 83
 - oreadPos, 83
 - osize, 83
 - ovmSize, 84
 - owriteNumFifoSamples, 84
 - owritePos, 84
 - read, 82
 - readAvailable, 82
 - readData, 82
 - readDone, 83
 - readWaitAvailable, 83
 - size, 83
 - unmapCircularBuffer, 83
 - write, 83
 - writeAvailable, 83
 - writeData, 83
 - writeDone, 83
 - writeWaitAvailable, 83
- BFifoCirc< Type >, 81

- BFifoCirc.cpp
 - dprintf, [254](#)
- BFifoCircPos, [84](#)
 - BFifoCircPos, [85](#)
 - BFifoCircPos, [85](#)
 - difference, [85](#)
 - increment, [85](#)
 - operator int, [85](#)
 - operator+=", [85](#)
 - operator==, [85](#)
 - opos, [85](#)
 - osize, [85](#)
 - pos, [85](#)
 - set, [85](#)
 - setSize, [85](#)
- BFile, [85](#)
 - ~BFile, [87](#)
 - BFile, [87](#)
 - BFile, [87](#)
 - close, [87](#)
 - fgets, [87](#)
 - fileName, [87](#)
 - flush, [87](#)
 - getFd, [87](#)
 - isEnd, [87](#)
 - isOpen, [87](#)
 - length, [87](#)
 - ofile, [89](#)
 - ofileName, [89](#)
 - omode, [89](#)
 - open, [87](#), [88](#)
 - operator=, [88](#)
 - position, [88](#)
 - printf, [88](#)
 - read, [88](#)
 - readString, [88](#)
 - seek, [88](#)
 - setVBuf, [88](#)
 - truncate, [88](#)
 - write, [88](#)
 - writeString, [88](#)
- BFile.cpp
 - STRBUF, [255](#)
- BFileCsv, [89](#)
 - BFileCsv, [89](#)
 - BFileCsv, [89](#)
 - oseparator, [89](#)
 - readCsv, [89](#)
 - writeCsv, [89](#)
- BFileData, [90](#)
 - del, [90](#)
 - find, [90](#)
 - getNextId, [90](#)
 - ofilename, [90](#)
 - open, [90](#)
 - read, [90](#)
 - write, [90](#)
- BFloat
 - BTypes.h, [283](#)
- BFloat32
 - BTypes.h, [283](#)
- BFloat64
 - BTypes.h, [283](#)
- BInt
 - BTypes.h, [283](#)
- BInt16
 - BTypes.h, [283](#)
- BInt32
 - BTypes.h, [283](#)
- BInt64
 - BTypes.h, [283](#)
- BInt8
 - BTypes.h, [283](#)
- BIter, [91](#)
 - BIter, [91](#)
 - BIter, [91](#)
 - oi, [91](#)
 - operator BNode *, [91](#)
 - operator==, [91](#)
 - valid, [91](#)
- BList
 - ~BList, [94](#)
 - append, [94](#)
 - BList, [94](#)
 - begin, [94](#)
 - BList, [94](#)
 - clear, [94](#)
 - del, [94](#)
 - deleteFirst, [94](#)
 - deleteLast, [95](#)
 - end, [95](#)
 - front, [95](#)
 - get, [95](#)
 - goTo, [95](#)
 - has, [95](#)
 - insert, [95](#)
 - insertAfter, [95](#)
 - isEnd, [95](#)
 - next, [95](#)
 - nodeCreate, [96](#)
 - nodeGet, [96](#)
 - number, [96](#)
 - olength, [97](#)
 - onodes, [97](#)
 - operator+, [96](#)
 - operator=, [96](#)
 - pop, [96](#)
 - position, [96](#)
 - prev, [96](#)
 - push, [96](#)
 - queueAdd, [96](#)
 - queueGet, [96](#)
 - rear, [97](#)
 - size, [97](#)
 - sort, [97](#)
 - SortFunc, [94](#)

- start, 97
- swap, 97
- BList< T >, 91
- BList< T >::Node, 200
- BList.h
 - BListLoop, 257
- BList::Node
 - item, 200
 - Node, 200
- BListLoop
 - BList.h, 257
- BMutex, 97
 - ~BMutex, 98
 - BMutex, 98
 - BMutex, 98
 - lock, 98
 - omutex, 99
 - operator=, 98
 - timedLock, 99
 - tryLock, 99
 - Type, 98
 - unlock, 99
- BMutex.cpp
 - MDEBUG, 257
- BMutexLock, 99
 - ~BMutexLock, 99
 - BMutexLock, 99
 - BMutexLock, 99
 - lock, 99
 - olock, 100
 - unlock, 99
- BMysql, 100
 - ~BMysql, 100
 - BMysql, 100
 - BMysql, 100
 - close, 100
 - db, 100
 - del, 100
 - escapeString, 100
 - flush, 101
 - get, 101
 - insert, 101
 - odb, 101
 - odebug, 101
 - olock, 101
 - opened, 101
 - open, 101
 - query, 101
 - setDebug, 101
 - update, 101
- BNameValue
 - BNameValue, 102
 - BNameValue, 102
 - getName, 102
 - getValue, 102
 - oname, 102
 - ovalue, 102
- BNameValue< T >, 101
- BNameValueList
 - find, 102
 - findPos, 102
- BNameValueList< T >, 102
- BNode, 103
 - BNode, 103
 - BNode, 103
 - next, 103
 - prev, 103
- BObj, 130
 - ~BObj, 130
 - BObj, 130
 - BObj, 130
 - getDebugString, 130
 - getMember, 130
 - getMembers, 130
 - getType, 130
 - membersPrint, 130
 - setMember, 130
 - setMembers, 131
- BObjMember, 131
 - dataOffset, 131
 - name, 131
 - size, 131
 - type, 131
 - typeComp, 131
 - typeName, 131
- BObjStringFormat.cpp
 - toBDictStringFromJson, 267
 - toBString, 267
 - toBStringJson, 267, 268
- BObjStringFormat.h
 - base64_decode, 269
 - base64_encode, 269
 - toBDictStringFromJson, 269
 - toBString, 269
 - toBStringJson, 269, 270
- BPoll, 131
 - ~BPoll, 132
 - append, 132
 - BPoll, 132
 - BPoll, 132
 - clear, 132
 - delFd, 133
 - doPoll, 133
 - doPollEvents, 133
 - getPollFds, 133
 - getPollFdsNum, 133
 - nextFd, 133
 - ofds, 133
 - ofdsNext, 133
 - ofdsNum, 133
 - PollFd, 132
- BQueue
 - ~BQueue, 134
 - BQueue, 134
 - BQueue, 134
 - clear, 134

- oLock, 135
- onumber, 135
- osize, 135
- read, 134
- readAvailable, 134
- write, 134
- writeAvailable, 134
- BQueue< T >, 133
- BQueue.h
 - BQueueInt, 271
- BQueueInt
 - BQueue.h, 271
- BRWLock, 139
 - ~BRWLock, 140
 - BRWLock, 140
 - BRWLock, 140
 - oLock, 140
 - operator=, 140
 - rdLock, 140
 - tryRdLock, 140
 - tryWrLock, 140
 - unlock, 140
 - wrLock, 140
- BRefData, 135
 - ~BRefData, 136
 - addRef, 136
 - BRefData, 136
 - BRefData, 136
 - copy, 136
 - data, 136
 - deleteRef, 136
 - len, 136
 - odata, 136
 - olen, 136
 - operator=, 136
 - orefCount, 136
 - setLen, 136
- BRefData.cpp
 - CHUNK, 271
- BRtc, 137
 - ~BRtc, 137
 - BRtc, 137
 - BRtc, 137
 - init, 137
 - ofd, 138
 - orate, 138
 - wait, 137
- BRtcThreaded, 138
 - ~BRtcThreaded, 139
 - BRtcThreaded, 139
 - BRtcThreaded, 139
 - function, 139
 - init, 139
 - ocond, 139
 - orate, 139
 - ortc, 139
 - wait, 139
- BSema, 141
 - ~BSema, 141
 - BSema, 141
 - BSema, 141
 - getValue, 141
 - operator=, 141
 - osema, 142
 - post, 141
 - timedWait, 141
 - tryWait, 142
 - wait, 142
- BSemaphore, 142
 - ~BSemaphore, 143
 - BSemaphore, 143
 - BSemaphore, 143
 - getValue, 143
 - operator=, 143
 - osema, 143
 - set, 143
 - wait, 143
- BSemaphoreCount, 143
 - ~BSemaphoreCount, 144
 - add, 144
 - BSemaphoreCount, 144
 - BSemaphoreCount, 144
 - oLock, 144
 - operator=, 144
 - osema, 144
 - oValue, 144
 - setValue, 144
 - take, 144
 - value, 144
 - wait, 144
- BSignal, 144
 - ~BSignal, 145
 - BSignal, 145
 - BSignal, 145
 - data, 145
 - id, 145
 - nextId, 145
 - numRepeat, 145
 - numSamples, 145
 - operator=, 145
- BSignalList
 - SigGen.h, 286
- BSize
 - BTypes.h, 283
- BSocket, 146
 - ~BSocket, 147
 - accept, 147
 - BSocket, 147
 - bind, 147
 - BSocket, 147
 - close, 147
 - connect, 147
 - getAddress, 148
 - getFd, 148
 - getMTU, 148
 - getSockOpt, 148

- init, 148
- listen, 148
- NType, 147
- osocket, 148
- Priority, 147
- recv, 148
- recvFrom, 148
- recvFromWithTimeout, 148
- recvWithTimeout, 148
- send, 148
- sendTo, 148
- setBroadCast, 148
- setFd, 148
- setPriority, 148
- setReuseAddress, 148
- setSockOpt, 148
- shutdown, 148
- BSocket.cpp
 - IP_MTU, 274
- BSocketAddress, 148
 - ~BSocketAddress, 149
 - BSocketAddress, 149
 - BSocketAddress, 149
 - len, 149
 - oaddress, 150
 - olen, 150
 - operator const SockAddr *, 150
 - operator=, 150
 - operator==, 150
 - raw, 150
 - set, 150
 - SockAddr, 149
- BSocketAddressINET, 150
 - address, 151
 - getHostName, 151
 - getIpAddressList, 151
 - getIpAddressListAll, 151
 - getIpAddresses, 151
 - getString, 151
 - port, 151
 - set, 152
 - setPort, 152
 - SockAddrIP, 151
- BSpi, 152
 - BSpi, 153
 - BSpi, 153
 - init, 153
 - Mode, 152
 - odev, 153
 - odevName, 153
 - transact, 153
- BString, 153
 - ~BString, 156
 - add, 156
 - append, 156
 - BString, 156
 - base64Decode, 157
 - base64Encode, 157
 - basename, 157
 - BString, 156
 - clear, 157
 - compare, 157
 - compareRegex, 157
 - compareWild, 157
 - compareWildExpression, 157
 - convert, 157
 - convertHex, 158
 - copy, 158
 - csvDecode, 158
 - csvEncode, 158
 - del, 158
 - dirname, 158
 - extension, 158
 - field, 158
 - fields, 158
 - find, 158
 - findReverse, 158
 - firstLine, 158
 - fixedLen, 158
 - get, 159
 - getTokenList, 159
 - hash, 159
 - inString, 159
 - init, 159
 - insert, 159
 - isSpace, 159
 - justify, 159
 - len, 159
 - lowerFirst, 159
 - operator const char *, 159
 - operator<, 160
 - operator<=, 160
 - operator>, 160
 - operator>=, 160
 - operator+, 159, 160
 - operator+=, 160
 - operator=, 160
 - operator==, 160
 - ostr, 162
 - pad, 160
 - printf, 160
 - pullLine, 160
 - pullSeparators, 160
 - pullToken, 160
 - pullWord, 160
 - removeNL, 160
 - removeSeparators, 161
 - retDouble, 161
 - retInt, 161
 - retStr, 161
 - retStrDup, 161
 - retUInt, 161
 - reverse, 161
 - split, 161
 - subString, 161
 - toLowerCase, 161

- toUpper, 161
- translateChar, 161
- truncate, 162
- BString.cpp
 - barrayToString, 276
 - base64_decode_table, 276
 - blstToString, 276
 - bstringListinList, 276
 - bstringToArray, 276
 - bstringToList, 276
 - charToArray, 276
 - charToList, 276
 - fromBString, 276
 - gmatch, 276
 - MINUS, 275
 - operator<<, 276
 - operator>>, 276
 - STRIP, 275
 - toBString, 276
- BString.h
 - fromBString, 277
 - operator<<, 277
 - operator>>, 277
 - toBString, 277, 278
- BStringLocked, 162
 - BStringLocked, 162
 - BStringLocked, 162
 - len, 162
 - lock, 163
 - operator BString, 162
 - operator+, 163
 - operator=, 163
 - ostr, 163
- BStringMutex, 163
 - BStringMutex, 163
 - BStringMutex, 163
- BTRACE_SIZE
 - BDebug.cpp, 242
- BTable, 163
 - ~BTable, 164
 - addRow, 164
 - BTable, 164
 - BTable, 164
 - calculateWidths, 164
 - clear, 164
 - ocolumnWidths, 164
 - odata, 164
 - otitle, 164
 - print, 164
 - printLine, 164
 - setTitle, 164
- BThread, 164
 - ~BThread, 165
 - BThread, 165
 - BThread, 165
 - cancel, 165
 - function, 165
 - getThread, 165
 - opolicy, 166
 - opriority, 166
 - oreult, 166
 - orunning, 166
 - ostackSize, 166
 - othread, 166
 - result, 166
 - running, 166
 - setInitPriority, 166
 - setInitStackSize, 166
 - setPriority, 166
 - start, 166
 - startFunc, 166
 - waitForCompletion, 166
- BTime, 166
 - addSeconds, 167
 - BTime, 167
 - BTime, 167
 - getDate, 167
 - getSeconds, 167
 - getString, 167
 - getTime, 167
 - isLeapYear, 168
 - isSet, 168
 - operator<, 168
 - operator<=, 168
 - operator>, 168
 - operator>=, 168
 - operator+, 168
 - operator+=, 168
 - operator==, 168
 - otime, 168
 - set, 168
 - setString, 168
 - setYearDay, 168
- BTime.cpp
 - monDays, 279
 - yearDays, 279
 - yearIsLeap, 279
- BTimeStamp, 171
 - ~BTimeStamp, 172
 - addMicroSeconds, 173
 - addMilliSeconds, 173
 - addSeconds, 173
 - BTimeStamp, 172
 - BTimeStamp, 172
 - clear, 173
 - compare, 173
 - day, 173
 - difference, 173
 - getDate, 173
 - getString, 173
 - getStringFormatted, 173
 - getStringNoMs, 173
 - getYearMicroSeconds, 173
 - getYearSeconds, 173
 - hour, 173
 - isLeap, 174

- isSet, 174
- microSecond, 174
- minute, 174
- month, 174
- ohour, 175
- omicroSecond, 175
- ominute, 175
- operator BString, 174
- operator<, 174
- operator<=, 174
- operator>, 174
- operator>=, 174
- operator=, 174
- operator==, 174
- osecond, 175
- ospare, 175
- oyday, 175
- oyear, 175
- second, 174
- set, 174
- setFirst, 174
- setLast, 174
- setNow, 174
- setString, 174
- setTime, 175
- setYDay, 175
- yday, 175
- year, 175
- BTimeStamp.cpp
 - fromBString, 280
 - mon_yday, 280
 - toBString, 280
- BTimeStamp.h
 - fromBString, 281
 - toBString, 281
- BTimeStampMs, 175
 - ~BTimeStampMs, 177
 - addMilliseconds, 177
 - addSeconds, 177
 - BTimeStampMs, 177
 - BTimeStampMs, 177
 - clear, 177
 - compare, 177
 - difference, 177
 - getDate, 177
 - getDurationString, 177
 - getDurationStringNoMs, 177
 - getString, 178
 - getStringNoMs, 178
 - getStringRaw, 178
 - getYearMilliseconds, 178
 - getYearSeconds, 178
 - hour, 179
 - isLeap, 178
 - milliSecond, 179
 - minute, 179
 - operator<, 178
 - operator<=, 178
 - operator>, 178
 - operator>=, 178
 - sampleNumber, 179
 - second, 179
 - setDurationString, 178
 - setNow, 178
 - setString, 178
 - subMilliseconds, 178
 - subSeconds, 178
 - yday, 179
 - year, 179
- BTimeStampMs.cpp
 - mon_yday, 281
- BTimeout
 - BTypes.h, 283
- BTimeoutForever
 - BTypes.h, 284
- BTimer, 169
 - ~BTimer, 170
 - add, 170
 - average, 170
 - BTimer, 170
 - BTimer, 170
 - clear, 170
 - getElapsedTime, 170
 - getTime, 170
 - oaverage, 170
 - oendTime, 170
 - oclock, 170
 - onum, 170
 - opeak, 170
 - ostartTime, 170
 - peak, 170
 - start, 170
 - stop, 170
- BType
 - BTypes.h, 284
- BTypeComp
 - BTypes.h, 284
- BTypes.h
 - BArrayDouble, 283
 - BArrayFloat, 283
 - BChar, 283
 - BDouble, 283
 - BFloat, 283
 - BFloat32, 283
 - BFloat64, 283
 - BInt, 283
 - BInt16, 283
 - BInt32, 283
 - BInt64, 283
 - BInt8, 283
 - BSize, 283
 - BTimeout, 283
 - BTimeoutForever, 284
 - BType, 284
 - BTypeComp, 284
 - BUInt, 283

- BUInt16, [283](#)
- BUInt32, [283](#)
- BUInt64, [283](#)
- BUInt8, [283](#)
- Bool, [283](#)
- byteSwap16, [284](#)
- byteSwap32, [284](#)
- byteSwap64, [284](#)
- byteSwap8, [284](#)
- timeoutTicks, [284](#)
- BUInt
 - BTypes.h, [283](#)
- BUInt16
 - BTypes.h, [283](#)
- BUInt32
 - BTypes.h, [283](#)
- BUInt64
 - BTypes.h, [283](#)
- BUInt8
 - BTypes.h, [283](#)
- BUrl, [179](#)
 - ~BUrl, [180](#)
 - BUrl, [180](#)
 - BUrl, [180](#)
 - oinit, [180](#)
 - ores, [180](#)
 - readString, [180](#)
 - writeData, [180](#)
- barrayToString
 - BString.cpp, [276](#)
- base64_decode
 - BObjStringFormat.h, [269](#)
- base64_decode_table
 - BString.cpp, [276](#)
- base64_encode
 - BObjStringFormat.h, [269](#)
- base64Decode
 - BString, [157](#)
- base64Encode
 - BString, [157](#)
- basename
 - BString, [157](#)
- bcrc16
 - BCrc16.cpp, [239](#)
 - BCrc16.h, [240](#)
- bdebug
 - BDebug.cpp, [243](#)
 - BDebug.h, [244](#)
- bdictStringToString
 - BDict.cpp, [245](#)
 - BDict.h, [245](#)
- be16toh
 - BEndian.h, [248](#)
- be32toh
 - BEndian.h, [248](#)
- be64toh
 - BEndian.h, [248](#)
- begin
 - BList, [94](#)
- betoh
 - BEndian.h, [249](#)
- beyondPeriod
 - Tms::DataInfo, [197](#)
- bind
 - BSocket, [147](#)
- bit6
 - Tms::TmsState, [233](#)
- bit7
 - Tms::TmsState, [233](#)
- blistToString
 - BString.cpp, [276](#)
- blr
 - Tms::TmsPhase, [228](#)
- blrPhase
 - Tms::CycleParamState, [192](#)
- blrWidth
 - Tms::CycleParamState, [192](#)
- Boap.h
 - BoapPriorityHigh, [260](#)
 - BoapPriorityLow, [260](#)
 - BoapPriorityNormal, [260](#)
 - BoapTypeRpc, [260](#)
 - BoapTypeRpcError, [260](#)
 - BoapTypeRpcReply, [260](#)
 - BoapTypeSignal, [260](#)
- Boap.cpp
 - APIVERSION_TEST, [258](#)
 - boapPort, [259](#)
 - DEBUG, [258](#)
 - dprintf, [258](#)
 - IS_BIG_ENDIAN, [259](#)
- Boap.h
 - BoapFunc, [260](#)
 - BoapMagic, [260](#)
 - BoapPriority, [260](#)
 - BoapService, [260](#)
 - BoapType, [260](#)
- BoapMc.h
 - BoapMcTypeReply, [261](#)
 - BoapMcTypeRequest, [261](#)
- BoapMcTypeReply
 - BoapMc.h, [261](#)
- BoapMcTypeRequest
 - BoapMc.h, [261](#)
- BoapPriorityHigh
 - Boap.h, [260](#)
- BoapPriorityLow
 - Boap.h, [260](#)
- BoapPriorityNormal
 - Boap.h, [260](#)
- BoapServer
 - NOTHREADS, [123](#)
 - THREADED, [123](#)
- BoapSimple.h
 - BoapTypeRpc, [265](#)
 - BoapTypeRpcError, [265](#)

- BoapTypeRpcReply, 265
- BoapTypeSignal, 265
- BoapTypeRpc
 - Boap.h, 260
 - BoapSimple.h, 265
- BoapTypeRpcError
 - Boap.h, 260
 - BoapSimple.h, 265
- BoapTypeRpcReply
 - Boap.h, 260
 - BoapSimple.h, 265
- BoapTypeSignal
 - Boap.h, 260
 - BoapSimple.h, 265
- BoapClientObject, 103
 - ~BoapClientObject, 105
 - BoapClientObject, 105
 - BoapClientObject, 105
 - checkApiVersion, 105
 - connectService, 105
 - disconnectService, 105
 - getServiceName, 105
 - handleReconnect, 105
 - oapiVersion, 106
 - oconnected, 106
 - olock, 106
 - omaxLength, 106
 - oname, 106
 - opriority, 106
 - oreconnect, 106
 - orx, 106
 - oservice, 106
 - otimeout, 106
 - otx, 106
 - performCall, 105
 - performRecv, 105
 - performSend, 105
 - ping, 105
 - pingLocked, 106
 - setConnectionPriority, 106
 - setMaxLength, 106
 - setTimeout, 106
- BoapEntry
 - Boapns::BoapEntry, 107
- BoapFunc
 - Boap.h, 260
 - BoapSimple.h, 265
- BoapFuncEntry, 107
 - BoapFuncEntry, 108
 - BoapFuncEntry, 108
 - ocmd, 108
 - ofunc, 108
- BoapMagic
 - Boap.h, 260
- BoapMc.cpp
 - DEBUG_LOCAL, 261
 - DEBUG_LOCAL1, 261
 - dl1printf, 261
 - dlprintf, 261
- BoapMc.h
 - __attribute__, 262
 - addressFrom, 262
 - addressTo, 262
 - BoapMcType, 261
 - checksum, 262
 - cmd, 262
 - error, 262
 - length, 262
- BoapMcClientObject, 108
 - ~BoapMcClientObject, 109
 - BoapMcClientObject, 109
 - BoapMcClientObject, 109
 - getApiVersion, 109
 - oaddressFrom, 109
 - oaddressTo, 109
 - oapiVersion, 109
 - ocomms, 109
 - opacket, 109
 - performCall, 109
 - performRecv, 109
 - performSend, 109
 - setAddress, 109
- BoapMcComms, 110
 - ~BoapMcComms, 111
 - BoapMcComms, 111
 - BoapMcComms, 111
 - getApiVersion, 111
 - oaddressFrom, 112
 - oaddressTo, 112
 - oapiVersion, 112
 - ocomms, 112
 - olockCall, 113
 - olockTx, 113
 - opacket, 113
 - opacketReqQueue, 113
 - opacketReqRx, 113
 - opacketReqTx, 113
 - opacketRx, 113
 - opacketRxSema, 113
 - opacketTx, 113
 - opacketTxQueue, 113
 - opacketTxQueueWriteNum, 113
 - opacketTxSema, 113
 - oslave, 113
 - othreaded, 114
 - otimeout, 114
 - packetRecv, 111
 - packetSend, 111
 - performCall, 111
 - performSend, 111
 - processPacket, 112
 - processRequest, 112
 - processRequests, 112
 - processRx, 112
 - setAddress, 112
 - setComms, 112

- setCommsMode, 112
 - setTimeout, 112
- BoapMcPacket, 114
 - data, 114
 - head, 114
- BoapMcPacketHead, 114
 - addressFrom, 115
 - addressTo, 115
 - checksum, 115
 - cmd, 115
 - error, 115
 - length, 115
- BoapMcServiceObject, 115
 - ~BoapMcServiceObject, 115
 - BoapMcServiceObject, 115
 - BoapMcServiceObject, 115
 - oapiVersion, 116
 - process, 115
 - processEvent, 115
 - sendEvent, 115
- BoapMcSignalObject, 116
 - BoapMcSignalObject, 116
 - BoapMcSignalObject, 116
 - ocomms, 116
 - performSend, 116
- BoapMcType
 - BoapMc.h, 261
- BoapPacket, 117
 - ~BoapPacket, 119
 - BoapPacket, 119
 - BoapPacket, 119
 - data, 119
 - getCmd, 119
 - nbytes, 119
 - odata, 120
 - onbytes, 120
 - opos, 120
 - osize, 120
 - peekHead, 119
 - pop, 119
 - popHead, 119
 - push, 119, 120
 - pushHead, 120
 - resize, 120
 - setData, 120
 - updateHead, 120
 - updateLen, 120
- BoapPacketHead, 120
 - cmd, 121
 - length, 121
 - reserved, 121
 - service, 121
 - type, 121
- boapPort
 - Boap.cpp, 259
- BoapPriority
 - Boap.h, 260
- BoapServer, 121
 - ~BoapServer, 123
 - addObject, 123
 - BoapServer, 123
 - BoapServer, 123
 - clientGone, 123
 - function, 123
 - getConnectionsNumber, 123
 - getEventSocket, 123
 - getHostName, 123
 - getSocket, 123
 - init, 123
 - newConnection, 123
 - oboapNs, 124
 - oboapns, 124
 - oclientGoneEvent, 124
 - oclients, 124
 - ohostName, 124
 - oisBoapns, 124
 - onet, 124
 - onetEvent, 124
 - onetEventAddress, 124
 - onumOperations, 124
 - opoll, 124
 - orx, 124
 - oservices, 124
 - othreaded, 124
 - otx, 124
 - process, 123
 - processEvent, 123, 124
 - run, 124
 - sendEvent, 124
- BoapServerConnection, 125
 - ~BoapServerConnection, 125
 - BoapServerConnection, 125
 - BoapServerConnection, 125
 - function, 125
 - getHead, 125
 - getSocket, 126
 - init, 126
 - oboapServer, 126
 - omaxLength, 126
 - orx, 126
 - osocket, 126
 - otx, 126
 - process, 126
 - setMaxLength, 126
 - validate, 126
- BoapService
 - Boap.h, 260
 - BoapSimple.h, 265
- BoapServiceEntry, 126
 - BoapServiceEntry, 126
 - BoapServiceEntry, 126
 - object, 127
 - oservice, 127
- BoapServiceObject, 127
 - ~BoapServiceObject, 128
 - BoapServiceObject, 128

- BoapServiceObject, 128
- doConnectionPriority, 128
- doPing, 128
- name, 128
- oapiVersion, 128
- ofuncList, 128
- oname, 128
- oserver, 128
- process, 128
- processEvent, 128
- sendEvent, 128
- setName, 128
- BoapSignalObject, 129
 - BoapSignalObject, 129
 - BoapSignalObject, 129
 - orx, 129
 - otx, 129
 - performSend, 129
- BoapSimple.cc
 - DEBUG, 264
 - dprintf, 264
 - roundSize, 264
- BoapSimple.h
 - BoapFunc, 265
 - BoapService, 265
 - BoapType, 265
 - Double, 265
 - Int16, 265
 - Int32, 265
 - Int8, 265
 - UInt16, 265
 - UInt32, 265
 - UInt8, 265
- BoapType
 - Boap.h, 260
 - BoapSimple.h, 265
- Boapns, 21
 - apiVersion, 21
 - Boapns::Boapns, 117
- Boapns::BoapEntry, 107
 - addressList, 107
 - BoapEntry, 107
 - hostName, 107
 - name, 107
 - port, 107
 - service, 107
- Boapns::Boapns, 116
 - addEntry, 117
 - Boapns, 117
 - delEntry, 117
 - getEntry, 117
 - getEntryList, 117
 - getNewName, 117
 - getVersion, 117
- Bool
 - BTypes.h, 283
- bstringListinList
 - BString.cpp, 276
- bstringToArray
 - BString.cpp, 276
- bstringToList
 - BString.cpp, 276
- bswap_copy
 - BEndian.cpp, 247
 - BEndian.h, 249
- bswap_p16
 - BEndian.h, 249
- bswap_p32
 - BEndian.h, 249
- bswap_p64
 - BEndian.h, 249
- bswap_p8
 - BEndian.h, 249
- bunch
 - Tms::CycleParamEdit, 189
- bunchMask
 - Tms::CycleInformationPeriod, 182
 - Tms::CycleParamState, 192
 - Tms::CycleTypeInfoInformationPeriod, 194
 - Tms::PuStateTable, 208
- bunchNumber
 - Tms::DataInfo, 197
- byteSwap16
 - BTypes.h, 284
- byteSwap32
 - BTypes.h, 284
- byteSwap64
 - BTypes.h, 284
- byteSwap8
 - BTypes.h, 284
- CHUNK
 - BRefData.cpp, 271
- calStart
 - Tms::TmsState, 233
- calStop
 - Tms::TmsState, 233
- calculateWidths
 - BTable, 164
- cancel
 - BThread, 165
- CaptureClock
 - Tms, 23
- captureDiagnostics
 - Tms::PuControl, 203
 - Tms::TmsControl, 220
- channel
 - Tms::CycleParam, 185
 - Tms::CycleParamItem, 190
 - Tms::DataInfo, 197
- charToArray
 - BString.cpp, 276
- charToList
 - BString.cpp, 276
- checkApiVersion
 - BoapClientObject, 105
- checksum

- BoapMc.h, [262](#)
- BoapMcPacketHead, [115](#)
- clear
 - BCondBool, [38](#)
 - BDate, [49](#)
 - BDict, [53](#)
 - BDictMap, [55](#)
 - BDir, [57](#)
 - BDuration, [59](#)
 - BEntryFile, [63](#)
 - BEntryList, [65](#)
 - BError, [68](#)
 - BErrorTime, [70](#)
 - BEvent1Int, [74](#)
 - BEvent1Pipe, [75](#)
 - BEventPipe, [77](#)
 - BFifo, [79](#)
 - BFifoCirc, [82](#)
 - BList, [94](#)
 - BPoll, [132](#)
 - BQueue, [134](#)
 - BString, [157](#)
 - BTable, [164](#)
 - BTimer, [170](#)
 - BTimeStamp, [173](#)
 - BTimeStampMs, [177](#)
 - Tms::CycleParamEdit, [189](#)
 - Tms::CycleParamState, [191](#)
- clientGone
 - BoapServer, [123](#)
- ClkAdcDiv_1
 - Tms, [23](#)
- ClkAdcDiv_10
 - Tms, [23](#)
- ClkAdcDiv_100
 - Tms, [23](#)
- ClkAdcDiv_1000
 - Tms, [23](#)
- ClkAdcDiv_10000
 - Tms, [23](#)
- ClkAdcDiv_100000
 - Tms, [23](#)
- ClkAdcDiv_2
 - Tms, [23](#)
- ClkAdcDiv_20
 - Tms, [23](#)
- ClkAdcDiv_200
 - Tms, [23](#)
- ClkAdcDiv_2000
 - Tms, [23](#)
- ClkAdcDiv_20000
 - Tms, [23](#)
- ClkAdcDiv_5
 - Tms, [23](#)
- ClkAdcDiv_50
 - Tms, [23](#)
- ClkAdcDiv_500
 - Tms, [23](#)
- ClkAdcDiv_5000
 - Tms, [23](#)
- ClkAdcDiv_50000
 - Tms, [23](#)
- ClkFref
 - Tms, [23](#)
- ClkMs
 - Tms, [23](#)
- clock
 - Tms::TestCaptureInfo, [218](#)
- close
 - BConfig, [47](#)
 - BFile, [87](#)
 - BMySQL, [100](#)
 - BSocket, [147](#)
- cmd
 - BoapMc.h, [262](#)
 - BoapMcPacketHead, [115](#)
 - BoapPacketHead, [121](#)
- compare
 - BDate, [49](#)
 - BString, [157](#)
 - BTimeStamp, [173](#)
 - BTimeStampMs, [177](#)
- compareRegex
 - BString, [157](#)
- compareWild
 - BString, [157](#)
- compareWildExpression
 - BString, [157](#)
- config
 - SigGen, [210](#)
 - SigGenBeam, [211](#)
 - SigGenNoise, [212](#)
 - SigGenPulse, [213](#)
 - SigGenSine, [214](#)
 - SigGenSquare, [215](#)
- ConfigInfo
 - Tms::ConfigInfo, [181](#)
- configure
 - Tms::PuControl, [203](#)
 - Tms::TmsControl, [221](#)
- connect
 - BSocket, [147](#)
- connectService
 - BoapClientObject, [105](#)
- convert
 - BString, [157](#)
- convertHex
 - BString, [158](#)
- copy
 - BError, [68](#)
 - BErrorTime, [70](#)
 - BRefData, [136](#)
 - BString, [158](#)
- csvDecode
 - BString, [158](#)
- csvEncode

- BString, 158
- CyclePeriodAll
 - Tms, 23
- CyclePeriodCalibration
 - Tms, 23
- CyclePeriodEvent0
 - Tms, 23
- CyclePeriodEvent1
 - Tms, 23
- CyclePeriodEvent2
 - Tms, 23
- CyclePeriodEvent3
 - Tms, 23
- CyclePeriodEvent4
 - Tms, 23
- CyclePeriodEvent5
 - Tms, 23
- CyclePeriodEvent6
 - Tms, 23
- CyclePeriodEvent7
 - Tms, 23
- CyclePeriodEvent8
 - Tms, 23
- CyclePeriodEvent9
 - Tms, 23
- CycleInformation
 - Tms::CycleInformation, 181
- CycleInformationPeriod
 - Tms::CycleInformationPeriod, 182
- cycleNumber
 - Tms::CycleInformation, 181
 - Tms::DataInfo, 197
- CycleParam
 - Tms::CycleParam, 184
- CycleParamDb
 - Tms::CycleParamDb, 187
- CycleParamEdit
 - Tms::CycleParamEdit, 189
- CycleParamItem
 - Tms::CycleParamItem, 190
- CycleParamState
 - Tms::CycleParamState, 191
- CyclePeriod
 - Tms, 23
- cyclePeriod
 - Tms::CycleInformationPeriod, 182
 - Tms::CycleTypeInfoInformationPeriod, 195
 - Tms::DataInfo, 198
- cycleStartEvent
 - Tms::TmsEvent, 226
 - Tms::TmsEventServerList, 227
- cycleStop
 - Tms::TmsState, 233
- cycleStopEvent
 - Tms::TmsEvent, 226
 - Tms::TmsEventServerList, 227
- cycleType
 - Tms::CycleInformation, 181
 - Tms::CycleParam, 185
 - Tms::CycleParamItem, 190
 - Tms::CycleTypeInfoInformation, 193
 - Tms::Simulation, 216
- CycleTypeInfoInformation
 - Tms::CycleTypeInfoInformation, 193
- CycleTypeInfoInformationPeriod
 - Tms::CycleTypeInfoInformationPeriod, 194
- DGRAM
 - BSocket, 147
- DEBUG
 - Boap.cpp, 258
 - BoapSimple.cc, 264
 - SigGen.cpp, 285
- DEBUG_LOCAL
 - BoapMc.cpp, 261
- DEBUG_LOCAL1
 - BoapMc.cpp, 261
- Data
 - Tms::Data, 196
- data
 - BBuffer, 31
 - BoapMcPacket, 114
 - BoapPacket, 119
 - BRefData, 136
 - BSignal, 145
 - Tms::Simulation, 216
- DataFunctionMean
 - Tms, 24
- DataFunctionMean0
 - Tms, 24
- DataFunctionMean1
 - Tms, 24
- DataFunctionMeanAll
 - Tms, 24
- DataFunctionRaw
 - Tms, 24
- DataTypeRaw
 - Tms, 24
- dataEvent
 - Tms::TmsEvent, 226
 - Tms::TmsEventServerList, 227
- DataFunction
 - Tms, 23
- DataInfo
 - Tms::DataInfo, 197
- dataOffset
 - BObjMember, 131
- DataType
 - Tms, 24
- dataType
 - Tms::Data, 196
- DataValue
 - Tms::DataValue, 199
- dataValues
 - Tms::Data, 196
- day
 - BDate, 49

- BTimeStamp, 173
- daysInMonth
 - BDate, 49
- db
 - BMySQL, 100
- decrement
 - BCondInt, 40
 - BCondValue, 43
 - BCondWrap, 45
- defaultSize
 - BFifoCirc, 82
- del
 - BArray, 28
 - BDict, 53
 - BDictMap, 55
 - BEntryList, 65
 - BFileData, 90
 - BList, 94
 - BMySQL, 100
 - BString, 158
 - Tms::TmsEventServerList, 227
- delControllInfo
 - Tms::TmsControl, 221
- delEntry
 - Boapns::Boapns, 117
- delFd
 - BPoll, 133
- delay
 - Tms::TmsState, 233
- deleteCycleParams
 - Tms::CycleParamDb, 187
- deleteEntry
 - BEntryList, 65
- deleteFirst
 - BList, 94
- deleteLast
 - BList, 95
- deleteRef
 - BRefData, 136
- deltaX
 - Tms::DataValue, 199
- deltaY
 - Tms::DataValue, 199
- diff
 - BCondWrap, 45
- difference
 - BFifoCircPos, 85
 - BTimeStamp, 173
 - BTimeStampMs, 177
- dirname
 - BString, 158
- disableBlr
 - Tms::PupeConfig, 205
- disconnectService
 - BoapClientObject, 105
- dl1printf
 - BoapMc.cpp, 261
- dlprintf
 - BoapMc.cpp, 261
- doConnectionPriority
 - BoapServiceObject, 128
- doPing
 - BoapServiceObject, 128
- doPoll
 - BPoll, 133
- doPollEvents
 - BPoll, 133
- Double
 - BoapSimple.h, 265
- doubleInjection
 - Tms::PupeConfig, 205
 - Tms::Simulation, 217
- dprintf
 - BDebug.h, 244
 - BFifoCirc.cpp, 254
 - Boap.cpp, 258
 - BoapSimple.cc, 264
 - SigGen.cpp, 285
- dumpBacktrace
 - BDebugBacktrace, 51
- dumpBacktraceFile
 - BDebugBacktrace, 51
- dumpBacktraceStdout
 - BDebugBacktrace, 51
- dumpBacktraceSyslog
 - BDebugBacktrace, 51
- end
 - BCondResource, 42
 - BList, 95
- endTime
 - Tms::CycleInformationPeriod, 183
- entryName
 - BDir, 57
- entryStat
 - BDir, 57
- entryStat64
 - BDir, 57
- eprintf
 - BDebug.h, 244
- Error
 - BErrorTime, 70
- error
 - BDir, 57
 - BoapMc.h, 262
 - BoapMcPacketHead, 115
 - Tms::PuStatus, 209
- ErrorAccessDenied
 - BError.h, 251
- ErrorAppBase
 - BError.h, 252
- ErrorChecksum
 - BError.h, 251
- ErrorComms
 - BError.h, 251
- ErrorConfig
 - BError.h, 251

- ErrorData
 - BError.h, [251](#)
- ErrorDataPresent
 - BError.h, [252](#)
- ErrorEndOfData
 - BError.h, [252](#)
- ErrorEndOfFile
 - BError.h, [251](#)
- ErrorFile
 - BError.h, [251](#)
- ErrorFormat
 - BError.h, [251](#)
- ErrorInit
 - BError.h, [251](#)
- ErrorMisc
 - BError.h, [251](#)
- ErrorNoData
 - BError.h, [252](#)
- ErrorNotAvailable
 - BError.h, [251](#)
- ErrorNotImplemented
 - BError.h, [251](#)
- ErrorOk
 - BError.h, [251](#)
- ErrorOverrun
 - BError.h, [251](#)
- ErrorParam
 - BError.h, [251](#)
- ErrorResourceLimit
 - BError.h, [251](#)
- ErrorTimeout
 - BError.h, [251](#)
- ErrorUnderrun
 - BError.h, [251](#)
- ErrorWarning
 - BError.h, [251](#)
- errorEvent
 - Tms::TmsEvent, [226](#)
 - Tms::TmsEventServerList, [227](#)
- errors
 - Tms::Data, [196](#)
- escapeString
 - BMySQL, [100](#)
- eventQueue
 - BComms, [36](#)
- extension
 - BString, [158](#)
- fgets
 - BFile, [87](#)
- field
 - BString, [158](#)
- fields
 - BString, [158](#)
- fileName
 - BConfig, [47](#)
 - BFile, [87](#)
- filename
 - BEntryFile, [63](#)
- find
 - BDict, [53](#)
 - BEntryList, [65](#)
 - BFileData, [90](#)
 - BNameValueList, [102](#)
 - BString, [158](#)
- findPos
 - BNameValueList, [102](#)
- findReverse
 - BString, [158](#)
- findValue
 - BConfig, [47](#)
 - BEntryList, [65](#)
- firstLine
 - BString, [158](#)
- fixedLen
 - BString, [158](#)
- flush
 - BFile, [87](#)
 - BMySQL, [101](#)
- frefPhaseDelay
 - Tms::CycleParam, [185](#)
- fromBString
 - BDate.cpp, [241](#)
 - BDate.h, [241](#)
 - BDict.cpp, [245](#)
 - BDict.h, [245](#)
 - BString.cpp, [276](#)
 - BString.h, [277](#)
 - BTimeStamp.cpp, [280](#)
 - BTimeStamp.h, [281](#)
- front
 - BList, [95](#)
- function
 - BoapServer, [123](#)
 - BoapServerConnection, [125](#)
 - BRtcThreaded, [139](#)
 - BThread, [165](#)
 - Tms::DataInfo, [198](#)
- gate
 - Tms::TmsPhase, [228](#)
- gatePhase
 - Tms::CycleParamState, [192](#)
- gateWidth
 - Tms::CycleParamState, [192](#)
- generate
 - SigGen, [210](#)
 - SigGenBeam, [211](#)
 - SigGenNoise, [212](#)
 - SigGenPulse, [213](#)
 - SigGenSine, [214](#)
 - SigGenSquare, [215](#)
- generateIntegrated
 - SigGenBeam, [211](#)
- generateState
 - Tms::CycleParamEdit, [189](#)
- get
 - BList, [95](#)

- BMySql, 101
- BString, 159
- getAddress
 - BSocket, 148
- getApiVersion
 - BoapMcClientObject, 109
 - BoapMcComms, 111
- getBinary
 - BEvent1, 72
 - BEvent1Error, 73
- getCmd
 - BoapPacket, 119
- getConfiguration
 - Tms::TmsControl, 221
- getConnectionsNumber
 - BoapServer, 123
- getControllInfo
 - Tms::TmsControl, 221
- getControllList
 - Tms::TmsControl, 221
- getCycleInfo
 - Tms::TmsProcess, 231
- getCycleInformation
 - Tms::PuProcess, 207
 - Tms::TmsProcess, 231
- getCycleParams
 - Tms::CycleParamDb, 187
- getCycleTypeInfoInformation
 - Tms::TmsProcess, 231
- getCycleTypes
 - Tms::CycleParamDb, 187
- getData
 - Tms::PuProcess, 207
 - Tms::TmsProcess, 231
- getDate
 - BDate, 49
 - BTime, 167
 - BTimeStamp, 173
 - BTimeStampMs, 177
- getDebugString
 - BObj, 130
- getDefaultState
 - Tms::CycleParamEdit, 189
- getDurationString
 - BTimeStampMs, 177
- getDurationStringNoMs
 - BTimeStampMs, 177
- getElapsedTime
 - BTimer, 170
- getEntry
 - Boapns::Boapns, 117
- getEntryList
 - Boapns::Boapns, 117
- getErrorNo
 - BError, 68
 - BErrorTime, 70
- getEvent
 - BEvent1Int, 74
 - BEvent1Pipe, 75
- getEventSocket
 - BoapServer, 123
- getFd
 - BEvent1Int, 74
 - BEventPipe, 77
 - BFile, 87
 - BSocket, 148
- getFileNames
 - Tms::CycleParamDb, 187
- getHead
 - BoapServerConnection, 125
- getHexString
 - BBufferStore, 33
- getHostName
 - BoapServer, 123
 - BSocketAddressINET, 151
- getIpAddressList
 - BSocketAddressINET, 151
- getIpAddressListAll
 - BSocketAddressINET, 151
- getIpAddresses
 - BSocketAddressINET, 151
- getMTU
 - BSocket, 148
- getMasterPuChannel
 - Tms::PuControl, 203
- getMember
 - BObj, 130
- getMembers
 - BObj, 130
- getMicroSeconds
 - BDuration, 59
- getName
 - BEntry, 61
 - BNameValue, 102
- getNewName
 - Boapns::Boapns, 117
- getNextId
 - BFileData, 90
- getNumber
 - BError, 68
- getPollFds
 - BPoll, 133
- getPollFdsNum
 - BPoll, 133
- getPos
 - BBufferStore, 33
- getPuChannel
 - Tms::TmsControl, 222
- getPupeConfig
 - Tms::PuControl, 203
 - Tms::TmsControl, 222
- getReceiveFd
 - BEvent1Pipe, 75
- getSeconds
 - BDuration, 59
 - BTime, 167

- getServiceName
 - BoapClientObject, 105
- getSimulation
 - Tms::TmsControl, 222
- getSockOpt
 - BSocket, 148
- getSocket
 - BoapServer, 123
 - BoapServerConnection, 126
- getStates
 - Tms::CycleParamEdit, 189
- getStatistics
 - Tms::PuControl, 203
 - Tms::TmsControl, 222
- getStatus
 - Tms::PuControl, 203
 - Tms::PuProcess, 207
 - Tms::TmsControl, 222
- getString
 - BDate, 49
 - BDuration, 59
 - BEntryList, 66
 - BError, 68
 - BErrorTime, 70
 - BSocketAddressINET, 151
 - BTime, 167
 - BTimeStamp, 173
 - BTimeStampMs, 178
 - Tms::CycleParamEdit, 189
 - Tms::CycleParamState, 191
- getStringFormatted
 - BDate, 49
 - BTimeStamp, 173
- getStringNoMs
 - BTimeStamp, 173
 - BTimeStampMs, 178
- getStringRaw
 - BTimeStampMs, 178
- getThread
 - BThread, 165
- getTime
 - BDebug.cpp, 242
 - BDebug.h, 244
 - BErrorTime, 70
 - BTime, 167
 - BTimer, 170
- getTimeout
 - BCondInt.cpp, 238
- getTokenList
 - BString, 159
- getType
 - BEvent1, 72
 - BObj, 130
- getValue
 - BAtomic, 29
 - BAtomicCount, 30
 - BEntry, 61
 - BNameValue, 102
 - BSEma, 141
 - BSemaphore, 143
- getVersion
 - Boapns::Boapns, 117
 - Tms::PuControl, 204
 - Tms::PuProcess, 207
 - Tms::TmsControl, 222
 - Tms::TmsProcess, 232
- getYearMicroSeconds
 - BTimeStamp, 173
- getYearMilliSeconds
 - BTimeStampMs, 178
- getYearSeconds
 - BTimeStamp, 173
 - BTimeStampMs, 178
- getDefaultPickupPositions
 - Tms::CycleParamEdit, 189
- gettid
 - BDebug.cpp, 242
 - BDebug.h, 244
- gmatch
 - BString.cpp, 276
- goTo
 - BList, 95
- handleReconnect
 - BoapClientObject, 105
- harmonic
 - Tms::CycleInformationPeriod, 183
 - Tms::CycleTypeInfoInformationPeriod, 195
 - Tms::PuStateTable, 208
- has
 - BList, 95
- hasKey
 - BDict, 53
 - BDictMap, 55
- hash
 - BString, 159
- hashAdd
 - BDict, 53
- hashDelete
 - BDict, 53
- hashFind
 - BDict, 53
- hashPrint
 - BDict, 53
- hchange
 - Tms::TmsState, 233
- hd32
 - BDebug.cpp, 242
 - BDebug.h, 244
- hd8
 - BDebug.cpp, 242
 - BDebug.h, 244
- hd8a
 - BDebug.cpp, 242
 - BDebug.h, 244
- hda32
 - BDebug.cpp, 242

- hda8
 - BDebug.cpp, 243
 - BDebug.h, 244
- hds32
 - BDebug.h, 244
- head
 - BoapMcPacket, 114
- hostName
 - Boapns::BoapEntry, 107
- hour
 - BDuration, 59
 - BTimeStamp, 173
 - BTimeStampMs, 179
- htobe
 - BEndian.h, 249
- htobe16
 - BEndian.h, 248
- htobe32
 - BEndian.h, 248
- htobe64
 - BEndian.h, 248
- htole
 - BEndian.h, 249, 250
- htole16
 - BEndian.h, 248
- htole32
 - BEndian.h, 248
- htole64
 - BEndian.h, 248
- IP_MTU
 - BSocket.cpp, 274
- IS_BIG_ENDIAN
 - Boap.cpp, 259
- id
 - BSignal, 145
- inString
 - BString, 159
- inUse
 - BCondResource, 42
- increment
 - BCondInt, 40
 - BCondValue, 43
 - BCondWrap, 46
 - BFifoCircPos, 85
- info
 - Tms::CycleParam, 185
 - Tms::CycleTypeInfo, 194
- init
 - BComms, 36
 - BoapServer, 123
 - BoapServerConnection, 126
 - BRtc, 137
 - BRtcThreaded, 139
 - BSocket, 148
 - BSpi, 153
 - BString, 159
 - Tms::PuControl, 204
 - Tms::TmsControl, 223
- injection
 - Tms::TmsState, 233
- insert
 - BArray, 28
 - BDict, 53
 - BEntryList, 66
 - BList, 95
 - BMySQL, 101
 - BString, 159
- insertAfter
 - BList, 95
- Int16
 - BoapSimple.h, 265
- Int32
 - BoapSimple.h, 265
- Int8
 - BoapSimple.h, 265
- internalTimingMask
 - Tms::PupeConfig, 205
- isEnd
 - BDictMap, 55
 - BFile, 87
 - BList, 95
- isLeap
 - BDate, 50
 - BTimeStamp, 174
 - BTimeStampMs, 178
- isLeapYear
 - BTime, 168
- isOpen
 - BFile, 87
- isSet
 - BDate, 50
 - BEntryList, 66
 - BTime, 168
 - BTimeStamp, 174
- isSpace
 - BString, 159
- item
 - BList::Node, 200
- iterator
 - BDict, 53
 - BDictMap, 55
- justify
 - BString, 159
- key
 - BDict, 53
 - BDictItem, 54
 - BDictMap, 55
- le16toh
 - BEndian.h, 248
- le32toh
 - BEndian.h, 249
- le64toh
 - BEndian.h, 249
- len

- BRefData, 136
- BSocketAddress, 149
- BString, 159
- BStringLocked, 162
- length
 - BFile, 87
 - BoapMc.h, 262
 - BoapMcPacketHead, 115
 - BoapPacketHead, 121
- letoh
 - BEndian.h, 250
- line
 - BEntry, 61
- listen
 - BSocket, 148
- lo1
 - Tms::TmsPhase, 228
- lo1Harmonic
 - Tms::CycleParamState, 192
- lo1Phase
 - Tms::CycleParamState, 192
- lo2
 - Tms::TmsPhase, 228
- lo2Harmonic
 - Tms::CycleParamState, 192
- lo2Phase
 - Tms::CycleParamState, 192
- lock
 - BCondResource, 42
 - BMutex, 98
 - BMutexLock, 99
- locked
 - BCondResource, 42
- lowerFirst
 - BString, 159
- MDEBUG
 - BMutex.cpp, 257
- MINUS
 - BString.cpp, 275
- main
 - test1.cpp, 286
- mapCircularBuffer
 - BFifoCirc, 82
- mean1Mask
 - Tms::CycleParamState, 192
- mean2Mask
 - Tms::CycleParamState, 193
- meanFilter1
 - Tms::TmsPhase, 228
- meanFilter2
 - Tms::TmsPhase, 228
- membersPrint
 - BObj, 130
- microSecond
 - BDuration, 60
 - BTimeStamp, 174
- milliSecond
 - BTimeStampMs, 179
- minute
 - BDuration, 60
 - BTimeStamp, 174
 - BTimeStampMs, 179
- Mode
 - BSpi, 152
- Mode0
 - BSpi, 152
- Mode1
 - BSpi, 152
- Mode2
 - BSpi, 152
- Mode3
 - BSpi, 152
- moduleNum
 - Tms::PuChannel, 201
- mon_yday
 - BDate.cpp, 241
 - BTimeStamp.cpp, 280
 - BTimeStampMs.cpp, 281
- monDays
 - BTime.cpp, 279
- month
 - BDate, 50
 - BTimeStamp, 174
- NOTHEADS
 - BoapServer, 123
- NType
 - BSocket, 147
- name
 - Boapns::BoapEntry, 107
 - BoapServiceObject, 128
 - BObjMember, 131
 - Tms::CycleParam, 185
 - Tms::NameValue, 200
- NameValue
 - Tms::NameValue, 200
- nbytes
 - BoapPacket, 119
- newConnection
 - BoapServer, 123
- next
 - BDictMap, 56
 - BList, 95
 - BNode, 103
- nextFd
 - BPoll, 133
- nextId
 - BSignal, 145
- Node
 - BList::Node, 200
- nodeCreate
 - BList, 96
- nodeGet
 - BList, 96
- None
 - BErrorTime, 70
- Normal

- BMutex, 98
- nprintf
 - BDebug.h, 244
- num
 - BError, 68
 - Tms::CycleParamState, 193
- NumChannels
 - BSignal, 145
- numBunches
 - Tms::CycleInformationPeriod, 183
 - Tms::CycleTypeInfoInformationPeriod, 195
 - Tms::Data, 196
 - Tms::PuStateTable, 208
- numChannels
 - Tms::Data, 196
- numRepeat
 - BSignal, 145
- numSamples
 - BSignal, 145
- numValues
 - Tms::CycleInformationPeriod, 183
 - Tms::Data, 196
 - Tms::DataInfo, 198
- number
 - BArray, 28
 - BList, 96
- oaddress
 - BSocketAddress, 150
- oaddressFrom
 - BoapMcClientObject, 109
 - BoapMcComms, 112
- oaddressTo
 - BoapMcClientObject, 109
 - BoapMcComms, 112
- oamplitude
 - SigGenBeam, 211
 - SigGenNoise, 213
 - SigGenPulse, 214
 - SigGenSine, 215
 - SigGenSquare, 216
- oapiVersion
 - BoapClientObject, 106
 - BoapMcClientObject, 109
 - BoapMcComms, 112
 - BoapMcServiceObject, 116
 - BoapServiceObject, 128
- oarg
 - BEvent, 71
- oaverage
 - BTimer, 170
- obaseDir
 - Tms::CycleParamDb, 187
- oblr
 - SigGenBeam, 211
- oboapNs
 - BoapServer, 124
- oboapServer
 - BoapServerConnection, 126
- oboapns
 - BoapServer, 124
- obunchSet
 - SigGenBeam, 212
- oclientGoneEvent
 - BoapServer, 124
- oclients
 - BoapServer, 124
- ocmd
 - BoapFuncEntry, 108
- ocolumnWidths
 - BTable, 164
- ocomments
 - BEntryFile, 64
- ocomms
 - BoapMcClientObject, 109
 - BoapMcComms, 112
 - BoapMcSignalObject, 116
- ocond
 - BCond, 37
 - BCondBool, 39
 - BCondInt, 41
 - BCondResource, 42
 - BCondValue, 44
 - BCondWrap, 46
 - BRtcThreaded, 139
- oconnected
 - BoapClientObject, 106
- odata
 - BBuffer, 32
 - BFifo, 80
 - BFifoCirc, 83
 - BoapPacket, 120
 - BRefData, 136
 - BTable, 164
- odataSize
 - BBuffer, 32
- odb
 - BMySQL, 101
- odebug
 - BMySQL, 101
- odev
 - BSpi, 153
- odevName
 - BSpi, 153
- odirname
 - BDir, 58
- oendTime
 - BTimer, 170
- oerrNo
 - BError, 69
 - BErrorTime, 70
- oerrStr
 - BError, 69
 - BErrorTime, 70
- oerrTime
 - BErrorTime, 70
- oerror

- BDir, 58
- oevent
 - BComms, 36
- oeventNum
 - BComms, 36
- oeventQueue
 - BComms, 36
- oeventServers
 - Tms::TmsEventServerList, 227
- ofd
 - BRtc, 138
- ofds
 - BEvent1Int, 74
 - BEvent1Pipe, 76
 - BEventPipe, 77
 - BPoll, 133
- ofdsNext
 - BPoll, 133
- ofdsNum
 - BPoll, 133
- ofile
 - BConfig, 48
 - BFile, 89
- ofileName
 - BConfig, 48
 - BFile, 89
- ofilename
 - BEntryFile, 64
 - BFileData, 90
- ofref
 - SigGenBeam, 212
- ofreq
 - SigGenPulse, 214
 - SigGenSine, 215
 - SigGenSquare, 216
- ofunc
 - BoapFuncEntry, 108
- ofuncList
 - BoapServiceObject, 128
- oharmonic
 - SigGenBeam, 212
- ohashLists
 - BDict, 54
- ohashSize
 - BDict, 54
- ohostName
 - BoapServer, 124
- ohour
 - BDuration, 60
 - BTimeStamp, 175
- oi
 - Blter, 91
- oinit
 - BUrl, 180
- oisBoapns
 - BoapServer, 124
- olastPos
 - BEntryList, 66
- olen
 - BRefData, 136
 - BSocketAddress, 150
- olength
 - BList, 97
- olock
 - BCondResource, 42
 - BConfig, 48
 - BFifo, 80
 - BFifoCirc, 83
 - BMutexLock, 100
 - BMysql, 101
 - BoapClientObject, 106
 - BQueue, 135
 - BRWLock, 140
 - BSemaphoreCount, 144
 - BStringLocked, 163
 - BTimer, 170
 - Tms::TmsEventServerList, 227
- olockCall
 - BoapMcComms, 113
- olockTx
 - BoapMcComms, 113
- omaxLength
 - BoapClientObject, 106
 - BoapServerConnection, 126
- omicroSecond
 - BDuration, 60
 - BTimeStamp, 175
- ominute
 - BDuration, 60
 - BTimeStamp, 175
- omode
 - BFile, 89
- omutex
 - BCond, 37
 - BCondBool, 39
 - BCondInt, 41
 - BCondResource, 42
 - BCondValue, 44
 - BCondWrap, 46
 - BMutex, 99
- oname
 - BEntry, 62
 - BNameValue, 102
 - BoapClientObject, 106
 - BoapServiceObject, 128
- onbytes
 - BoapPacket, 120
- onet
 - BoapServer, 124
- onetEvent
 - BoapServer, 124
- onetEventAddress
 - BoapServer, 124
- onodes
 - BList, 97
- onum

- BTimer, 170
- onumOperations
 - BoapServer, 124
- onumber
 - BQueue, 135
- oobject
 - BoapServiceEntry, 127
- ooffset
 - SigGenSquare, 216
- oonTime
 - SigGenPulse, 214
- oopened
 - BMysql, 101
- opacket
 - BoapMcClientObject, 109
 - BoapMcComms, 113
- opacketMode
 - BComms, 36
- opacketReqQueue
 - BoapMcComms, 113
- opacketReqRx
 - BoapMcComms, 113
- opacketReqTx
 - BoapMcComms, 113
- opacketRx
 - BoapMcComms, 113
- opacketRxSema
 - BoapMcComms, 113
- opacketTx
 - BoapMcComms, 113
- opacketTxQueue
 - BoapMcComms, 113
- opacketTxQueueWriteNum
 - BoapMcComms, 113
- opacketTxSema
 - BoapMcComms, 113
- opeak
 - BTimer, 170
- open
 - BConfig, 47
 - BDir, 57
 - BEntryFile, 63
 - BFile, 87, 88
 - BFileData, 90
 - BMysql, 101
- operator BNode *
 - Blter, 91
- operator BString
 - BDate, 50
 - BStringLocked, 162
 - BTimeStamp, 174
- operator const char *
 - BString, 159
- operator const SocketAddr *
 - BSocketAddress, 150
- operator int
 - BCondBool, 38
 - BError, 68
 - BErrorTime, 70
 - BFifoCircPos, 85
- operator long
 - BAtomicCount, 30
- operator Type
 - BAtomic, 29
- operator <
 - BDate, 50
 - BString, 160
 - BTime, 168
 - BTimeStamp, 174
 - BTimeStampMs, 178
- operator <<
 - BString.cpp, 276
 - BString.h, 277
- operator <=
 - BDate, 50
 - BString, 160
 - BTime, 168
 - BTimeStamp, 174
 - BTimeStampMs, 178
- operator >
 - BDate, 50
 - BString, 160
 - BTime, 168
 - BTimeStamp, 174
 - BTimeStampMs, 178
- operator >>
 - BString.cpp, 276
 - BString.h, 277
- operator >=
 - BDate, 50
 - BString, 160
 - BTime, 168
 - BTimeStamp, 174
 - BTimeStampMs, 178
- operator+
 - BDict, 53
 - BList, 96
 - BString, 159, 160
 - BStringLocked, 163
 - BTime, 168
- operator++
 - BAtomic, 29
 - BAtomicCount, 30
 - BCondInt, 40
 - BCondValue, 43
 - BCondWrap, 46
- operator+=
 - BCondInt, 40
 - BCondValue, 44
 - BCondWrap, 46
 - BFifoCircPos, 85
 - BString, 160
 - BTime, 168
- operator--
 - BAtomic, 29
 - BAtomicCount, 30

- BCondInt, 40
- BCondValue, 44
- BCondWrap, 46
- operator==
 - BCondInt, 40
 - BCondValue, 44
 - BCondWrap, 46
- operator=
 - BDict, 53
 - BEntryList, 66
 - BFile, 88
 - BList, 96
 - BMutex, 98
 - BRefData, 136
 - BRWLock, 140
 - BSema, 141
 - BSemaphore, 143
 - BSemaphoreCount, 144
 - BSignal, 145
 - BSocketAddress, 150
 - BString, 160
 - BStringLocked, 163
 - BTimeStamp, 174
- operator===
 - BDate, 50
 - BFifoCircPos, 85
 - BIter, 91
 - BSocketAddress, 150
 - BString, 160
 - BTime, 168
 - BTimeStamp, 174
- opolicy
 - BThread, 166
- opoll
 - BoapServer, 124
- opos
 - BBufferStore, 34
 - BFifoCircPos, 85
 - BoapPacket, 120
- opriority
 - BoapClientObject, 106
 - BThread, 166
- orate
 - BRtc, 138
 - BRtcThreaded, 139
- orbitNumber
 - Tms::DataInfo, 198
- oreadPos
 - BFifo, 80
 - BFifoCirc, 83
- oreconnect
 - BoapClientObject, 106
- oreduce
 - SigGenBeam, 212
- orefCount
 - BRefData, 136
- ores
 - BUrl, 180
- orresult
 - BThread, 166
- ortc
 - BRtcThreaded, 139
- orunning
 - BThread, 166
- orx
 - BoapClientObject, 106
 - BoapServer, 124
 - BoapServerConnection, 126
 - BoapSignalObject, 129
- osampleRate
 - SigGen, 210
- osecond
 - BDuration, 60
 - BTimeStamp, 175
- osema
 - BSema, 142
 - BSemaphore, 143
 - BSemaphoreCount, 144
- oseparator
 - BFileCsv, 89
- oserver
 - BoapServiceObject, 128
- oservice
 - BoapClientObject, 106
 - BoapServiceEntry, 127
- oservices
 - BoapServer, 124
- osize
 - BBuffer, 32
 - BFifo, 80
 - BFifoCirc, 83
 - BFifoCircPos, 85
 - BoapPacket, 120
 - BQueue, 135
- oslave
 - BoapMcComms, 113
- osocket
 - BoapServerConnection, 126
 - BSocket, 148
- osort
 - BDir, 58
- ospare
 - BDuration, 60
 - BTimeStamp, 175
- ostackSize
 - BThread, 166
- ostartTime
 - BTimer, 170
 - SigGenPulse, 214
- ostr
 - BString, 162
 - BStringLocked, 163
- oswapBytes
 - BBufferStore, 34
- othread
 - BThread, 166

- othreaded
 - BoapMcComms, 114
 - BoapServer, 124
- otime
 - BTime, 168
- otimeout
 - BComms, 36
 - BoapClientObject, 106
 - BoapMcComms, 114
- otitle
 - BTable, 164
- otx
 - BoapClientObject, 106
 - BoapServer, 124
 - BoapServerConnection, 126
 - BoapSignalObject, 129
- otype
 - BEvent, 71
 - BEvent1, 73
- ouse
 - BCondResource, 42
- ovalue
 - BAtomic, 29
 - BAtomicCount, 30
 - BCondBool, 39
 - BCondInt, 41
 - BCondValue, 44
 - BCondWrap, 46
 - BEntry, 62
 - BNameValue, 102
 - BSemaphoreCount, 144
- overview.dox, 285
- ovmSize
 - BFifoCirc, 84
- owild
 - BDir, 58
- owriteNumFifoSamples
 - BFifoCirc, 84
- owritePos
 - BFifo, 80
 - BFifoCirc, 84
- ox
 - SigGen, 210
- oyday
 - BDate, 51
 - BTimeStamp, 175
- oyear
 - BDate, 51
 - BTimeStamp, 175
- packetMode
 - BComms, 36
- packetRecv
 - BoapMcComms, 111
- packetSend
 - BoapMcComms, 111
- pad
 - BString, 160
- peak
 - BTimer, 170
- peekHead
 - BoapPacket, 119
- performCall
 - BoapClientObject, 105
 - BoapMcClientObject, 109
 - BoapMcComms, 111
- performRecv
 - BoapClientObject, 105
 - BoapMcClientObject, 109
- performSend
 - BoapClientObject, 105
 - BoapMcClientObject, 109
 - BoapMcComms, 111
 - BoapMcSignalObject, 116
 - BoapSignalObject, 129
- period
 - Tms::CycleParamState, 193
 - Tms::PuStateTable, 208
- periods
 - Tms::CycleInformation, 181
 - Tms::CycleTypeInfoInformation, 194
- phaseTable
 - Tms::PuStateTable, 208
- ping
 - BoapClientObject, 105
- pingLocked
 - BoapClientObject, 106
- pllCycleStartFrequency
 - Tms::CycleParam, 185
- pllDdsMaximum
 - Tms::CycleParam, 185
- pllDdsMinimum
 - Tms::CycleParam, 185
- pllFeedbackSelect
 - Tms::TmsState, 233
- pllFrefGain
 - Tms::CycleParam, 185
- pllGain
 - Tms::CycleParam, 185
- pllInitialFrequency
 - Tms::CycleParam, 185
- pllInitialFrequencyDelay
 - Tms::CycleParam, 185
- pllLO1FromAddress
 - Tms::TmsState, 233
- pllLO2FromAddress
 - Tms::TmsState, 233
- pllReference1
 - Tms::TmsState, 233
- pllReference2
 - Tms::TmsState, 233
- PollFd
 - BPoll, 132
- pop
 - BBufferStore, 33, 34
 - BList, 96
 - BoapPacket, 119

- popHead
 - BoapPacket, 119
- port
 - Boapns::BoapEntry, 107
 - BSocketAddressINET, 151
- pos
 - BFifoCircPos, 85
- position
 - BFile, 88
 - BList, 96
- post
 - BSema, 141
- postTriggerDelay
 - Tms::TestCaptureInfo, 218
- prev
 - BList, 96
 - BNode, 103
- print
 - BEntry, 62
 - BEntryList, 66
 - BTable, 164
- printCycleParams
 - test1.cpp, 286
- printLine
 - BTable, 164
- printf
 - BFile, 88
 - BString, 160
- Priority
 - BSocket, 147
 - Tms, 24
- PriorityHigh
 - BSocket, 147
 - Tms, 24
- PriorityLow
 - BSocket, 147
 - Tms, 24
- PriorityNormal
 - BSocket, 147
 - Tms, 24
- process
 - BoapMcServiceObject, 115
 - BoapServer, 123
 - BoapServerConnection, 126
 - BoapServiceObject, 128
- processEvent
 - BoapMcServiceObject, 115
 - BoapServer, 123, 124
 - BoapServiceObject, 128
- processPacket
 - BoapMcComms, 112
- processRequest
 - BoapMcComms, 112
- processRequests
 - BoapMcComms, 112
- processRx
 - BoapMcComms, 112
- PuChannel
 - Tms::PuChannel, 201
- PuControl
 - Tms::PuControl, 203
- PuProcess
 - Tms::PuProcess, 207
- puReferences
 - Tms::ConfigInfo, 181
- puServerStarted
 - Tms::TmsControl, 223
- PuStateTable
 - Tms::PuStateTable, 208
- PuStatus
 - Tms::PuStatus, 209
- pullLine
 - BString, 160
- pullSeparators
 - BString, 160
- pullToken
 - BString, 160
- pullWord
 - BString, 160
- pupeChan
 - Tms::PuChannel, 201
- PupeConfig
 - Tms::PupeConfig, 205
- pupeNum
 - Tms::PuChannel, 201
- push
 - BBufferStore, 34
 - BList, 96
 - BoapPacket, 119, 120
- pushHead
 - BoapPacket, 120
- query
 - BMySQL, 101
- queueAdd
 - BList, 96
- queueGet
 - BList, 96
- raw
 - BSocketAddress, 150
- rdLock
 - BRWLock, 140
- read
 - BComms, 36
 - BConfig, 48
 - BDir, 57
 - BEntryFile, 63
 - BEventPipe, 77
 - BFifo, 79
 - BFifoCirc, 82
 - BFile, 88
 - BFileData, 90
 - BQueue, 134
- readAvailable
 - BComms, 36
 - BEventPipe, 77

- BFifo, 79
- BFifoCirc, 82
- BQueue, 134
- readAvailableChunk
 - BFifo, 79
- readCsv
 - BFileCsv, 89
- readCycleParams
 - Tms::CycleParamDb, 187
- readData
 - BFifo, 79
 - BFifoCirc, 82
- readDone
 - BFifo, 79
 - BFifoCirc, 83
- readFromFile
 - Tms::CycleParamEdit, 189
- readPos
 - BFifo, 79
- readString
 - BFile, 88
 - BUrl, 180
- readWaitAvailable
 - BFifoCirc, 83
- rear
 - BArray, 28
 - BList, 97
- Recursive
 - BMutex, 98
- recv
 - BSocket, 148
- recvFrom
 - BSocket, 148
- recvFromWithTimeout
 - BSocket, 148
- recvWithTimeout
 - BSocket, 148
- removeNL
 - BString, 160
- removeSeparators
 - BString, 161
- requestData
 - Tms::PuProcess, 207
 - Tms::TmsProcess, 232
- reserved
 - BoapPacketHead, 121
- resize
 - BBuffer, 31
 - BFifo, 79
 - BoapPacket, 120
- result
 - BThread, 166
- retDouble
 - BString, 161
- retInt
 - BString, 161
- retStr
 - BString, 161
- retStrDup
 - BString, 161
- retUInt
 - BString, 161
- reverse
 - BString, 161
- ring
 - Tms::CycleParam, 186
 - Tms::CycleParamItem, 190
- roundSize
 - BBuffer.cpp, 236
 - BoapSimple.cc, 264
- run
 - BoapServer, 124
- running
 - BThread, 166
 - Tms::PuStatus, 209
- STREAM
 - BSocket, 147
- STRBUF
 - BFile.cpp, 255
- STRBUF_SIZE
 - BDebug.cpp, 243
- STRIP
 - BString.cpp, 275
- Sample
 - SigGen.h, 286
- sampleNumber
 - BTimeStampMs, 179
- second
 - BDuration, 60
 - BTimeStamp, 174
 - BTimeStampMs, 179
- seek
 - BFile, 88
- send
 - BSocket, 148
- sendEvent
 - BEvent1Int, 74
 - BEvent1Pipe, 76
 - BoapMcServiceObject, 115
 - BoapServer, 124
 - BoapServiceObject, 128
- sendTo
 - BSocket, 148
- service
 - Boapns::BoapEntry, 107
 - BoapPacketHead, 121
- set
 - BCondBool, 38
 - BDate, 50
 - BDuration, 60
 - BError, 68
 - BErrorTime, 70
 - BFifoCircPos, 85
 - BSemaphore, 143
 - BSocketAddress, 150
 - BSocketAddressINET, 152

- BTime, 168
- BTimeStamp, 174
- setAddress
 - BoapMcClientObject, 109
 - BoapMcComms, 112
- setBinary
 - BEvent1, 72
 - BEvent1Error, 73
- setBroadCast
 - BSocket, 148
- setComms
 - BoapMcComms, 112
- setCommsMode
 - BoapMcComms, 112
- setConnectionPriority
 - BoapClientObject, 106
- setControllInfo
 - Tms::PuControl, 204
 - Tms::TmsControl, 223
- setCycleParams
 - Tms::CycleParamDb, 187
- setData
 - BBuffer, 31
 - BoapPacket, 120
- setDebug
 - BDebug.cpp, 243
 - BDebug.h, 244
 - BMySQL, 101
- setDurationString
 - BTimeStampMs, 178
- setError
 - BError, 68
- setFd
 - BSocket, 148
- setFirst
 - BDate, 50
 - BTimeStamp, 174
- setHexString
 - BBufferStore, 34
- setInitPriority
 - BThread, 166
- setInitStackSize
 - BThread, 166
- setLast
 - BDate, 50
 - BTimeStamp, 174
- setLen
 - BRefData, 136
- setLine
 - BEntry, 62
- setMaxLength
 - BoapClientObject, 106
 - BoapServerConnection, 126
- setMember
 - BObj, 130
- setMembers
 - BObj, 131
- setName
 - BEntry, 62
 - BoapServiceObject, 128
- setNext
 - Tms::CycleParamState, 192
- setNextCycle
 - Tms::PuControl, 204
 - Tms::Simulation, 217
 - Tms::TmsControl, 223
- setNow
 - BDate, 50
 - BTimeStamp, 174
 - BTimeStampMs, 178
- setPacketMode
 - BComms, 36
- setPort
 - BSocketAddressINET, 152
- setPos
 - BBufferStore, 34
- setPriority
 - BSocket, 148
 - BThread, 166
- setProcessPriority
 - Tms::PuControl, 204
 - Tms::TmsControl, 223
- setPupeConfig
 - Tms::PuControl, 204
 - Tms::TmsControl, 224
- setReuseAddress
 - BSocket, 148
- setSimulation
 - Tms::TmsControl, 224
- setSize
 - BBuffer, 31
 - BFifoCircPos, 85
- setSockOpt
 - BSocket, 148
- setSort
 - BDir, 57
- setStates
 - Tms::CycleParamEdit, 189
- setString
 - BDate, 50
 - BDuration, 60
 - BTime, 168
 - BTimeStamp, 174
 - BTimeStampMs, 178
 - Tms::CycleParamEdit, 189
 - Tms::CycleParamState, 192
- setTestData
 - Tms::PuControl, 204
 - Tms::TmsControl, 224
- setTestMode
 - Tms::PuControl, 204
 - Tms::TmsControl, 224
- setTime
 - BTimeStamp, 175
- setTimeout
 - BComms, 36

- BoapClientObject, 106
- BoapMcComms, 112
- setTimingSignals
 - Tms::PuControl, 204
 - Tms::TmsControl, 224
- setTitle
 - BTable, 164
- setVBuf
 - BFile, 88
- setValue
 - BCondInt, 40
 - BCondValue, 44
 - BCondWrap, 46
 - BEntry, 62
 - BEntryList, 66
 - BSemaphoreCount, 144
- setValueRaw
 - BEntryList, 66
- setWild
 - BDir, 58
- setYDay
 - BDate, 50
 - BTimeStamp, 175
- setYearDay
 - BTime, 168
- settings
 - Tms::CycleParam, 186
- shutdown
 - BSocket, 148
- SigGen, 210
 - ~SigGen, 210
 - config, 210
 - generate, 210
 - osampleRate, 210
 - ox, 210
 - SigGen, 210
 - SigGen, 210
- SigGen.cpp, 285
 - DEBUG, 285
 - dprintf, 285
- SigGen.h, 285
 - BSignalList, 286
 - Sample, 286
- SigGenBeam, 211
 - ~SigGenBeam, 211
 - config, 211
 - generate, 211
 - generateIntegrated, 211
 - oamplitude, 211
 - oblr, 211
 - obunchSet, 212
 - ofref, 212
 - oharmonic, 212
 - oreduce, 212
 - SigGenBeam, 211
 - SigGenBeam, 211
- SigGenNoise, 212
 - ~SigGenNoise, 212
- config, 212
- generate, 212
- oamplitude, 213
- SigGenNoise, 212
- SigGenNoise, 212
- SigGenPulse, 213
 - ~SigGenPulse, 213
 - config, 213
 - generate, 213
 - oamplitude, 214
 - ofreq, 214
 - oonTime, 214
 - ostartTime, 214
 - SigGenPulse, 213
 - SigGenPulse, 213
- SigGenSine, 214
 - ~SigGenSine, 214
 - config, 214
 - generate, 214
 - oamplitude, 215
 - ofreq, 215
 - SigGenSine, 214
 - SigGenSine, 214
- SigGenSquare, 215
 - ~SigGenSquare, 215
 - config, 215
 - generate, 215
 - oamplitude, 216
 - ofreq, 216
 - ooffset, 216
 - SigGenSquare, 215
 - SigGenSquare, 215
- sigma
 - Tms::DataValue, 199
- signal
 - BCond, 37
- Simulation
 - Tms::Simulation, 216
- size
 - BBuffer, 31
 - BDictMap, 56
 - BFifo, 79
 - BFifoCirc, 83
 - BList, 97
 - BObjMember, 131
- SockAddr
 - BSocketAddress, 149
- SockAddrIP
 - BSocketAddressINET, 151
- sort
 - BArray, 28
 - BList, 97
- SortFunc
 - BArray, 28
 - BList, 94
- source
 - Tms::TestCaptureInfo, 218
- spare

- Tms::TmsPhase, 228
- split
 - BString, 161
- start
 - BCondResource, 42
 - BDictMap, 56
 - BList, 97
 - BThread, 166
 - BTimer, 170
- startFunc
 - BThread, 166
- startTime
 - Tms::CycleInformationPeriod, 183
 - Tms::DataInfo, 198
 - Tms::TestCaptureInfo, 218
- state
 - Tms::CycleParamState, 193
 - Tms::PuStateTable, 208
- stateDelay
 - Tms::CycleParam, 186
- stateTable
 - Tms::CycleParam, 186
- stop
 - BTimer, 170
- str
 - BError, 68
- subMilliseconds
 - BTimeStampMs, 178
- subSeconds
 - BTimeStampMs, 178
- subString
 - BString, 161
- swap
 - BList, 97
- THREADED
 - BoapServer, 123
- table_crc_hi
 - BCrc16.cpp, 239
- table_crc_lo
 - BCrc16.cpp, 240
- take
 - BSemaphoreCount, 144
- test
 - Tms::PuControl, 204
 - Tms::TmsControl, 225
- test1.cpp, 286
 - main, 286
 - printCycleParams, 286
- TestOutputFrefLocal
 - Tms, 24
- TestOutputPIIL1
 - Tms, 24
- TestOutputPIIL2
 - Tms, 24
- TestCaptureInfo
 - Tms::TestCaptureInfo, 218
- TestOutput
 - Tms, 24
- time
 - Tms::DataValue, 199
- timedLock
 - BMutex, 99
- timedWait
 - BCond, 37
 - BCondBool, 38
 - BSema, 141
- timeoutTicks
 - BTypes.h, 284
- timing
 - Tms::Simulation, 217
- TimingSigCalStart
 - Tms, 24
- TimingSigCalStop
 - Tms, 24
- TimingSigClock
 - Tms, 24
- TimingSigCycleStart
 - Tms, 24
- TimingSigCycleStop
 - Tms, 24
- TimingSigFRef
 - Tms, 24
- TimingSigHChange
 - Tms, 24
- TimingSigInjection
 - Tms, 24
- TimingSig
 - Tms, 24
- Tms, 21
 - apiVersion, 25
 - CaptureClock, 23
 - ClkAdcDiv_1, 23
 - ClkAdcDiv_10, 23
 - ClkAdcDiv_100, 23
 - ClkAdcDiv_1000, 23
 - ClkAdcDiv_10000, 23
 - ClkAdcDiv_100000, 23
 - ClkAdcDiv_2, 23
 - ClkAdcDiv_20, 23
 - ClkAdcDiv_200, 23
 - ClkAdcDiv_2000, 23
 - ClkAdcDiv_20000, 23
 - ClkAdcDiv_5, 23
 - ClkAdcDiv_50, 23
 - ClkAdcDiv_500, 23
 - ClkAdcDiv_5000, 23
 - ClkAdcDiv_50000, 23
 - ClkFref, 23
 - ClkMs, 23
 - CyclePeriodAll, 23
 - CyclePeriodCalibration, 23
 - CyclePeriodEvent0, 23
 - CyclePeriodEvent1, 23
 - CyclePeriodEvent2, 23
 - CyclePeriodEvent3, 23
 - CyclePeriodEvent4, 23

- CyclePeriodEvent5, 23
- CyclePeriodEvent6, 23
- CyclePeriodEvent7, 23
- CyclePeriodEvent8, 23
- CyclePeriodEvent9, 23
- CyclePeriod, 23
- DataFunctionMean, 24
- DataFunctionMean0, 24
- DataFunctionMean1, 24
- DataFunctionMeanAll, 24
- DataFunctionRaw, 24
- Data TypeRaw, 24
- DataFunction, 23
- Data Type, 24
- Priority, 24
- PriorityHigh, 24
- PriorityLow, 24
- PriorityNormal, 24
- TestOutputFrefLocal, 24
- TestOutputPIL1, 24
- TestOutputPIL2, 24
- TestOutput, 24
- TimingSigCalStart, 24
- TimingSigCalStop, 24
- TimingSigClock, 24
- TimingSigCycleStart, 24
- TimingSigCycleStop, 24
- TimingSigFRef, 24
- TimingSigHChange, 24
- TimingSigInjection, 24
- TimingSig, 24
- tmsNumPickups, 25
- tmsPhaseTableSize, 25
- Tms::ConfigInfo, 180
 - ConfigInfo, 181
 - puReferences, 181
- Tms::CycleInformation, 181
 - CycleInformation, 181
 - cycleNumber, 181
 - cycleType, 181
 - periods, 181
- Tms::CycleInformationPeriod, 182
 - bunchMask, 182
 - CycleInformationPeriod, 182
 - cyclePeriod, 182
 - endTime, 183
 - harmonic, 183
 - numBunches, 183
 - numValues, 183
 - startTime, 183
- Tms::CycleParam, 183
 - channel, 185
 - CycleParam, 184
 - cycleType, 185
 - frefPhaseDelay, 185
 - info, 185
 - name, 185
 - pllCycleStartFrequency, 185
 - pllDdsMaximum, 185
 - pllDdsMinimum, 185
 - pllFrefGain, 185
 - pllGain, 185
 - pllInitialFrequency, 185
 - pllInitialFrequencyDelay, 185
 - ring, 186
 - settings, 186
 - stateDelay, 186
 - stateTable, 186
- Tms::CycleParamDb, 186
 - CycleParamDb, 187
 - deleteCycleParams, 187
 - getCycleParams, 187
 - getCycleTypes, 187
 - getFileNames, 187
 - obaseDir, 187
 - readCycleParams, 187
 - setCycleParams, 187
 - writeCycleParams, 187
- Tms::CycleParamEdit, 188
 - bunch, 189
 - clear, 189
 - CycleParamEdit, 189
 - generateState, 189
 - getDefaultState, 189
 - getStates, 189
 - getString, 189
 - getDefaultPickupPositions, 189
 - readFromFile, 189
 - setStates, 189
 - setString, 189
 - value, 189
 - writeToFile, 189
- Tms::CycleParamItem, 190
 - channel, 190
 - CycleParamItem, 190
 - cycleType, 190
 - ring, 190
- Tms::CycleParamState, 190
 - blrPhase, 192
 - blrWidth, 192
 - bunchMask, 192
 - clear, 191
 - CycleParamState, 191
 - gatePhase, 192
 - gateWidth, 192
 - getString, 191
 - lo1Harmonic, 192
 - lo1Phase, 192
 - lo2Harmonic, 192
 - lo2Phase, 192
 - mean1Mask, 192
 - mean2Mask, 193
 - num, 193
 - period, 193
 - setNext, 192
 - setString, 192

- state, 193
- Tms::CycleTypeInfo, 193
 - cycleType, 193
 - CycleTypeInfo, 193
 - info, 194
 - periods, 194
- Tms::CycleTypeInfoPeriod, 194
 - bunchMask, 194
 - cyclePeriod, 195
 - CycleTypeInfoPeriod, 194
 - harmonic, 195
 - numBunches, 195
- Tms::Data, 195
 - Data, 196
 - dataType, 196
 - dataValues, 196
 - errors, 196
 - numBunches, 196
 - numChannels, 196
 - numValues, 196
- Tms::DataInfo, 196
 - argument, 197
 - beyondPeriod, 197
 - bunchNumber, 197
 - channel, 197
 - cycleNumber, 197
 - cyclePeriod, 198
 - DataInfo, 197
 - function, 198
 - numValues, 198
 - orbitNumber, 198
 - startTime, 198
- Tms::DataValue, 198
 - DataValue, 199
 - deltaX, 199
 - deltaY, 199
 - sigma, 199
 - time, 199
- Tms::NameValue, 199
 - name, 200
 - NameValue, 200
 - value, 200
- Tms::PuChannel, 201
 - moduleNum, 201
 - PuChannel, 201
 - pupeChan, 201
 - pupeNum, 201
- Tms::PuControl, 202
 - captureDiagnostics, 203
 - configure, 203
 - getMasterPuChannel, 203
 - getPupeConfig, 203
 - getStatistics, 203
 - getStatus, 203
 - getVersion, 204
 - init, 204
 - PuControl, 203
 - setControlInfo, 204
 - setNextCycle, 204
 - setProcessPriority, 204
 - setPupeConfig, 204
 - setTestData, 204
 - setTestMode, 204
 - setTimingSignals, 204
 - test, 204
- Tms::PuProcess, 206
 - addEventServer, 207
 - getCycleInformation, 207
 - getData, 207
 - getStatus, 207
 - getVersion, 207
 - PuProcess, 207
 - requestData, 207
- Tms::PuStateTable, 207
 - bunchMask, 208
 - harmonic, 208
 - numBunches, 208
 - period, 208
 - phaseTable, 208
 - PuStateTable, 208
 - state, 208
- Tms::PuStatus, 209
 - error, 209
 - PuStatus, 209
 - running, 209
- Tms::PupeConfig, 205
 - adcSysclkSync, 205
 - disableBlr, 205
 - doubleInjection, 205
 - internalTimingMask, 205
 - PupeConfig, 205
- Tms::Simulation, 216
 - cycleType, 216
 - data, 216
 - doubleInjection, 217
 - setNextCycle, 217
 - Simulation, 216
 - timing, 217
- Tms::TestCaptureInfo, 217
 - clock, 218
 - postTriggerDelay, 218
 - source, 218
 - startTime, 218
 - TestCaptureInfo, 218
 - triggerAnd, 218
 - triggerMask, 218
 - triggerSourceData, 218
 - triggerState, 218
 - triggerStateEnable, 218
 - triggerStore, 219
- Tms::TmsControl, 219
 - captureDiagnostics, 220
 - configure, 221
 - delControlInfo, 221
 - getConfiguration, 221
 - getControlInfo, 221

- getControlList, [221](#)
- getPuChannel, [222](#)
- getPupeConfig, [222](#)
- getSimulation, [222](#)
- getStatistics, [222](#)
- getStatus, [222](#)
- getVersion, [222](#)
- init, [223](#)
- puServerStarted, [223](#)
- setControlInfo, [223](#)
- setNextCycle, [223](#)
- setProcessPriority, [223](#)
- setPupeConfig, [224](#)
- setSimulation, [224](#)
- setTestData, [224](#)
- setTestMode, [224](#)
- setTimingSignals, [224](#)
- test, [225](#)
- TmsControl, [220](#)
- Tms::TmsEvent, [225](#)
 - cycleStartEvent, [226](#)
 - cycleStopEvent, [226](#)
 - dataEvent, [226](#)
 - errorEvent, [226](#)
 - TmsEvent, [226](#)
- Tms::TmsEventServerList, [226](#)
 - ~TmsEventServerList, [227](#)
 - append, [227](#)
 - cycleStartEvent, [227](#)
 - cycleStopEvent, [227](#)
 - dataEvent, [227](#)
 - del, [227](#)
 - errorEvent, [227](#)
 - oeventServers, [227](#)
 - olock, [227](#)
 - TmsEventServerList, [227](#)
- Tms::TmsPhase, [228](#)
 - blr, [228](#)
 - gate, [228](#)
 - lo1, [228](#)
 - lo2, [228](#)
 - meanFilter1, [228](#)
 - meanFilter2, [228](#)
 - spare, [228](#)
 - value, [228](#)
- Tms::TmsProcess, [228](#)
 - addEventServer, [230](#)
 - getCycleInfo, [231](#)
 - getCycleInformation, [231](#)
 - getCycleTypeInfo, [231](#)
 - getData, [231](#)
 - getVersion, [232](#)
 - requestData, [232](#)
 - TmsProcess, [229](#)
- Tms::TmsState, [232](#)
 - acquireData, [233](#)
 - bit6, [233](#)
 - bit7, [233](#)
 - calStart, [233](#)
 - calStop, [233](#)
 - cycleStop, [233](#)
 - delay, [233](#)
 - hchange, [233](#)
 - injection, [233](#)
 - pllFeedbackSelect, [233](#)
 - pllLO1FromAddress, [233](#)
 - pllLO2FromAddress, [233](#)
 - pllReference1, [233](#)
 - pllReference2, [233](#)
 - value, [233](#)
- TmsC.cc, [286](#)
- TmsC.h, [287](#)
- TmsControl
 - Tms::TmsControl, [220](#)
- TmsCycleParam.cc, [287](#)
- TmsCycleParam.h, [288](#)
- TmsD.cc, [288](#)
- TmsD.h, [288](#)
- TmsEvent
 - Tms::TmsEvent, [226](#)
- TmsEventServerList
 - Tms::TmsEventServerList, [227](#)
- TmsEventServerList.cc, [289](#)
- TmsEventServerList.h, [290](#)
- tmsFunctions.dox, [290](#)
- TmsLib.cc, [290](#)
- TmsLib.h, [290](#)
- tmsNumPickups
 - Tms, [25](#)
- tmsPhaseTableSize
 - Tms, [25](#)
- TmsProcess
 - Tms::TmsProcess, [229](#)
- TmsS.cc, [291](#)
- TmsT.cc, [291](#)
- toBDictStringFromJson
 - BObjStringFormat.cpp, [267](#)
 - BObjStringFormat.h, [269](#)
- toBString
 - BDate.cpp, [241](#)
 - BDate.h, [241](#)
 - BDict.cpp, [245](#)
 - BDict.h, [245](#)
 - BObjStringFormat.cpp, [267](#)
 - BObjStringFormat.h, [269](#)
 - BString.cpp, [276](#)
 - BString.h, [277](#), [278](#)
 - BTimeStamp.cpp, [280](#)
 - BTimeStamp.h, [281](#)
- toBStringJson
 - BObjStringFormat.cpp, [267](#), [268](#)
 - BObjStringFormat.h, [269](#), [270](#)
- toLower
 - BString, [161](#)
- toUpper
 - BString, [161](#)

- tprintf
 - BDebug.cpp, 243
 - BDebug.h, 244
- transact
 - BSpi, 153
- translateChar
 - BString, 161
- triggerAnd
 - Tms::TestCaptureInfo, 218
- triggerMask
 - Tms::TestCaptureInfo, 218
- triggerSourceData
 - Tms::TestCaptureInfo, 218
- triggerState
 - Tms::TestCaptureInfo, 218
- triggerStateEnable
 - Tms::TestCaptureInfo, 218
- triggerStore
 - Tms::TestCaptureInfo, 219
- truncate
 - BFile, 88
 - BString, 162
- tryLock
 - BMutex, 99
- tryRdLock
 - BRWLock, 140
- tryWait
 - BSema, 142
- tryWrLock
 - BRWLock, 140
- Type
 - BErrorTime, 70
 - BMutex, 98
- type
 - BEvent, 71
 - BoapPacketHead, 121
 - BObjMember, 131
- typeComp
 - BObjMember, 131
- typeName
 - BObjMember, 131
- UInt16
 - BoapSimple.h, 265
- UInt32
 - BoapSimple.h, 265
- UInt8
 - BoapSimple.h, 265
- unlock
 - BCondResource, 42
 - BMutex, 99
 - BMutexLock, 99
 - BRWLock, 140
- unmapCircularBuffer
 - BFifoCirc, 83
- update
 - BMysql, 101
- updateHead
 - BoapPacket, 120
- updateLen
 - BoapPacket, 120
- valid
 - BIter, 91
- validate
 - BoapServerConnection, 126
- value
 - BCondBool, 38
 - BCondInt, 40
 - BCondValue, 44
 - BCondWrap, 46
 - BDictItem, 54
 - BSemaphoreCount, 144
 - Tms::CycleParamEdit, 189
 - Tms::NameValue, 200
 - Tms::TmsPhase, 228
 - Tms::TmsState, 233
- Wait
 - BComms, 35
- wait
 - BComms, 36
 - BCond, 37
 - BCondBool, 38
 - BRtc, 137
 - BRtcThreaded, 139
 - BSema, 142
 - BSemaphore, 143
 - BSemaphoreCount, 144
- WaitError
 - BComms, 35
- WaitNone
 - BComms, 35
- WaitRead
 - BComms, 35
- WaitWrite
 - BComms, 35
- waitForCompletion
 - BThread, 166
- waitLessThan
 - BCondInt, 40
 - BCondValue, 44
 - BCondWrap, 46
- waitLessThanOrEqual
 - BCondInt, 41
 - BCondValue, 44
 - BCondWrap, 46
- waitMoreThanOrEqual
 - BCondInt, 41
 - BCondValue, 44
 - BCondWrap, 46
- wild
 - BDir.cpp, 246
- wildString
 - BDir.cpp, 246
- wprintf
 - BDebug.h, 244
- wrLock

- BRWLock, [140](#)
- write
 - BComms, [36](#)
 - BConfig, [48](#)
 - BEntryFile, [64](#)
 - BEventPipe, [77](#)
 - BFifo, [79](#), [80](#)
 - BFifoCirc, [83](#)
 - BFile, [88](#)
 - BFileData, [90](#)
 - BQueue, [134](#)
- writeAvailable
 - BComms, [36](#)
 - BEventPipe, [77](#)
 - BFifo, [80](#)
 - BFifoCirc, [83](#)
 - BQueue, [134](#)
- writeAvailableChunk
 - BFifo, [80](#)
- writeBackup
 - BFifo, [80](#)
- writeCsv
 - BFileCsv, [89](#)
- writeCycleParams
 - Tms::CycleParamDb, [187](#)
- writeData
 - BBuffer, [31](#)
 - BFifo, [80](#)
 - BFifoCirc, [83](#)
 - BUrl, [180](#)
- writeDone
 - BFifo, [80](#)
 - BFifoCirc, [83](#)
- writeList
 - BEntryFile, [64](#)
- writeString
 - BFile, [88](#)
- writeToFile
 - Tms::CycleParamEdit, [189](#)
- writeWaitAvailable
 - BFifoCirc, [83](#)
- yday
 - BDate, [50](#)
 - BTimeStamp, [175](#)
 - BTimeStampMs, [179](#)
- year
 - BDate, [50](#)
 - BTimeStamp, [175](#)
 - BTimeStampMs, [179](#)
- yearDays
 - BTime.cpp, [279](#)
- yearIsLeap
 - BTime.cpp, [279](#)