

**BdsApi**  
1.0.0

Generated by Doxygen 1.5.6

Wed Jun 3 13:59:30 2009



# Contents

<b>1</b>	<b>Main Page</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Overview . . . . .	1
1.3	Examples . . . . .	2
<b>2</b>	<b>Directory Hierarchy</b>	<b>7</b>
2.1	Directories . . . . .	7
<b>3</b>	<b>Namespace Index</b>	<b>9</b>
3.1	Namespace List . . . . .	9
<b>4</b>	<b>Class Index</b>	<b>11</b>
4.1	Class Hierarchy . . . . .	11
<b>5</b>	<b>Class Index</b>	<b>13</b>
5.1	Class List . . . . .	13
<b>6</b>	<b>File Index</b>	<b>15</b>
6.1	File List . . . . .	15
<b>7</b>	<b>Directory Documentation</b>	<b>17</b>
7.1	/src/blacknest/bds/bds/bdsDataLib/ Directory Reference . . . . .	17
<b>8</b>	<b>Namespace Documentation</b>	<b>19</b>
8.1	Bds Namespace Reference . . . . .	19
8.1.1	Enumeration Type Documentation . . . . .	22
8.1.1.1	BdsDataType . . . . .	22
8.1.1.2	Errors . . . . .	22
8.1.1.3	FileHeaderType . . . . .	23
8.1.1.4	FileSampleType . . . . .	23
8.1.1.5	Priority . . . . .	23

8.1.1.6	SampleFormat	24
8.1.1.7	SelectionGroup	24
8.1.2	Function Documentation	24
8.1.2.1	bdsChannelGetName	24
8.1.2.2	bdsChannelGetTypeAux	24
8.1.2.3	bdsDataInfoFromInfo	24
8.1.2.4	bdsDataInfoSetTimeRange	25
8.1.2.5	bdsDumpDataInfo	25
8.1.2.6	bdsInfoFromDataInfo	25
8.1.2.7	bdsUnCompressCm8	25
8.1.2.8	crc	25
8.1.2.9	dataCalculateDifference	25
8.1.2.10	dataChecksum	25
8.1.2.11	dataCompressCm6	25
8.1.2.12	dataConvert	25
8.1.3	Variable Documentation	25
8.1.3.1	apiVersion	25
8.1.3.2	BdsVersion	25
8.1.3.3	cm6Table	25
8.1.3.4	dataFormats	25
8.1.3.5	Scale	25
<b>9</b>	<b>Class Documentation</b>	<b>27</b>
9.1	Bds::AccessGroup Class Reference	27
9.1.1	Detailed Description	28
9.1.2	Constructor & Destructor Documentation	28
9.1.2.1	AccessGroup	28
9.1.3	Member Function Documentation	28
9.1.3.1	getType	28
9.1.3.2	setMembers	28
9.1.3.3	getMembers	28
9.1.4	Member Data Documentation	28
9.1.4.1	id	28
9.1.4.2	group	28
9.1.4.3	startTime	28
9.1.4.4	endTime	28
9.1.4.5	network	28

---

9.1.4.6	station	28
9.2	Bds::AdminAccess Class Reference	29
9.2.1	Detailed Description	32
9.2.2	Constructor & Destructor Documentation	32
9.2.2.1	AdminAccess	32
9.2.3	Member Function Documentation	32
9.2.3.1	connect	32
9.2.3.2	setUser	32
9.2.3.3	setUserReal	32
9.2.3.4	getVersion	32
9.2.3.5	userGetList	32
9.2.3.6	userUpdate	33
9.2.3.7	userDelete	33
9.2.3.8	userGetGroups	33
9.2.3.9	groupGetList	33
9.2.3.10	groupUpdate	33
9.2.3.11	groupDelete	33
9.2.3.12	accessGroupGetList	33
9.2.3.13	accessGroupUpdate	33
9.2.3.14	accessGroupDelete	33
9.2.3.15	getSelectionInfo	33
9.2.3.16	getSelections	33
9.2.3.17	networkGetList	34
9.2.3.18	networkUpdate	34
9.2.3.19	networkDelete	34
9.2.3.20	stationGetList	34
9.2.3.21	stationUpdate	34
9.2.3.22	stationDelete	34
9.2.3.23	locationGetList	34
9.2.3.24	locationUpdate	34
9.2.3.25	locationDelete	34
9.2.3.26	channelGetList	34
9.2.3.27	channelGet	34
9.2.3.28	channelUpdate	34
9.2.3.29	channelDelete	34
9.2.3.30	channelInstrumentGetList	34

---

9.2.3.31	channelInstrumentUpdate	35
9.2.3.32	channelInstrumentDelete	35
9.2.3.33	digitiserGetList	35
9.2.3.34	digitiserGet	35
9.2.3.35	digitiserUpdate	35
9.2.3.36	digitiserDelete	35
9.2.3.37	sensorGetList	35
9.2.3.38	sensorGet	35
9.2.3.39	sensorUpdate	35
9.2.3.40	sensorDelete	35
9.2.3.41	sensorLocationGetList	35
9.2.3.42	sensorLocationUpdate	35
9.2.3.43	sensorLocationDelete	35
9.2.3.44	calibrationGetList	35
9.2.3.45	calibrationUpdate	36
9.2.3.46	calibrationDelete	36
9.2.3.47	responseGetList	36
9.2.3.48	responseUpdate	36
9.2.3.49	responseDelete	36
9.2.3.50	dataSearch	36
9.2.3.51	dataGetChannelInfo	36
9.2.3.52	dataOpen	36
9.2.3.53	dataGetInfo	36
9.2.3.54	dataGetWarnings	36
9.2.3.55	dataGetBlock	36
9.2.3.56	dataSeekBlock	36
9.2.3.57	dataSetInfo	37
9.2.3.58	dataPutBlock	37
9.2.3.59	dataClose	37
9.2.3.60	dataFormattedRead	37
9.2.3.61	dataFormattedGetLength	37
9.2.3.62	changeGroupStart	38
9.2.3.63	changeGroupEnd	38
9.2.3.64	changeGroupGetList	38
9.2.3.65	changeGroupDelete	38
9.2.3.66	changeGetListNumber	38

---

9.2.3.67	changeGetList	38
9.2.3.68	changeDelete	38
9.2.3.69	noteGetList	38
9.2.3.70	noteUpdate	38
9.2.3.71	noteDelete	38
9.2.3.72	statisticsGet	38
9.2.3.73	sqlQuery	38
9.3	Bds::ArrayChannel Class Reference	39
9.3.1	Detailed Description	39
9.3.2	Constructor & Destructor Documentation	39
9.3.2.1	ArrayChannel	39
9.3.3	Member Data Documentation	39
9.3.3.1	station	39
9.3.3.2	channel	39
9.4	Bds::BdsDataBlock Struct Reference	40
9.4.1	Member Data Documentation	40
9.4.1.1	header	40
9.4.1.2	data	40
9.5	Bds::BdsDataBlockHeader Struct Reference	41
9.5.1	Member Data Documentation	41
9.5.1.1	type	41
9.5.1.2	length	41
9.5.1.3	packetOffset	41
9.6	Bds::BdsDataPacket Class Reference	42
9.6.1	Constructor & Destructor Documentation	42
9.6.1.1	BdsDataPacket	42
9.6.1.2	~BdsDataPacket	42
9.6.2	Member Function Documentation	42
9.6.2.1	clear	42
9.6.2.2	setChecksumAndLength	42
9.6.2.3	validateChecksum	42
9.6.2.4	setHeader	42
9.6.2.5	getHeader	42
9.6.2.6	dump	42
9.7	Bds::BdsDataPacketHeader Struct Reference	43
9.7.1	Member Data Documentation	43

9.7.1.1	type	43
9.7.1.2	length	43
9.7.1.3	streamlet	43
9.7.1.4	sequence	43
9.7.1.5	checksum	44
9.7.1.6	startTime	44
9.7.1.7	endTime	44
9.8	Bds::BdsDataStreamlet Class Reference	45
9.8.1	Constructor & Destructor Documentation	45
9.8.1.1	BdsDataStreamlet	45
9.8.2	Member Data Documentation	45
9.8.2.1	packetNumber	45
9.8.2.2	position	45
9.9	Bds::Calibration Class Reference	46
9.9.1	Detailed Description	47
9.9.2	Constructor & Destructor Documentation	47
9.9.2.1	Calibration	47
9.9.3	Member Function Documentation	47
9.9.3.1	getType	47
9.9.3.2	setMembers	47
9.9.3.3	getMembers	47
9.9.4	Member Data Documentation	47
9.9.4.1	id	47
9.9.4.2	startTime	47
9.9.4.3	endTime	47
9.9.4.4	network	47
9.9.4.5	station	47
9.9.4.6	channel	47
9.9.4.7	samplingFrequency	48
9.9.4.8	calibrationFrequency	48
9.9.4.9	calibrationFactor	48
9.9.4.10	calibrationUnits	48
9.10	Bds::Change Class Reference	49
9.10.1	Detailed Description	49
9.10.2	Constructor & Destructor Documentation	50
9.10.2.1	Change	50

---

9.10.3	Member Function Documentation	50
9.10.3.1	getType	50
9.10.3.2	setMembers	50
9.10.3.3	getMembers	50
9.10.4	Member Data Documentation	50
9.10.4.1	id	50
9.10.4.2	changeGroupId	50
9.10.4.3	time	50
9.10.4.4	type	50
9.10.4.5	table	50
9.10.4.6	rowId	50
9.11	Bds::ChangeGroup Class Reference	51
9.11.1	Detailed Description	51
9.11.2	Constructor & Destructor Documentation	52
9.11.2.1	ChangeGroup	52
9.11.3	Member Function Documentation	52
9.11.3.1	getType	52
9.11.3.2	setMembers	52
9.11.3.3	getMembers	52
9.11.4	Member Data Documentation	52
9.11.4.1	id	52
9.11.4.2	time	52
9.11.4.3	type	52
9.11.4.4	user	52
9.11.4.5	title	52
9.11.4.6	description	52
9.12	Bds::Channel Class Reference	53
9.12.1	Detailed Description	53
9.12.2	Constructor & Destructor Documentation	54
9.12.2.1	Channel	54
9.12.3	Member Function Documentation	54
9.12.3.1	getType	54
9.12.3.2	setMembers	54
9.12.3.3	getMembers	54
9.12.4	Member Data Documentation	54
9.12.4.1	id	54

9.12.4.2	startTime	54
9.12.4.3	endTime	54
9.12.4.4	network	54
9.12.4.5	station	54
9.12.4.6	channel	54
9.12.4.7	channelType	54
9.12.4.8	channelAux	55
9.12.4.9	description	55
9.13	Bds::ChannelInfo Class Reference	56
9.13.1	Constructor & Destructor Documentation	56
9.13.1.1	ChannelInfo	56
9.13.2	Member Data Documentation	56
9.13.2.1	station	56
9.13.2.2	location	57
9.13.2.3	channel	57
9.13.2.4	digitiser	57
9.13.2.5	sensor	57
9.13.2.6	sensorLocation	57
9.13.2.7	calibration	57
9.13.2.8	responses	57
9.14	Bds::ChannelInstrument Class Reference	58
9.14.1	Constructor & Destructor Documentation	59
9.14.1.1	ChannelInstrument	59
9.14.2	Member Function Documentation	59
9.14.2.1	getType	59
9.14.2.2	setMembers	59
9.14.2.3	getMembers	59
9.14.3	Member Data Documentation	59
9.14.3.1	id	59
9.14.3.2	startTime	59
9.14.3.3	endTime	59
9.14.3.4	channelId	59
9.14.3.5	digitiserId	59
9.14.3.6	sensorId	59
9.15	Bds::DataAccess Class Reference	60
9.15.1	Detailed Description	61

---

9.15.2	Constructor & Destructor Documentation	61
9.15.2.1	DataAccess	61
9.15.3	Member Function Documentation	61
9.15.3.1	connect	61
9.15.3.2	setUser	61
9.15.3.3	setUserReal	61
9.15.3.4	getVersion	62
9.15.3.5	userGetGroups	62
9.15.3.6	networkGetList	62
9.15.3.7	stationGetList	62
9.15.3.8	getSelectionInfo	62
9.15.3.9	getSelections	62
9.15.3.10	dataSearch	62
9.15.3.11	dataGetChannelInfo	62
9.15.3.12	dataOpen	62
9.15.3.13	dataGetInfo	62
9.15.3.14	dataGetWarnings	62
9.15.3.15	dataSeekBlock	63
9.15.3.16	dataGetBlock	63
9.15.3.17	dataClose	63
9.15.3.18	dataFormattedRead	63
9.15.3.19	dataFormattedGetLength	63
9.15.3.20	noteUpdate	63
9.16	Bds::DataAddAccess Class Reference	64
9.16.1	Detailed Description	65
9.16.2	Constructor & Destructor Documentation	65
9.16.2.1	DataAddAccess	65
9.16.3	Member Function Documentation	65
9.16.3.1	connect	65
9.16.3.2	setUser	65
9.16.3.3	setUserReal	65
9.16.3.4	getVersion	66
9.16.3.5	userGetGroups	66
9.16.3.6	networkGetList	66
9.16.3.7	stationGetList	66
9.16.3.8	getSelectionInfo	66

9.16.3.9	getSelections	66
9.16.3.10	dataSearch	66
9.16.3.11	dataGetChannelInfo	66
9.16.3.12	dataOpen	66
9.16.3.13	dataGetInfo	66
9.16.3.14	dataGetWarnings	67
9.16.3.15	dataSeekBlock	67
9.16.3.16	dataGetBlock	67
9.16.3.17	dataSetInfo	67
9.16.3.18	dataPutBlock	67
9.16.3.19	dataClose	67
9.16.3.20	dataFormattedRead	67
9.16.3.21	dataFormattedGetLength	67
9.16.3.22	noteUpdate	67
9.17	Bds::DataBlock Class Reference	68
9.17.1	Constructor & Destructor Documentation	68
9.17.1.1	DataBlock	68
9.17.2	Member Data Documentation	68
9.17.2.1	startTime	68
9.17.2.2	endTime	68
9.17.2.3	channelNumber	69
9.17.2.4	segmentNumber	69
9.17.2.5	channelData	69
9.17.2.6	info	69
9.18	Bds::DataChannel Class Reference	70
9.18.1	Detailed Description	71
9.18.2	Constructor & Destructor Documentation	71
9.18.2.1	DataChannel	71
9.18.3	Member Function Documentation	71
9.18.3.1	getType	71
9.18.3.2	setMembers	71
9.18.3.3	getMembers	71
9.18.4	Member Data Documentation	71
9.18.4.1	id	71
9.18.4.2	startTime	71
9.18.4.3	endTime	71

---

9.18.4.4	network	72
9.18.4.5	station	72
9.18.4.6	channel	72
9.18.4.7	source	72
9.18.4.8	numBlocks	72
9.18.4.9	numSamples	72
9.18.4.10	sampleRate	72
9.18.4.11	sampleFormat	72
9.18.4.12	dataFileId	72
9.18.4.13	dataFileChannel	72
9.18.4.14	info	72
9.19	Bds::DataCollate Class Reference	74
9.19.1	Detailed Description	74
9.19.2	Constructor & Destructor Documentation	74
9.19.2.1	DataCollate	74
9.19.2.2	~DataCollate	74
9.19.3	Member Function Documentation	74
9.19.3.1	addSource	74
9.19.3.2	readData	74
9.20	Bds::DataFile Class Reference	75
9.20.1	Detailed Description	77
9.20.2	Member Typedef Documentation	77
9.20.2.1	Options	77
9.20.3	Member Enumeration Documentation	77
9.20.3.1	DataOrder	77
9.20.3.2	Features	77
9.20.3.3	OptionsList	77
9.20.4	Constructor & Destructor Documentation	78
9.20.4.1	DataFile	78
9.20.4.2	~DataFile	78
9.20.5	Member Function Documentation	78
9.20.5.1	init	78
9.20.5.2	open	78
9.20.5.3	close	78
9.20.5.4	setFormat	78
9.20.5.5	getDataOrder	78

9.20.5.6	getFeatures	78
9.20.5.7	setInfo	78
9.20.5.8	start	79
9.20.5.9	writeData	79
9.20.5.10	end	79
9.20.5.11	fileNameProcess	79
9.20.5.12	getFormat	79
9.20.5.13	getInfo	79
9.20.5.14	seekBlock	79
9.20.5.15	readData	79
9.20.5.16	getFormats	79
9.20.6	Member Data Documentation	80
9.20.6.1	ofileName	80
9.20.6.2	omode	80
9.20.6.3	ofileNameTime	80
9.20.6.4	ofile	80
9.20.6.5	oformat	80
9.21	Bds::DataFileAscii Class Reference	81
9.21.1	Constructor & Destructor Documentation	81
9.21.1.1	DataFileAscii	81
9.21.2	Member Function Documentation	81
9.21.2.1	open	81
9.21.2.2	setInfo	82
9.21.2.3	writeData	82
9.21.2.4	getFormats	82
9.21.3	Member Data Documentation	82
9.21.3.1	odataInfo	82
9.22	Bds::DataFileBdrs Class Reference	83
9.22.1	Constructor & Destructor Documentation	84
9.22.1.1	DataFileBdrs	84
9.22.1.2	DataFileBdrs	84
9.22.2	Member Function Documentation	84
9.22.2.1	open	84
9.22.2.2	getFeatures	84
9.22.2.3	getDataOrder	84
9.22.2.4	getInfo	84

9.22.2.5	readData	85
9.22.2.6	getFormats	85
9.22.2.7	getFeatures	85
9.22.2.8	getDataOrder	85
9.22.2.9	getInfo	85
9.22.2.10	readData	85
9.22.2.11	getFormats	85
9.22.2.12	readBlock	86
9.22.3	Member Data Documentation	86
9.22.3.1	oyear	86
9.22.3.2	ofileNameDate	86
9.22.3.3	oblockSize	86
9.22.3.4	osampleRate	86
9.22.3.5	odataInfo	86
9.22.3.6	omagic	86
9.22.3.7	oblockYear	86
9.22.3.8	oblockPositions	86
9.23	Bds::DataFileBds Class Reference	87
9.23.1	Detailed Description	89
9.23.2	Member Enumeration Documentation	90
9.23.2.1	"@0	90
9.23.2.2	"@1	90
9.23.2.3	PackFormat	90
9.23.3	Constructor & Destructor Documentation	90
9.23.3.1	DataFileBds	90
9.23.3.2	~DataFileBds	90
9.23.4	Member Function Documentation	90
9.23.4.1	open	90
9.23.4.2	close	90
9.23.4.3	setFormat	90
9.23.4.4	setInfo	91
9.23.4.5	writeData	91
9.23.4.6	getDataOrder	91
9.23.4.7	getInfo	91
9.23.4.8	seekBlock	91
9.23.4.9	readData	91

9.23.4.10	getFormats	91
9.23.4.11	setDiskBlockSize	91
9.23.4.12	getDiskBlockSize	91
9.23.4.13	clear	92
9.23.4.14	setInfoRepeat	92
9.23.4.15	infoSet	92
9.23.4.16	infoGet	92
9.23.4.17	addInfoPacket	92
9.23.4.18	dataAppend	92
9.23.4.19	dataGet	92
9.23.4.20	packetRead	92
9.23.4.21	packetWrite	92
9.23.4.22	packetSeek	92
9.23.4.23	diskBlockReset	92
9.23.4.24	diskBlockWrite	92
9.23.4.25	diskBlockWriteFlush	93
9.23.4.26	diskBlockRead	93
9.23.4.27	diskBlockSeek	93
9.23.5	Member Data Documentation	93
9.23.5.1	oformat	93
9.23.5.2	opackFormat	93
9.23.5.3	odataInfo	93
9.23.5.4	ochannelInfos	93
9.23.5.5	oinfoRepeat	93
9.23.5.6	oinfo	93
9.23.5.7	oinfoHeader	93
9.23.5.8	odiskBlockSize	93
9.23.5.9	odiskBlock	94
9.23.5.10	odiskPosition	94
9.23.5.11	ostreamlets	94
9.23.5.12	opacket	94
9.24	Bds::DataFileBknas Class Reference	95
9.24.1	Constructor & Destructor Documentation	95
9.24.1.1	DataFileBknas	95
9.24.2	Member Function Documentation	95
9.24.2.1	open	95

---

9.24.2.2	setInfo	96
9.24.2.3	writeData	96
9.24.2.4	getFormats	96
9.24.3	Member Data Documentation	96
9.24.3.1	odataInfo	96
9.25	Bds::DataFileGcf Class Reference	97
9.25.1	Constructor & Destructor Documentation	97
9.25.1.1	DataFileGcf	97
9.25.2	Member Function Documentation	97
9.25.2.1	getFeatures	97
9.25.2.2	getInfo	98
9.25.2.3	readData	98
9.25.2.4	getFormats	98
9.25.3	Member Data Documentation	98
9.25.3.1	oblockSize	98
9.25.3.2	odataInfo	98
9.25.3.3	osystemId	98
9.25.3.4	ostreamId	98
9.25.3.5	oformat	98
9.25.3.6	ofilterCoef	98
9.25.3.7	osampleRate	98
9.26	Bds::DataFileIms Class Reference	99
9.26.1	Constructor & Destructor Documentation	100
9.26.1.1	DataFileIms	100
9.26.2	Member Function Documentation	100
9.26.2.1	open	100
9.26.2.2	close	100
9.26.2.3	getDataOrder	100
9.26.2.4	getFeatures	100
9.26.2.5	setInfo	100
9.26.2.6	start	100
9.26.2.7	writeData	101
9.26.2.8	end	101
9.26.2.9	getFormats	101
9.26.3	Member Data Documentation	101
9.26.3.1	odataInfo	101

9.26.3.2	ochannelInfos	101
9.26.3.3	owriteChannel	101
9.26.3.4	owriteChecksum	101
9.26.3.5	owriteColumn	101
9.26.3.6	oprevValue1	101
9.26.3.7	oprevValue2	101
9.27	Bds::DataFileTapeDigitiser Class Reference	102
9.27.1	Detailed Description	103
9.27.2	Constructor & Destructor Documentation	103
9.27.2.1	DataFileTapeDigitiser	103
9.27.3	Member Function Documentation	103
9.27.3.1	open	103
9.27.3.2	getInfo	103
9.27.3.3	readData	103
9.27.3.4	getFormats	103
9.27.3.5	blockSize	103
9.27.3.6	blockNumSamples	103
9.27.3.7	readHeader	104
9.27.3.8	computeChecksum	104
9.27.4	Member Data Documentation	104
9.27.4.1	oheader	104
9.27.4.2	oheaderSize	104
9.27.4.3	oblockSize	104
9.27.4.4	oblockNumSamples	104
9.27.4.5	osampleRate	104
9.27.4.6	obuffer	104
9.27.4.7	oversion	104
9.28	Bds::DataFileWra Class Reference	105
9.28.1	Constructor & Destructor Documentation	106
9.28.1.1	DataFileWra	106
9.28.1.2	DataFileWra	106
9.28.2	Member Function Documentation	106
9.28.2.1	open	106
9.28.2.2	setFormat	106
9.28.2.3	getFeatures	106
9.28.2.4	getDataOrder	107

---

9.28.2.5	getInfo	107
9.28.2.6	readData	107
9.28.2.7	getFormats	107
9.28.2.8	readBlock	107
9.28.2.9	setFormat	107
9.28.2.10	getFeatures	107
9.28.2.11	getDataOrder	107
9.28.2.12	getInfo	107
9.28.2.13	readData	107
9.28.2.14	getFormats	108
9.28.2.15	readBlock	108
9.28.3	Member Data Documentation	108
9.28.3.1	oyear	108
9.28.3.2	oblockYear	108
9.28.3.3	ofileNameDate	108
9.28.3.4	oblockSize	108
9.28.3.5	onumChannels	108
9.28.3.6	osampleRate	108
9.28.3.7	odataInfo	108
9.28.3.8	oblockPositions	108
9.28.3.9	omagic	108
9.29	Bds::DataFileWraAgso Class Reference	109
9.29.1	Constructor & Destructor Documentation	110
9.29.1.1	DataFileWraAgso	110
9.29.2	Member Function Documentation	110
9.29.2.1	getFeatures	110
9.29.2.2	getDataOrder	110
9.29.2.3	getInfo	110
9.29.2.4	readData	110
9.29.2.5	getFormats	110
9.29.2.6	readBlock	111
9.29.3	Member Data Documentation	111
9.29.3.1	oyear	111
9.29.3.2	oblockYear	111
9.29.3.3	oblockSize	111
9.29.3.4	osampleRate	111

9.29.3.5	odataInfo	111
9.29.3.6	ochannelStartBlocks	111
9.29.3.7	oreadChannel	111
9.29.3.8	oreadBlock	111
9.29.3.9	ocurrentBlock	111
9.29.3.10	ofieldWidths	111
9.30	Bds::DataFormats Class Reference	112
9.30.1	Detailed Description	112
9.30.2	Constructor & Destructor Documentation	112
9.30.2.1	DataFormats	112
9.30.2.2	~DataFormats	112
9.30.3	Member Function Documentation	112
9.30.3.1	formatList	112
9.30.3.2	formatGet	112
9.30.3.3	findString	112
9.31	Bds::DataHandle Class Reference	113
9.31.1	Detailed Description	113
9.31.2	Constructor & Destructor Documentation	113
9.31.2.1	DataHandle	113
9.31.3	Member Data Documentation	113
9.31.3.1	handle	113
9.31.3.2	startBlock	113
9.32	Bds::DataInfo Class Reference	114
9.32.1	Constructor & Destructor Documentation	114
9.32.1.1	DataInfo	114
9.32.2	Member Data Documentation	114
9.32.2.1	startTime	114
9.32.2.2	endTime	115
9.32.2.3	array	115
9.32.2.4	description	115
9.32.2.5	synchronous	115
9.32.2.6	channels	115
9.32.2.7	info	115
9.33	Bds::Digitiser Class Reference	116
9.33.1	Detailed Description	117
9.33.2	Constructor & Destructor Documentation	117

---

9.33.2.1	Digitiser	117
9.33.3	Member Function Documentation	117
9.33.3.1	getType	117
9.33.3.2	setMembers	117
9.33.3.3	getMembers	117
9.33.4	Member Data Documentation	117
9.33.4.1	id	117
9.33.4.2	startTime	117
9.33.4.3	endTime	117
9.33.4.4	name	117
9.33.4.5	type	117
9.33.4.6	serialNumber	117
9.33.4.7	numberChannels	118
9.33.4.8	baseSamplingFrequency	118
9.33.4.9	initialSamplingFrequency	118
9.33.4.10	gain	118
9.34	Bds::Fap Class Reference	119
9.34.1	Detailed Description	119
9.34.2	Constructor & Destructor Documentation	119
9.34.2.1	Fap	119
9.34.3	Member Data Documentation	119
9.34.3.1	frequency	119
9.34.3.2	amplitude	119
9.34.3.3	phase	119
9.35	Bds::FileDataHeaderTape_1v1 Struct Reference	121
9.35.1	Member Data Documentation	121
9.35.1.1	type	121
9.35.1.2	headerLength	122
9.35.1.3	dataLength	122
9.35.1.4	dataType	122
9.35.1.5	numChannels	122
9.35.1.6	numSamples	122
9.35.1.7	startTime	122
9.35.1.8	endTime	122
9.35.1.9	tapeLastVelaCode	122
9.35.1.10	tapeSignalLevelErrors	122

---

9.35.1.11	tapeSpeedVariationErrors	122
9.35.1.12	tapeVelaBitErrors	122
9.35.1.13	tapeFmSignalLevels	123
9.36	Bds::FileDataHeaderTape_2v0 Struct Reference	124
9.36.1	Member Data Documentation	125
9.36.1.1	type	125
9.36.1.2	headerLength	125
9.36.1.3	dataLength	125
9.36.1.4	dataType	125
9.36.1.5	numChannels	125
9.36.1.6	numSamples	125
9.36.1.7	startTime	125
9.36.1.8	endTime	125
9.36.1.9	tapeLastVelaCode	125
9.36.1.10	tapeSignalLevelErrors	125
9.36.1.11	tapeSpeedVariationErrors	125
9.36.1.12	tapeVelaBitErrors	126
9.36.1.13	tapeFmSignalLevels	126
9.36.1.14	headerChecksum	126
9.36.1.15	dataChecksum	126
9.37	Bds::Group Class Reference	127
9.37.1	Detailed Description	127
9.37.2	Constructor & Destructor Documentation	127
9.37.2.1	Group	127
9.37.3	Member Function Documentation	127
9.37.3.1	getType	127
9.37.3.2	setMembers	127
9.37.3.3	getMembers	127
9.37.4	Member Data Documentation	127
9.37.4.1	id	127
9.37.4.2	group	128
9.37.4.3	description	128
9.38	Bds::ListRange Class Reference	129
9.38.1	Detailed Description	129
9.38.2	Constructor & Destructor Documentation	129
9.38.2.1	ListRange	129

---

9.38.3	Member Function Documentation	129
9.38.3.1	getType	129
9.38.3.2	setMembers	129
9.38.3.3	getMembers	129
9.38.4	Member Data Documentation	129
9.38.4.1	start	129
9.38.4.2	number	130
9.38.4.3	reverse	130
9.39	Bds::Location Class Reference	131
9.39.1	Detailed Description	132
9.39.2	Constructor & Destructor Documentation	132
9.39.2.1	Location	132
9.39.3	Member Function Documentation	132
9.39.3.1	getType	132
9.39.3.2	setMembers	132
9.39.3.3	getMembers	132
9.39.4	Member Data Documentation	132
9.39.4.1	id	132
9.39.4.2	startTime	132
9.39.4.3	endTime	132
9.39.4.4	network	132
9.39.4.5	station	132
9.39.4.6	datum	132
9.39.4.7	longitude	133
9.39.4.8	latitude	133
9.39.4.9	elevation	133
9.39.4.10	arrayOffsetEast	133
9.39.4.11	arrayOffsetNorth	133
9.40	Bds::Network Class Reference	134
9.40.1	Detailed Description	134
9.40.2	Constructor & Destructor Documentation	135
9.40.2.1	Network	135
9.40.3	Member Function Documentation	135
9.40.3.1	getType	135
9.40.3.2	setMembers	135
9.40.3.3	getMembers	135

---

9.40.4	Member Data Documentation	135
9.40.4.1	id	135
9.40.4.2	network	135
9.40.4.3	description	135
9.40.4.4	stations	135
9.41	Bds::Note Class Reference	136
9.41.1	Detailed Description	136
9.41.2	Constructor & Destructor Documentation	137
9.41.2.1	Note	137
9.41.3	Member Function Documentation	137
9.41.3.1	getType	137
9.41.3.2	setMembers	137
9.41.3.3	getMembers	137
9.41.4	Member Data Documentation	137
9.41.4.1	id	137
9.41.4.2	time	137
9.41.4.3	type	137
9.41.4.4	user	137
9.41.4.5	title	137
9.41.4.6	description	137
9.42	Bds::Point Class Reference	138
9.42.1	Constructor & Destructor Documentation	138
9.42.1.1	Point	138
9.42.2	Member Data Documentation	138
9.42.2.1	x	138
9.42.2.2	y	138
9.43	Bds::PoleZero Class Reference	139
9.43.1	Detailed Description	139
9.43.2	Constructor & Destructor Documentation	139
9.43.2.1	PoleZero	139
9.43.3	Member Data Documentation	139
9.43.3.1	poles	139
9.43.3.2	zeros	139
9.44	Bds::Response Class Reference	140
9.44.1	Detailed Description	141
9.44.2	Constructor & Destructor Documentation	141

---

9.44.2.1	Response	141
9.44.3	Member Data Documentation	141
9.44.3.1	id	141
9.44.3.2	startTime	141
9.44.3.3	endTime	141
9.44.3.4	network	142
9.44.3.5	station	142
9.44.3.6	channel	142
9.44.3.7	name	142
9.44.3.8	type	142
9.44.3.9	poleZeros	142
9.44.3.10	faps	142
9.44.3.11	firCoefficients	142
9.44.3.12	gain	142
9.44.3.13	gainFrequency	142
9.44.3.14	decimation	142
9.44.3.15	symmetry	143
9.45	Bds::ResponseEdit Class Reference	144
9.45.1	Detailed Description	144
9.45.2	Constructor & Destructor Documentation	144
9.45.2.1	ResponseEdit	144
9.45.2.2	~ResponseEdit	144
9.45.3	Member Function Documentation	144
9.45.3.1	getString	144
9.45.3.2	setString	144
9.46	Bds::Selection Class Reference	145
9.46.1	Constructor & Destructor Documentation	146
9.46.1.1	Selection	146
9.46.2	Member Data Documentation	146
9.46.2.1	range	146
9.46.2.2	startTime	146
9.46.2.3	endTime	146
9.46.2.4	channels	146
9.46.2.5	channelId	146
9.46.2.6	digitiserId	146
9.46.2.7	sensorId	146

---

9.46.2.8	sensorOldId	146
9.47	Bds::SelectionChannel Class Reference	147
9.47.1	Constructor & Destructor Documentation	147
9.47.1.1	SelectionChannel	147
9.47.2	Member Data Documentation	147
9.47.2.1	network	147
9.47.2.2	station	147
9.47.2.3	channel	147
9.47.2.4	source	147
9.48	Bds::SelectionInfo Class Reference	148
9.48.1	Detailed Description	148
9.48.2	Constructor & Destructor Documentation	149
9.48.2.1	SelectionInfo	149
9.48.3	Member Data Documentation	149
9.48.3.1	startTime	149
9.48.3.2	endTime	149
9.48.3.3	networks	149
9.48.3.4	stations	149
9.48.3.5	channels	149
9.48.3.6	sources	149
9.48.3.7	numDataChannels	149
9.49	Bds::Sensor Class Reference	150
9.49.1	Detailed Description	151
9.49.2	Constructor & Destructor Documentation	151
9.49.2.1	Sensor	151
9.49.3	Member Function Documentation	151
9.49.3.1	getType	151
9.49.3.2	setMembers	151
9.49.3.3	getMembers	151
9.49.4	Member Data Documentation	151
9.49.4.1	id	151
9.49.4.2	startTime	151
9.49.4.3	endTime	151
9.49.4.4	name	151
9.49.4.5	type	151
9.49.4.6	serialNumber	151

---

9.49.4.7	numberChannels	152
9.49.4.8	gainUnits	152
9.49.4.9	gain	152
9.49.4.10	oldId	152
9.50	Bds::SensorLocation Class Reference	153
9.50.1	Detailed Description	153
9.50.2	Constructor & Destructor Documentation	154
9.50.2.1	SensorLocation	154
9.50.3	Member Function Documentation	154
9.50.3.1	getType	154
9.50.3.2	setMembers	154
9.50.3.3	getMembers	154
9.50.4	Member Data Documentation	154
9.50.4.1	id	154
9.50.4.2	startTime	154
9.50.4.3	endTime	154
9.50.4.4	sensorId	154
9.50.4.5	depth	154
9.50.4.6	horizontalAngle	154
9.50.4.7	verticalAngle	154
9.51	Bds::Station Class Reference	155
9.51.1	Constructor & Destructor Documentation	155
9.51.1.1	Station	155
9.51.2	Member Data Documentation	155
9.51.2.1	id	155
9.51.2.2	name	155
9.51.2.3	type	155
9.51.2.4	description	156
9.51.2.5	channels	156
9.52	Bds::TimePeriod Class Reference	157
9.52.1	Constructor & Destructor Documentation	157
9.52.1.1	TimePeriod	157
9.52.2	Member Function Documentation	157
9.52.2.1	getType	157
9.52.2.2	setMembers	157
9.52.2.3	getMembers	157

9.52.3	Member Data Documentation	157
9.52.3.1	startTime	157
9.52.3.2	endTime	157
9.53	Bds::User Class Reference	158
9.53.1	Detailed Description	158
9.53.2	Constructor & Destructor Documentation	159
9.53.2.1	User	159
9.53.3	Member Function Documentation	159
9.53.3.1	getType	159
9.53.3.2	setMembers	159
9.53.3.3	getMembers	159
9.53.4	Member Data Documentation	159
9.53.4.1	id	159
9.53.4.2	user	159
9.53.4.3	password	159
9.53.4.4	name	159
9.53.4.5	email	159
9.53.4.6	telephone	159
9.53.4.7	address	159
9.53.4.8	enabled	159
9.53.4.9	groups	160
<b>10</b>	<b>File Documentation</b>	<b>161</b>
10.1	/src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp File Reference	161
10.2	/src/blacknest/bds/bds/bdsDataLib/BdsCompress.h File Reference	162
10.3	/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp File Reference	163
10.4	/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h File Reference	164
10.5	/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp File Reference	165
10.6	/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h File Reference	166
10.7	/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp File Reference	167
10.8	/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h File Reference	168
10.9	/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs-1.cpp File Reference	169
10.10	/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs-1.h File Reference	170
10.11	/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp File Reference	171
10.12	/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h File Reference	172
10.13	/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp File Reference	173
10.13.1	Define Documentation	173

10.13.1.1 <code>dprintf</code> . . . . .	173
10.13.1.2 <code>LDEBUG</code> . . . . .	173
10.14/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h File Reference . . . . .	174
10.14.1 Define Documentation . . . . .	174
10.14.1.1 <code>BdsDataFileBds_H</code> . . . . .	174
10.15/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp File Reference . . . . .	175
10.16/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h File Reference . . . . .	176
10.17/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp File Reference . . . . .	177
10.17.1 Define Documentation . . . . .	177
10.17.1.1 <code>DEBUG</code> . . . . .	177
10.18/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h File Reference . . . . .	178
10.19/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp File Reference . . . . .	179
10.20/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h File Reference . . . . .	180
10.21/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp File Reference . . . . .	181
10.22/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h File Reference . . . . .	182
10.22.1 Define Documentation . . . . .	182
10.22.1.1 <code>BdsDataFileTapeDigitiser_H</code> . . . . .	182
10.23/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra-1.cpp File Reference . . . . .	183
10.24/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra-1.h File Reference . . . . .	184
10.25/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp File Reference . . . . .	185
10.26/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h File Reference . . . . .	186
10.27/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp File Reference . . . . .	187
10.27.1 Function Documentation . . . . .	187
10.27.1.1 <code>parseStringFixedFields</code> . . . . .	187
10.28/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h File Reference . . . . .	188
10.29/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp File Reference . . . . .	189
10.30/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h File Reference . . . . .	190
10.31/src/blacknest/bds/bds/bdsDataLib/canada_compress.c File Reference . . . . .	191
10.31.1 Define Documentation . . . . .	191
10.31.1.1 <code>htonl</code> . . . . .	191
10.31.1.2 <code>ntohl</code> . . . . .	191
10.31.2 Function Documentation . . . . .	191
10.31.2.1 <code>canada_compress</code> . . . . .	191
10.31.2.2 <code>canada_uncompress</code> . . . . .	191
10.31.2.3 <code>pack</code> . . . . .	191
10.31.2.4 <code>unpack</code> . . . . .	191

---

10.31.3 Variable Documentation . . . . .	191
10.31.3.1 corrupt . . . . .	191
10.32/src/blacknest/bds/bds/bdsDataLib/canada_compress.h File Reference . . . . .	192
10.32.1 Define Documentation . . . . .	192
10.32.1.1 CANCOMP_CORRUPT . . . . .	192
10.32.1.2 CANCOMP_ERR . . . . .	192
10.32.1.3 CANCOMP_EXCEED . . . . .	192
10.32.1.4 CANCOMP_NOT_20 . . . . .	192
10.32.1.5 CANCOMP_SUCCESS . . . . .	192
10.32.2 Function Documentation . . . . .	192
10.32.2.1 canada_compress . . . . .	192
10.32.2.2 canada_uncompress . . . . .	192
10.33BdsC.cc File Reference . . . . .	193
10.34BdsC.h File Reference . . . . .	194
10.34.1 Define Documentation . . . . .	194
10.34.1.1 BDSC_H . . . . .	194
10.35BdsD.cc File Reference . . . . .	195
10.36BdsD.h File Reference . . . . .	196
10.36.1 Define Documentation . . . . .	198
10.36.1.1 BSDSD_H . . . . .	198
10.37BdsLib.cpp File Reference . . . . .	199
10.38BdsLib.h File Reference . . . . .	200
10.39BdsS.cc File Reference . . . . .	201
10.40BdsT.cc File Reference . . . . .	202
10.41overview.dox File Reference . . . . .	203

# Chapter 1

## Main Page

**Author:**

Dr Terry Barnaby

**Version:**

1.0.0

**Date:**

2009-02-18

### 1.1 Introduction

This document covers the BEAM BdsApi software API for the Blacknest Data System. This API provides the ability to access data and administer the BDS 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 BdsApi API makes use of the BEAM 'C++' class library. The BEAM 'C++' class library provides a small set of low level 'C++' classes for strings, lists and system interface functions. It also implements the BOAP RPC mechanism used to implement the BdsApi. There is some brief information on the BEAM class library later on in this page and a full API definition is available in the libBeam documentation.

### 1.2 Overview

Generally users of the system are only concerned with the DataAccess API.

The BdsApi has been developed using the BOAP (BEAM Object Access Protocol). This provides a simple but powerful Object Orientated RPC mechanism. The BdsApi 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 BDS Controller. More information on the BOAP system can be found in the libBeam documentation.

THE BDS API implements a number of data storage classes and three interface objects. The interface objects are:

1. **Bds::DataAccess** BDS Data API: This will provide read only access to the data and meta data. It will be used by the AutoDRM email and Web systems as well as for program access to the data.
2. **Bds::DataAddAccess** BDS DataAdd API: This will provide read and restricted write access to enable the adding of data to the system. It will not allow deletions of data to be performed. It is designed to be used by manual and automatic data adding programs.
3. **Bds::AdminAccess** BDS Admin API: This will provide full read/write access to the data and meta data as well as administrative configuration information.

## 1.3 Examples

There are some examples of client applications using the BdsApi in the **bdsExamples** directory of the source code. Some simple Data Access client examples are listed below:

```

/*****
 *      BdsDataClient1.cpp      BDS API example code for a Data Client
 *                               T.Barnaby,      BEAM Ltd,      2008-09-02
 *****/
 *
 *      This is a very basic example of using the BdsApi from a data access
 *      perspective. It is designed to give an overview of using the API.
 *      This program gets data in the BKNAS format.
 */
#include <iostream>
#include <stdio.h>
#include <BdsD.h>
#include <BdsC.h>

using namespace Bds;
using namespace std;

// Function to read some data
BError bdsTest(DataAccess& bds){
    BError          err;
    Selection       selection;
    DataInfo        dataInfo;
    DataHandle      dataHandle;
    BArray<BUInt8> data;

    // Set up selection
#ifdef ZAP
    selection.startTime.setString("2002-01-01T00:00:00.000000");
    selection.endTime.setString("2002-01-01T00:01:00.000000");
#else
    selection.startTime.setString("2002-01-01T23:59:00.000000");
    selection.endTime.setString("2002-01-02T00:01:00.000000");
#endif

    selection.channels.append(SelectionChannel("BN", "EKA", "", ""));

    // Get list of all data available for the selection
    if(err = bds.dataSearch(selection, dataInfo){
        return err.set(1, BString("Error: Searching for data: ") + err.getString());
    }

    // We should now choose which set of data we would like from the list, here we just
    // choose the first entry and get the data in appropriate format.
    if(!dataInfo.channels.size())
        return err.set(1, "No data found");

    if(err = bds.dataOpen(dataInfo, "r", "IMS", dataHandle)){
        return err;
    }

```

```

    }

    while(1){
        if(err = bds.dataFormattedRead(dataHandle, 1024, data)){
            return err;
        }

        if(data.size() == 0)
            break;

        fwrite(data.data(), 1, data.size(), stdout);
    }

    return err;
}

int main(int argc, char** argv){
    BError          err;
    BString         hostName;
    DataAccess      bds;

    hostName = getenv("BDS_HOST");
    if(hostName == "")
        hostName = "localhost";

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

    // Connect to the DataAccess service
    if(err = bds.connectService(BString("/") + hostName + "/bdsDataAccess")){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }

    // Connect to service
    if(!err)
        err = bds.connect("test", "beam00");

    // Run a normal data gathering as a normal data access client would.
    if(!err)
        err = bdsTest(bds);

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

    return 0;
}

/*****
 *   BdsDataClient2.cpp      BDS API example code for a Data Client
 *   T.Barnaby,      BEAM Ltd,      2008-09-02
 *****/
 *
 *   This is a very basic example of using the BdsApi from a data access
 *   perspective. It is designed to give an overview of using the API.
 *   This program gets data in raw format and outputs it in ASCII.
 */
#include <iostream>
#include <iomanip>
#include <stdio.h>
#include <BdsD.h>
#include <BdsC.h>

using namespace Bds;
using namespace std;

```

```

// Function to read some data
BError bdsTest(DataAccess& bds){
    BError          err;
    Selection       selection;
    DataInfo        dataInfo;
    DataHandle      dataHandle;
    DataBlock       data;
    BUInt32         blockNumber = 0;
    BUInt           c;
    BUInt           s;

    // Set up selection
    selection.startTime.setString("2008-01-03T00:00:00.000000");
    selection.endTime.setString("2008-01-03T00:01:00.000000");
    selection.channels.append(SelectionChannel("BN", "EKA", "", ""));

    // Get list of all data available for the selection
    if(err = bds.dataSearch(selection, dataInfo)){
        return err.set(1, BString("Error: Searching for data: ") + err.getString());
    }

    // We should now choose which set of data we would like from the list, here we just
    // choose the first entry and get the data in appropriate format.
    if(!dataInfo.channels.size())
        return err.set(1, "No data found");

    if(err = bds.dataOpen(dataInfo, "r", "API", dataHandle)){
        return err;
    }

    while(1){
        if(err = bds.dataGetBlock(dataHandle, 0, 0, blockNumber, data)){
            return err;
        }

        if(data.startTime >= dataInfo.endTime)
            break;

        for(s = 0; s < data.channelData[0].size(); s++){
            for(c = 0; c < data.channelData.size(); c++){
                if(c != 0)
                    std::cout << ", ";
                std::cout << setw(8) << data.channelData[c][s];
            }
            std::cout << "\n";
        }
        blockNumber++;
    }

    return err;
}

int main(int argc, char** argv){
    BError          err;
    BString         hostName;
    DataAccess      bds;

    hostName = getenv("BDS_HOST");
    if(hostName == "")
        hostName = "localhost";

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

    // Connect to the DataAccess service

```

```

    if(err = bds.connectService(BString("/") + hostName + "/bdsDataAccess")){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }

    // Connect to service
    if(!err)
        err = bds.connect("test", "beam00");

    // Run a normal data gathering as a normal data access client would.
    if(!err)
        err = bdsTest(bds);

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

    return 0;
}

/*****
 *   BdsDataClient3.cpp       BDS API example code for a Data Client
 *                           T.Barnaby,      BEAM Ltd,      2008-09-02
 *****/
*
*   This is a very basic example of using the BdsApi from a data access
*   perspective. It is designed to give an overview of using the API.
*   This program gets information on the data channels.
*/
#include <iostream>
#include <iomanip>
#include <stdio.h>
#include <BdsD.h>
#include <BdsC.h>

using namespace Bds;
using namespace std;

// Function to read some data
BError bdsTest(DataAccess& bds){
    BError      err;
    Selection   selection;
    DataInfo    dataInfo;
    BList<ChannelInfo> channelInfos;
    BIter       i;
    BUInt       n;

    // Set up selection
    selection.startTime.setString("2008-01-03T00:00:00.000000");
    selection.endTime.setString("2008-01-03T00:01:00.000000");
    selection.channels.append(SelectionChannel("BN", "EKA", "", ""));

    // Get list of all data available for the selection
    if(err = bds.dataSearch(selection, dataInfo)){
        return err.set(1, BString("Error: Searching for data: ") + err.getString());
    }

    // We should now choose which set of data we would like from the list, here we just
    // choose the first entry and get the data in appropriate format.
    if(!dataInfo.channels.size())
        return err.set(1, "No data found");

    if(err = bds.dataGetChannelInfo(dataInfo, channelInfos)){
        return err;
    }
}

```

```
// This displays some of the information available
for(channelInfos.start(i), n = 0; !channelInfos.isEnd(i); channelInfos.next(i), n++){
    ChannelInfo& c = channelInfos[i];

    cout << n << ": Station: " << c.station.name << " Channel: " << c.channel.channel << "\n";
    cout << "      " << "Calibration: " << c.calibration.calibrationFactor << " Frequency: " <<
        " Units: " << c.calibration.calibrationUnits << "\n";
    cout << "      " << "Instrument: " << c.digitiser.name << "\n";
}

return err;
}

int main(int argc, char** argv){
    BError err;
    BString hostName;
    DataAccess bds;

    hostName = getenv("BDS_HOST");
    if(hostName == "")
        hostName = "localhost";

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

    // Connect to the DataAccess service
    if(err = bds.connectService(BString("//") + hostName + "/bdsDataAccess")){
        cerr << "Error: " << err.getString() << "\n";
        return 1;
    }

    // Connect to service
    if(!err)
        err = bds.connect("test", "beam00");

    // Run a normal data gathering as a normal data access client would.
    if(!err)
        err = bdsTest(bds);

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

    return 0;
}
```

## Chapter 2

# Directory Hierarchy

### 2.1 Directories

This directory hierarchy is sorted roughly, but not completely, alphabetically:

bdsDataLib . . . . .	17
----------------------	----



# Chapter 3

## Namespace Index

### 3.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">Bds</a> .....	19
---------------------------	----



# Chapter 4

## Class Index

### 4.1 Class Hierarchy

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

Bds::AccessGroup . . . . .	27
Bds::AdminAccess . . . . .	29
Bds::ArrayChannel . . . . .	39
Bds::BdsDataBlock . . . . .	40
Bds::BdsDataBlockHeader . . . . .	41
Bds::BdsDataPacket . . . . .	42
Bds::BdsDataPacketHeader . . . . .	43
Bds::BdsDataStreamlet . . . . .	45
Bds::Calibration . . . . .	46
Bds::Change . . . . .	49
Bds::ChangeGroup . . . . .	51
Bds::Channel . . . . .	53
Bds::ChannelInfo . . . . .	56
Bds::ChannelInstrument . . . . .	58
Bds::DataAccess . . . . .	60
Bds::DataAddAccess . . . . .	64
Bds::DataBlock . . . . .	68
Bds::DataChannel . . . . .	70
Bds::DataCollate . . . . .	74
Bds::DataFile . . . . .	75
Bds::DataFileAscii . . . . .	81
Bds::DataFileBdrs . . . . .	83
Bds::DataFileBdrs . . . . .	83
Bds::DataFileBds . . . . .	87
Bds::DataFileBknas . . . . .	95
Bds::DataFileGcf . . . . .	97
Bds::DataFileIms . . . . .	99
Bds::DataFileTapeDigitiser . . . . .	102
Bds::DataFileWra . . . . .	105
Bds::DataFileWra . . . . .	105
Bds::DataFileWraAgso . . . . .	109
Bds::DataFormats . . . . .	112
Bds::DataHandle . . . . .	113

---

Bds::DataInfo . . . . .	114
Bds::Digitiser . . . . .	116
Bds::Fap . . . . .	119
Bds::FileDataHeaderTape_1v1 . . . . .	121
Bds::FileDataHeaderTape_2v0 . . . . .	124
Bds::Group . . . . .	127
Bds::ListRange . . . . .	129
Bds::Location . . . . .	131
Bds::Network . . . . .	134
Bds::Note . . . . .	136
Bds::Point . . . . .	138
Bds::PoleZero . . . . .	139
Bds::Response . . . . .	140
Bds::ResponseEdit . . . . .	144
Bds::Selection . . . . .	145
Bds::SelectionChannel . . . . .	147
Bds::SelectionInfo . . . . .	148
Bds::Sensor . . . . .	150
Bds::SensorLocation . . . . .	153
Bds::Station . . . . .	155
Bds::TimePeriod . . . . .	157
Bds::User . . . . .	158

# Chapter 5

## Class Index

### 5.1 Class List

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

<a href="#">Bds::AccessGroup</a> (This holds information on data access groups ) . . . . .	27
<a href="#">Bds::AdminAccess</a> (This is the <a href="#">AdminAccess</a> API interface ) . . . . .	29
<a href="#">Bds::ArrayChannel</a> (This class defines an arrays channel ) . . . . .	39
<a href="#">Bds::BdsDataBlock</a> . . . . .	40
<a href="#">Bds::BdsDataBlockHeader</a> . . . . .	41
<a href="#">Bds::BdsDataPacket</a> . . . . .	42
<a href="#">Bds::BdsDataPacketHeader</a> . . . . .	43
<a href="#">Bds::BdsDataStreamlet</a> . . . . .	45
<a href="#">Bds::Calibration</a> (This class defines a calibration setting ) . . . . .	46
<a href="#">Bds::Change</a> (This holds information on a change ) . . . . .	49
<a href="#">Bds::ChangeGroup</a> (This holds information on a change ) . . . . .	51
<a href="#">Bds::Channel</a> (This class defines a <a href="#">Channel</a> ) . . . . .	53
<a href="#">Bds::ChannelInfo</a> . . . . .	56
<a href="#">Bds::ChannelInstrument</a> . . . . .	58
<a href="#">Bds::DataAccess</a> . . . . .	60
<a href="#">Bds::DataAddAccess</a> (This is the DataAdd Access API interface ) . . . . .	64
<a href="#">Bds::DataBlock</a> . . . . .	68
<a href="#">Bds::DataChannel</a> (This class defines information on a single channels set of data ) . . . . .	70
<a href="#">Bds::DataCollate</a> (This class defines the interface for generic data file access ) . . . . .	74
<a href="#">Bds::DataFile</a> (This class defines the interface for generic data file access ) . . . . .	75
<a href="#">Bds::DataFileAscii</a> . . . . .	81
<a href="#">Bds::DataFileBdrs</a> . . . . .	83
<a href="#">Bds::DataFileBds</a> (This class implements the BDS Data File/Stream access system ) . . . . .	87
<a href="#">Bds::DataFileBknas</a> . . . . .	95
<a href="#">Bds::DataFileGcf</a> . . . . .	97
<a href="#">Bds::DataFileImS</a> . . . . .	99
<a href="#">Bds::DataFileTapeDigitiser</a> (This class implements the TapeDigitiser's file output conversion and storing system ) . . . . .	102
<a href="#">Bds::DataFileWra</a> . . . . .	105
<a href="#">Bds::DataFileWraAgso</a> . . . . .	109
<a href="#">Bds::DataFormats</a> (This class defines the interface for generic data file access ) . . . . .	112
<a href="#">Bds::DataHandle</a> (This defines a handle to a data stream/file ) . . . . .	113
<a href="#">Bds::DataInfo</a> . . . . .	114

---

Bds::Digitiser (This class defines a <a href="#">Digitiser</a> ) . . . . .	116
Bds::Fap (This class defines a Amplitude/Phase <a href="#">Response</a> ) . . . . .	119
Bds::FileDataHeaderTape_1v1 . . . . .	121
Bds::FileDataHeaderTape_2v0 . . . . .	124
Bds::Group (This holds information on a groups) . . . . .	127
Bds::ListRange (This class defines a <a href="#">TimePeriod</a> ) . . . . .	129
Bds::Location (This class defines a <a href="#">Location</a> . This will be the stations location) . . . . .	131
Bds::Network (This class defines a <a href="#">Network</a> organisation. Could have other information here like website, contacts etc) . . . . .	134
Bds::Note (This holds information on a <a href="#">Note</a> ) . . . . .	136
Bds::Point . . . . .	138
Bds::PoleZero (This class defines a Pole/Zero <a href="#">Response</a> ) . . . . .	139
Bds::Response (This class defines a <a href="#">Response</a> ) . . . . .	140
Bds::ResponseEdit ( <a href="#">Response</a> Editor object) . . . . .	144
Bds::Selection . . . . .	145
Bds::SelectionChannel . . . . .	147
Bds::SelectionInfo (This class defines the set of data selection criteria) . . . . .	148
Bds::Sensor (This class defines a <a href="#">Sensor</a> ) . . . . .	150
Bds::SensorLocation (This class defines a <a href="#">Sensor</a> location) . . . . .	153
Bds::Station . . . . .	155
Bds::TimePeriod . . . . .	157
Bds::User (This holds information on a user) . . . . .	158

# Chapter 6

## File Index

### 6.1 File List

Here is a list of all files with brief descriptions:

<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp</a>	161
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsCompress.h</a>	162
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp</a>	163
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h</a>	164
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp</a>	165
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h</a>	166
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp</a>	167
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h</a>	168
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs-1.cpp</a>	169
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs-1.h</a>	170
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp</a>	171
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h</a>	172
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp</a>	173
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h</a>	174
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp</a>	175
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h</a>	176
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp</a>	177
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h</a>	178
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp</a>	179
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h</a>	180
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp</a>	181
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h</a>	182
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra-1.cpp</a>	183
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra-1.h</a>	184
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp</a>	185
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h</a>	186
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp</a>	187
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h</a>	188
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp</a>	189
<a href="#">/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h</a>	190
<a href="#">/src/blacknest/bds/bds/bdsDataLib/canada_compress.c</a>	191
<a href="#">/src/blacknest/bds/bds/bdsDataLib/canada_compress.h</a>	192
<a href="#">BdsC.cc</a>	193

---

BdsC.h	194
BdsD.cc	195
BdsD.h	196
BdsLib.cpp	199
BdsLib.h	200
BdsS.cc	201
BdsT.cc	202

# Chapter 7

## Directory Documentation

### 7.1 /src/blacknest/bds/bds/bdsDataLib/ Directory Reference

#### Files

- file [BdsCompress.cpp](#)
- file [BdsCompress.h](#)
- file [BdsDataCollate.cpp](#)
- file [BdsDataCollate.h](#)
- file [BdsDataFile.cpp](#)
- file [BdsDataFile.h](#)
- file [BdsDataFileAscii.cpp](#)
- file [BdsDataFileAscii.h](#)
- file [BdsDataFileBdrs-1.cpp](#)
- file [BdsDataFileBdrs-1.h](#)
- file [BdsDataFileBdrs.cpp](#)
- file [BdsDataFileBdrs.h](#)
- file [BdsDataFileBds.cpp](#)
- file [BdsDataFileBds.h](#)
- file [BdsDataFileBknas.cpp](#)
- file [BdsDataFileBknas.h](#)
- file [BdsDataFileGcf.cpp](#)
- file [BdsDataFileGcf.h](#)
- file [BdsDataFileIms.cpp](#)
- file [BdsDataFileIms.h](#)
- file [BdsDataFileTapeDigitiser.cpp](#)
- file [BdsDataFileTapeDigitiser.h](#)
- file [BdsDataFileWra-1.cpp](#)
- file [BdsDataFileWra-1.h](#)
- file [BdsDataFileWra.cpp](#)
- file [BdsDataFileWra.h](#)
- file [BdsDataFileWraAgso.cpp](#)
- file [BdsDataFileWraAgso.h](#)
- file [BdsDataLib.cpp](#)
- file [BdsDataLib.h](#)
- file [canada\\_compress.c](#)
- file [canada\\_compress.h](#)



# Chapter 8

## Namespace Documentation

### 8.1 Bds Namespace Reference

#### Classes

- class [DataAccess](#)
- class [DataAddAccess](#)

*This is the DataAdd Access API interface.*

- class [AdminAccess](#)

*This is the AdminAccess API interface.*

- class [Point](#)
- class [TimePeriod](#)
- class [ListRange](#)

*This class defines a TimePeriod.*

- class [Network](#)

*This class defines a Network organisation. Could have other information here like website, contacts etc.*

- class [ArrayChannel](#)

*This class defines an arrays channel.*

- class [Station](#)
- class [Location](#)

*This class defines a Location. This will be the stations location.*

- class [PoleZero](#)

*This class defines a Pole/Zero Response.*

- class [Fap](#)

*This class defines a Amplitude/Phase Response.*

- class [Response](#)

*This class defines a Response.*

- class [Calibration](#)  
*This class defines a calibration setting.*
- class [Digitiser](#)  
*This class defines a [Digitiser](#).*
- class [Sensor](#)  
*This class defines a [Sensor](#).*
- class [SensorLocation](#)  
*This class defines a [Sensor](#) location.*
- class [ChannelInstrument](#)
- class [Channel](#)  
*This class defines a [Channel](#).*
- class [SelectionInfo](#)  
*This class defines the set of data selection criteria.*
- class [SelectionChannel](#)
- class [Selection](#)
- class [ChannelInfo](#)
- class [DataChannel](#)  
*This class defines information on a single channels set of data.*
- class [DataInfo](#)
- class [DataHandle](#)  
*This defines a handle to a data stream/file.*
- class [DataBlock](#)
- class [User](#)  
*This holds information on a user.*
- class [Group](#)  
*This holds information on a groups.*
- class [AccessGroup](#)  
*This holds information on data access groups.*
- class [Change](#)  
*This holds information on a change.*
- class [ChangeGroup](#)  
*This holds information on a change.*
- class [Note](#)  
*This holds information on a [Note](#).*
- class [ResponseEdit](#)

*Response Editor object.*

- class [DataCollate](#)

*This class defines the interface for generic data file access.*

- class [DataFile](#)

*This class defines the interface for generic data file access.*

- class [DataFileAscii](#)
- class [DataFileBdrs](#)
- struct [BdsDataBlockHeader](#)
- struct [BdsDataBlock](#)
- struct [BdsDataPacketHeader](#)
- class [BdsDataPacket](#)
- class [BdsDataStreamlet](#)
- class [DataFileBds](#)

*This class implements the BDS Data File/Stream access system.*

- class [DataFileBknas](#)
- class [DataFileGcf](#)
- class [DataFileImms](#)
- struct [FileDataHeaderTape\\_1v1](#)
- struct [FileDataHeaderTape\\_2v0](#)
- class [DataFileTapeDigitiser](#)

*This class implements the TapeDigitiser's file output conversion and storing system.*

- class [DataFileWra](#)
- class [DataFileWraAgso](#)
- class [DataFormats](#)

*This class defines the interface for generic data file access.*

## Enumerations

- enum [Errors](#) {  
[ErrorOk](#), [ErrorMisc](#), [ErrorWarning](#), [ErrorEndOfFile](#),  
[ErrorFile](#), [ErrorInit](#), [ErrorConfig](#), [ErrorResourceLimit](#),  
[ErrorParam](#), [ErrorNotImplemented](#), [ErrorComms](#), [ErrorTimeout](#),  
[ErrorValidate](#), [ErrorValidateMissingBlocks](#), [ErrorValidateTimeBackwards](#), [ErrorValidateFilename-](#)  
[Time](#),  
[ErrorValidateMetaData](#), [ErrorValidateFix](#), [ErrorFormat](#), [ErrorTimeStamp](#),  
[ErrorEndOfData](#), [ErrorNoData](#), [ErrorDataPresent](#), [ErrorAccessDenied](#) }
- enum [Priority](#) { [PriorityLow](#), [PriorityNormal](#), [PriorityHigh](#) }
- enum [SelectionGroup](#) { [SelectionGroupData](#), [SelectionGroupMetaData](#) }
- enum [SampleFormat](#) { [SampleFormatUnknown](#), [SampleFormatInt16](#), [SampleFormatInt32](#), [Sample-](#)  
[FormatFloat32](#) }
- enum [BdsDataType](#) { [BdsDataTypeBlock](#) = 0x42534442, [BdsDataTypeInfo](#) = 0x30534442, [Bds-](#)  
[DataTypeData](#) = 0x31534442 }
- enum [FileHeaderType](#) { [FileHeaderType\\_Standard](#) = 1, [FileHeaderType\\_TapeDigitiser](#) = 10 }

- enum `FileSampleType` {  
`FileSampleType_Unknown`, `FileSampleType_Float32`, `FileSampleType_Float64`, `FileSampleType_UnknownInt16`,  
`FileSampleType_Int32` }

## Functions

- void `bdsChannelGetTypeAux` (BString name, BString &type, BString &aux)
- BString `bdsChannelGetName` (BString type, BString aux)
- BError `bdsDataInfoSetTimeRange` (`DataInfo` &dataInfo)
- BError `bdsDataInfoFromInfo` (BDictString info, `DataInfo` &dataInfo)  
*Convert info to `DataInfo`.*
- BError `bdsInfoFromDataInfo` (`DataInfo` &dataInfo, BDictString &info)
- void `bdsDumpDataInfo` (`DataInfo` &dataInfo)
- BError `bdsUnCompressCm8` (BUInt8 \*buffer, BUInt n, BArray< BInt32 > &data)
- BUInt32 `crc` (BUInt32 crc, void \*data, int numBytes)
- void `dataCalculateDifference` (BInt32 &prevValue, BArray< BInt32 > &data)
- BInt32 `dataChecksum` (BInt32 checksum, BArray< BInt32 > &data)
- BError `dataCompressCm6` (int &prevValue1, int &prevValue2, BArray< BInt32 > &data, BString &d)
- void `dataConvert` (BArray< BFloat64 > &dataIn, BArray< BInt32 > &dataOut)

## Variables

- const BUInt32 `apiVersion` = 0
- const BString `BdsVersion` = "1.0.0"
- static char `cm6Table` [64]
- const double `Scale` = 16777216.0
- `DataFormats` `dataFormats`

## 8.1.1 Enumeration Type Documentation

### 8.1.1.1 enum Bds::BdsDataType

Enumerator:

*BdsDataTypeBlock*

*BdsDataTypeInfo*

*BdsDataTypeData*

### 8.1.1.2 enum Bds::Errors

Enumerator:

*ErrorOk*

*ErrorMisc*

*ErrorWarning*

*ErrorEndOfFile*  
*ErrorFile*  
*ErrorInit*  
*ErrorConfig*  
*ErrorResourceLimit*  
*ErrorParam*  
*ErrorNotImplemented*  
*ErrorComms*  
*ErrorTimeout*  
*ErrorValidate*  
*ErrorValidateMissingBlocks*  
*ErrorValidateTimeBackwards*  
*ErrorValidateFilenameTime*  
*ErrorValidateMetaData*  
*ErrorValidateFix*  
*ErrorFormat*  
*ErrorTimeStamp*  
*ErrorEndOfData*  
*ErrorNoData*  
*ErrorDataPresent*  
*ErrorAccessDenied*

#### 8.1.1.3 enum Bds::FileHeaderType

Enumerator:

*FileHeaderType\_Standard*  
*FileHeaderType\_TapeDigitiser*

#### 8.1.1.4 enum Bds::FileSampleType

Enumerator:

*FileSampleType\_Unknown*  
*FileSampleType\_Float32*  
*FileSampleType\_Float64*  
*FileSampleType\_Int16*  
*FileSampleType\_Int32*

#### 8.1.1.5 enum Bds::Priority

Enumerator:

*PriorityLow*  
*PriorityNormal*  
*PriorityHigh*

#### 8.1.1.6 enum Bds::SampleFormat

Enumerator:

*SampleFormatUnknown*

*SampleFormatInt16*

*SampleFormatInt32*

*SampleFormatFloat32*

#### 8.1.1.7 enum Bds::SelectionGroup

Enumerator:

*SelectionGroupData*

*SelectionGroupMetaData*

### 8.1.2 Function Documentation

8.1.2.1 BString Bds::bdsChannelGetName (BString *type*, BString *aux*)

8.1.2.2 void Bds::bdsChannelGetTypeAux (BString *name*, BString & *type*, BString & *aux*)

8.1.2.3 BError Bds::bdsDataInfoFromInfo (BDictString *info*, DataInfo & *dataInfo*)

Convert info to [DataInfo](#).

- 8.1.2.4 **BError** Bds::bdsDataInfoSetTimeRange (DataInfo & *dataInfo*)
- 8.1.2.5 **void** Bds::bdsDumpDataInfo (DataInfo & *dataInfo*)
- 8.1.2.6 **BError** Bds::bdsInfoFromDataInfo (DataInfo & *dataInfo*, BDictString & *info*)
- 8.1.2.7 **BError** Bds::bdsUnCompressCm8 (BUInt8 \* *buffer*, BUInt n, BArray< BInt32 > & *data*)
- 8.1.2.8 **BUInt32** Bds::crc (BUInt32 *crc*, void \* *data*, int *numBytes*)
- 8.1.2.9 **void** Bds::dataCalculateDifference (BInt32 & *prevValue*, BArray< BInt32 > & *data*)
- 8.1.2.10 **BInt32** Bds::dataChecksum (BInt32 *checksum*, BArray< BInt32 > & *data*)
- 8.1.2.11 **BError** Bds::dataCompressCm6 (int & *prevValue1*, int & *prevValue2*, BArray< BInt32 > & *data*, BString & *d*)
- 8.1.2.12 **void** Bds::dataConvert (BArray< BFloat64 > & *dataIn*, BArray< BInt32 > & *dataOut*)

### 8.1.3 Variable Documentation

- 8.1.3.1 **const** BUInt32 Bds::apiVersion = 0
- 8.1.3.2 **const** BString Bds::BdsVersion = "1.0.0"
- 8.1.3.3 **char** Bds::cm6Table[64] [static]

**Initial value:**

```
{
    '+', '-', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B',
    'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P',
    'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b', 'c', 'd',
    'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',
    's', 't', 'u', 'v', 'w', 'x', 'y', 'z'
}
```

- 8.1.3.4 **DataFormats** Bds::dataFormats
- 8.1.3.5 **const double** Bds::Scale = 16777216.0



# Chapter 9

## Class Documentation

### 9.1 Bds::AccessGroup Class Reference

This holds information on data access groups.

```
#include <BdsD.h>
```

#### Public Member Functions

- [AccessGroup](#) (BUInt32 pid=BUInt32(), BString pgroup=BString(), BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BString pnetwork=BString(), BString pstation=BString())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

#### Public Attributes

- BUInt32 [id](#)  
*The unique id.*
- BString [group](#)  
*The Group name.*
- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time.*
- BString [network](#)  
*The Network Name.*
- BString [station](#)  
*The Station name.*

## 9.1.1 Detailed Description

This holds information on data access groups.

## 9.1.2 Constructor & Destructor Documentation

**9.1.2.1 Bds::AccessGroup::AccessGroup (BUInt32 *pid* = BUInt32 (), BString *pgroup* = BString (), BTimeStamp *pstartTime* = BTimeStamp (), BTimeStamp *pendTime* = BTimeStamp (), BString *pnetwork* = BString (), BString *pstation* = BString ())**

## 9.1.3 Member Function Documentation

**9.1.3.1 BString Bds::AccessGroup::getType ()**

**9.1.3.2 BError Bds::AccessGroup::setMembers (BDictString & *members*)**

**9.1.3.3 BError Bds::AccessGroup::getMembers (BDictString & *members*)**

## 9.1.4 Member Data Documentation

**9.1.4.1 BUInt32 Bds::AccessGroup::id**

The unique id.

**9.1.4.2 BString Bds::AccessGroup::group**

The [Group](#) name.

**9.1.4.3 BTimeStamp Bds::AccessGroup::startTime**

The Start Time.

**9.1.4.4 BTimeStamp Bds::AccessGroup::endTime**

The End Time.

**9.1.4.5 BString Bds::AccessGroup::network**

The [Network](#) Name.

**9.1.4.6 BString Bds::AccessGroup::station**

The [Station](#) name.

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

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

## 9.2 Bds::AdminAccess Class Reference

This is the [AdminAccess](#) API interface.

```
#include <BdsC.h>
```

### Public Member Functions

- [AdminAccess](#) (BString name="")
- BError [connect](#) (BString user, BString password)  
*Provides user/password information.*
- BError [setUser](#) (BString user, BString email)  
*Sets user to given name or email.*
- BError [setUserReal](#) ()  
*Sets user back to real user.*
- BError [getVersion](#) (BString &version)  
*Gets the software version.*
- BError [userGetList](#) (BList< [User](#) > &users)  
*Get list of Users.*
- BError [userUpdate](#) (BInt32 append, [User](#) user, BUInt32 &id)  
*Update or append a user entry.*
- BError [userDelete](#) (BUInt32 id)  
*Delete a user entry.*
- BError [userGetGroups](#) (BList< BString > &groups)
- BError [groupGetList](#) (BList< [Group](#) > &groups)  
*Get list of groups the user belongs to.*
- BError [groupUpdate](#) (BInt32 append, [Group](#) group, BUInt32 &id)  
*Update or append a group entry.*
- BError [groupDelete](#) (BUInt32 id)  
*Delete a group entry.*
- BError [accessGroupGetList](#) (BList< [AccessGroup](#) > &accessGroups)  
*Get list of AccessGroups.*
- BError [accessGroupUpdate](#) (BInt32 append, [AccessGroup](#) group, BUInt32 &id)  
*Update or append an [AccessGroup](#) entry.*
- BError [accessGroupDelete](#) (BUInt32 id)  
*Delete an [AccessGroup](#) entry.*
- BError [getSelectionInfo](#) ([SelectionGroup](#) group, [SelectionInfo](#) &selectionInfo)

*Get selection info.*

- BError [getSelections](#) ([SelectionGroup](#) group, [Selection](#) selectionIn, [Selection](#) &selectionOut)  
*Get selection list.*
- BError [networkGetList](#) (BList< [Network](#) > &networks)  
*Get list of Networks.*
- BError [networkUpdate](#) (BInt32 append, [Network](#) network, BUInt32 &id)
- BError [networkDelete](#) (BUInt32 id)
- BError [stationGetList](#) ([Selection](#) sel, BList< [Station](#) > &stations)  
*Get list of Stations.*
- BError [stationUpdate](#) (BInt32 append, [Station](#) station, BUInt32 &id)
- BError [stationDelete](#) (BUInt32 id)
- BError [locationGetList](#) ([Selection](#) sel, BList< [Location](#) > &locations)  
*Get list of Station Locations.*
- BError [locationUpdate](#) (BInt32 append, [Location](#) location, BUInt32 &id)
- BError [locationDelete](#) (BUInt32 id)
- BError [channelGetList](#) ([Selection](#) sel, BList< [Channel](#) > &channels)  
*Get list of Channels.*
- BError [channelGet](#) (BUInt32 id, [Channel](#) &channel)  
*Get a channel.*
- BError [channelUpdate](#) (BInt32 append, [Channel](#) channel, BUInt32 &id)
- BError [channelDelete](#) (BUInt32 id)
- BError [channelInstrumentGetList](#) ([Selection](#) sel, BList< [ChannelInstrument](#) > &channelInstruments)  
*Get list of Channel Instruments.*
- BError [channelInstrumentUpdate](#) (BInt32 append, [ChannelInstrument](#) channelInstrument, BUInt32 &id)
- BError [channelInstrumentDelete](#) (BUInt32 id)
- BError [digitiserGetList](#) ([Selection](#) sel, BList< [Digitiser](#) > &digitisers)  
*Get list of Channels.*
- BError [digitiserGet](#) (BUInt32 id, [Digitiser](#) &digitiser)
- BError [digitiserUpdate](#) (BInt32 append, [Digitiser](#) digitiser, BUInt32 &id)
- BError [digitiserDelete](#) (BUInt32 id)
- BError [sensorGetList](#) ([Selection](#) sel, BList< [Sensor](#) > &sensors)  
*Get list of Channels.*
- BError [sensorGet](#) (BUInt32 id, [Sensor](#) &sensor)
- BError [sensorUpdate](#) (BInt32 append, [Sensor](#) sensor, BUInt32 &id)
- BError [sensorDelete](#) (BUInt32 id)
- BError [sensorLocationGetList](#) ([Selection](#) sel, BList< [SensorLocation](#) > &sensorLocations)  
*Get list of Sensor Locations.*

- BError [sensorLocationUpdate](#) (BInt32 append, [SensorLocation](#) sensorLocation, BUInt32 &id)
- BError [sensorLocationDelete](#) (BUInt32 id)
- BError [calibrationGetList](#) ([Selection](#) sel, BList< [Calibration](#) > &calibrations)  
*Get list of Channels.*
- BError [calibrationUpdate](#) (BInt32 append, [Calibration](#) calibration, BUInt32 &id)
- BError [calibrationDelete](#) (BUInt32 id)
- BError [responseGetList](#) ([Selection](#) sel, BList< [Response](#) > &responses)  
*Get list of Channels.*
- BError [responseUpdate](#) (BInt32 append, [Response](#) response, BUInt32 &id)
- BError [responseDelete](#) (BUInt32 id)
- BError [dataSearch](#) ([Selection](#) selection, [DataInfo](#) &dataInfo)  
*Search for data matching the given selection parameters.*
- BError [dataGetChannelInfo](#) ([DataInfo](#) dataInfo, BList< [ChannelInfo](#) > &channelInfos)  
*Return the channel MetaData in structured form.*
- BError [dataOpen](#) ([DataInfo](#) dataInfo, BString mode, BString format, [DataHandle](#) &dataHandle)  
*Open a data stream.*
- BError [dataGetInfo](#) ([DataHandle](#) dataHandle, [DataInfo](#) &dataInfo)  
*Get information on the data stream.*
- BError [dataGetWarnings](#) ([DataHandle](#) dataHandle, BList< BString > &warnings)  
*Get information on the data stream.*
- BError [dataGetBlock](#) ([DataHandle](#) dataHandle, BUInt32 channel, BUInt32 segment, BUInt32 block-Number, [DataBlock](#) &data)  
*Return a block of data.*
- BError [dataSeekBlock](#) ([DataHandle](#) dataHandle, BUInt32 channel, BUInt32 segment, BTimeStamp time, BUInt32 &blockNumber)  
*Searches for a data block matching the time given.*
- BError [dataSetInfo](#) ([DataHandle](#) dataHandle, [DataInfo](#) dataInfo)  
*Set the info when writing to a stream.*
- BError [dataPutBlock](#) ([DataHandle](#) dataHandle, [DataBlock](#) data)  
*Send a block of data.*
- BError [dataClose](#) ([DataHandle](#) dataHandle)  
*Close a stream.*
- BError [dataFormattedRead](#) ([DataHandle](#) dataHandle, BUInt32 number, BArray< BUInt8 > &data)  
*Read the raw data from the stream.*
- BError [dataFormattedGetLength](#) ([DataHandle](#) dataHandle, BUInt64 &length)  
*Read the raw data from the stream.*

- BError [changeGroupStart](#) ([ChangeGroup](#) changeGroup)
- BError [changeGroupEnd](#) ()
- BError [changeGroupGetList](#) ([ListRange](#) range, BList< [ChangeGroup](#) > &changeGroups)
- BError [changeGroupDelete](#) (BTimeStamp beforeDate, BString type, BInt32 empty)
- BError [changeGetListNumber](#) (BUInt32 id, BUInt32 &number)
- BError [changeGetList](#) (BUInt32 id, [ListRange](#) range, BList< [Change](#) > &changes)
- BError [changeDelete](#) (BTimeStamp beforeDate, BString type)
- BError [noteGetList](#) ([ListRange](#) range, BList< [Note](#) > &notes)
- BError [noteUpdate](#) (BInt32 append, [Note](#) note, BUInt32 &id)
- BError [noteDelete](#) (BUInt32 id)
- BError [statisticsGet](#) (BDict< BString > &info)
- BError [sqlQuery](#) (BString query, BList< BDict< BString > > &result)

*A low level SQL access function.*

## 9.2.1 Detailed Description

This is the [AdminAccess](#) API interface.

## 9.2.2 Constructor & Destructor Documentation

### 9.2.2.1 Bds::AdminAccess::AdminAccess (BString *name* = "")

## 9.2.3 Member Function Documentation

### 9.2.3.1 BError Bds::AdminAccess::connect (BString *user*, BString *password*)

Provides user/password information.

### 9.2.3.2 BError Bds::AdminAccess::setUser (BString *user*, BString *email*)

Sets user to given name or email.

### 9.2.3.3 BError Bds::AdminAccess::setUserReal ()

Sets user back to real user.

### 9.2.3.4 BError Bds::AdminAccess::getVersion (BString & *version*)

Gets the software version.

### 9.2.3.5 BError Bds::AdminAccess::userGetList (BList< User > & *users*)

Get list of Users.

**9.2.3.6 BError Bds::AdminAccess::userUpdate (BInt32 *append*, User *user*, BUInt32 & *id*)**

Update or append a user entry.

**9.2.3.7 BError Bds::AdminAccess::userDelete (BUInt32 *id*)**

Delete a user entry.

**9.2.3.8 BError Bds::AdminAccess::userGetGroups (BList< BString > & *groups*)****9.2.3.9 BError Bds::AdminAccess::groupGetList (BList< Group > & *groups*)**

Get list of groups the user belongs to.

Get list of Groups

**9.2.3.10 BError Bds::AdminAccess::groupUpdate (BInt32 *append*, Group *group*, BUInt32 & *id*)**

Update or append a group entry.

**9.2.3.11 BError Bds::AdminAccess::groupDelete (BUInt32 *id*)**

Delete a group entry.

**9.2.3.12 BError Bds::AdminAccess::accessGroupGetList (BList< AccessGroup > & *accessGroups*)**

Get list of AccessGroups.

**9.2.3.13 BError Bds::AdminAccess::accessGroupUpdate (BInt32 *append*, AccessGroup *group*, BUInt32 & *id*)**

Update or append an [AccessGroup](#) entry.

**9.2.3.14 BError Bds::AdminAccess::accessGroupDelete (BUInt32 *id*)**

Delete an [AccessGroup](#) entry.

**9.2.3.15 BError Bds::AdminAccess::getSelectionInfo (SelectionGroup *group*, SelectionInfo & *selectionInfo*)**

Get selection info.

**9.2.3.16 BError Bds::AdminAccess::getSelections (SelectionGroup *group*, Selection *selectionIn*, Selection & *selectionOut*)**

Get selection list.

**9.2.3.17 BError Bds::AdminAccess::networkGetList (BList< Network > & networks)**

Get list of Networks.

**9.2.3.18 BError Bds::AdminAccess::networkUpdate (BInt32 *append*, Network *network*, BUInt32 & *id*)****9.2.3.19 BError Bds::AdminAccess::networkDelete (BUInt32 *id*)****9.2.3.20 BError Bds::AdminAccess::stationGetList (Selection *sel*, BList< Station > & *stations*)**

Get list of Stations.

**9.2.3.21 BError Bds::AdminAccess::stationUpdate (BInt32 *append*, Station *station*, BUInt32 & *id*)****9.2.3.22 BError Bds::AdminAccess::stationDelete (BUInt32 *id*)****9.2.3.23 BError Bds::AdminAccess::locationGetList (Selection *sel*, BList< Location > & *locations*)**

Get list of [Station](#) Locations.

**9.2.3.24 BError Bds::AdminAccess::locationUpdate (BInt32 *append*, Location *location*, BUInt32 & *id*)****9.2.3.25 BError Bds::AdminAccess::locationDelete (BUInt32 *id*)****9.2.3.26 BError Bds::AdminAccess::channelGetList (Selection *sel*, BList< Channel > & *channels*)**

Get list of Channels.

**9.2.3.27 BError Bds::AdminAccess::channelGet (BUInt32 *id*, Channel & *channel*)**

Get a channel.

**9.2.3.28 BError Bds::AdminAccess::channelUpdate (BInt32 *append*, Channel *channel*, BUInt32 & *id*)****9.2.3.29 BError Bds::AdminAccess::channelDelete (BUInt32 *id*)****9.2.3.30 BError Bds::AdminAccess::channelInstrumentGetList (Selection *sel*, BList< ChannelInstrument > & *channelInstruments*)**

Get list of [Channel](#) Instruments.

**9.2.3.31** BError Bds::AdminAccess::channelInstrumentUpdate (BInt32 *append*, ChannelInstrument *channelInstrument*, BUInt32 & *id*)

**9.2.3.32** BError Bds::AdminAccess::channelInstrumentDelete (BUInt32 *id*)

**9.2.3.33** BError Bds::AdminAccess::digitiserGetList (Selection *sel*, BList< Digitiser > & *digitisers*)

Get list of Channels.

**9.2.3.34** BError Bds::AdminAccess::digitiserGet (BUInt32 *id*, Digitiser & *digitiser*)

**9.2.3.35** BError Bds::AdminAccess::digitiserUpdate (BInt32 *append*, Digitiser *digitiser*, BUInt32 & *id*)

**9.2.3.36** BError Bds::AdminAccess::digitiserDelete (BUInt32 *id*)

**9.2.3.37** BError Bds::AdminAccess::sensorGetList (Selection *sel*, BList< Sensor > & *sensors*)

Get list of Channels.

**9.2.3.38** BError Bds::AdminAccess::sensorGet (BUInt32 *id*, Sensor & *sensor*)

**9.2.3.39** BError Bds::AdminAccess::sensorUpdate (BInt32 *append*, Sensor *sensor*, BUInt32 & *id*)

**9.2.3.40** BError Bds::AdminAccess::sensorDelete (BUInt32 *id*)

**9.2.3.41** BError Bds::AdminAccess::sensorLocationGetList (Selection *sel*, BList< SensorLocation > & *sensorLocations*)

Get list of [Sensor](#) Locations.

**9.2.3.42** BError Bds::AdminAccess::sensorLocationUpdate (BInt32 *append*, SensorLocation *sensorLocation*, BUInt32 & *id*)

**9.2.3.43** BError Bds::AdminAccess::sensorLocationDelete (BUInt32 *id*)

**9.2.3.44** BError Bds::AdminAccess::calibrationGetList (Selection *sel*, BList< Calibration > & *calibrations*)

Get list of Channels.

**9.2.3.45** **BError Bds::AdminAccess::calibrationUpdate** (**BInt32** *append*, **Calibration** *calibration*, **BUInt32** & *id*)

**9.2.3.46** **BError Bds::AdminAccess::calibrationDelete** (**BUInt32** *id*)

**9.2.3.47** **BError Bds::AdminAccess::responseGetList** (**Selection** *sel*, **BList**< **Response** > & *responses*)

Get list of Channels.

**9.2.3.48** **BError Bds::AdminAccess::responseUpdate** (**BInt32** *append*, **Response** *response*, **BUInt32** & *id*)

**9.2.3.49** **BError Bds::AdminAccess::responseDelete** (**BUInt32** *id*)

**9.2.3.50** **BError Bds::AdminAccess::dataSearch** (**Selection** *selection*, **DataInfo** & *dataInfo*)

Search for data matching the given selection parameters.

**9.2.3.51** **BError Bds::AdminAccess::dataGetChannelInfo** (**DataInfo** *dataInfo*, **BList**< **ChannelInfo** > & *channelInfos*)

Return the channel MetaData in structured form.

**9.2.3.52** **BError Bds::AdminAccess::dataOpen** (**DataInfo** *dataInfo*, **BString** *mode*, **BString** *format*, **DataHandle** & *dataHandle*)

Open a data stream.

**9.2.3.53** **BError Bds::AdminAccess::dataGetInfo** (**DataHandle** *dataHandle*, **DataInfo** & *dataInfo*)

Get information on the data stream.

**9.2.3.54** **BError Bds::AdminAccess::dataGetWarnings** (**DataHandle** *dataHandle*, **BList**< **BString** > & *warnings*)

Get information on the data stream.

**9.2.3.55** **BError Bds::AdminAccess::dataGetBlock** (**DataHandle** *dataHandle*, **BUInt32** *channel*, **BUInt32** *segment*, **BUInt32** *blockNumber*, **DataBlock** & *data*)

Return a block of data.

**9.2.3.56** **BError Bds::AdminAccess::dataSeekBlock** (**DataHandle** *dataHandle*, **BUInt32** *channel*, **BUInt32** *segment*, **BTimeStamp** *time*, **BUInt32** & *blockNumber*)

Searches for a data block matching the time given.

**9.2.3.57 BError Bds::AdminAccess::dataSetInfo (DataHandle *dataHandle*, DataInfo *dataInfo*)**

Set the info when writing to a stream.

**9.2.3.58 BError Bds::AdminAccess::dataPutBlock (DataHandle *dataHandle*, DataBlock *data*)**

Send a block of data.

**9.2.3.59 BError Bds::AdminAccess::dataClose (DataHandle *dataHandle*)**

Close a stream.

**9.2.3.60 BError Bds::AdminAccess::dataFormattedRead (DataHandle *dataHandle*, BUInt32 *number*, BArray< BUInt8 > & *data*)**

Read the raw data from the stream.

**9.2.3.61 BError Bds::AdminAccess::dataFormattedGetLength (DataHandle *dataHandle*, BUInt64 & *length*)**

Read the raw data from the stream.

- 9.2.3.62 **BError Bds::AdminAccess::changeGroupStart** (*ChangeGroup changeGroup*)
- 9.2.3.63 **BError Bds::AdminAccess::changeGroupEnd** ()
- 9.2.3.64 **BError Bds::AdminAccess::changeGroupGetList** (*ListRange range*, *BList< ChangeGroup > & changeGroups*)
- 9.2.3.65 **BError Bds::AdminAccess::changeGroupDelete** (*BTimeStamp beforeDate*, *BString type*, *BInt32 empty*)
- 9.2.3.66 **BError Bds::AdminAccess::changeGetListNumber** (*BUInt32 id*, *BUInt32 & number*)
- 9.2.3.67 **BError Bds::AdminAccess::changeGetList** (*BUInt32 id*, *ListRange range*, *BList< Change > & changes*)
- 9.2.3.68 **BError Bds::AdminAccess::changeDelete** (*BTimeStamp beforeDate*, *BString type*)
- 9.2.3.69 **BError Bds::AdminAccess::noteGetList** (*ListRange range*, *BList< Note > & notes*)
- 9.2.3.70 **BError Bds::AdminAccess::noteUpdate** (*BInt32 append*, *Note note*, *BUInt32 & id*)
- 9.2.3.71 **BError Bds::AdminAccess::noteDelete** (*BUInt32 id*)
- 9.2.3.72 **BError Bds::AdminAccess::statisticsGet** (*BDict< BString > & info*)
- 9.2.3.73 **BError Bds::AdminAccess::sqlQuery** (*BString query*, *BList< BDict< BString > > & result*)

A low level SQL access function.

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

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

## 9.3 Bds::ArrayChannel Class Reference

This class defines an arrays channel.

```
#include <BdsD.h>
```

### Public Member Functions

- [ArrayChannel](#) (BString pstation=BString(), BString pchannel=BString())

### Public Attributes

- BString [station](#)  
*The Stations name.*
- BString [channel](#)  
*The Channels name.*

#### 9.3.1 Detailed Description

This class defines an arrays channel.

#### 9.3.2 Constructor & Destructor Documentation

**9.3.2.1** [Bds::ArrayChannel::ArrayChannel](#) (BString *pstation* = BString(), BString *pchannel* = BString())

#### 9.3.3 Member Data Documentation

##### 9.3.3.1 BString Bds::ArrayChannel::station

The Stations name.

##### 9.3.3.2 BString Bds::ArrayChannel::channel

The Channels name.

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

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

## 9.4 Bds::BdsDataBlock Struct Reference

```
#include <BdsDataFileBds.h>
```

### Public Attributes

- [BdsDataBlockHeader](#) header
- char [data](#) []

*The packet data.*

### 9.4.1 Member Data Documentation

#### 9.4.1.1 BdsDataBlockHeader Bds::BdsDataBlock::header

#### 9.4.1.2 char Bds::BdsDataBlock::data[]

The packet data.

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

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h](#)

## 9.5 Bds::BdsDataBlockHeader Struct Reference

```
#include <BdsDataFileBds.h>
```

### Public Attributes

- BUInt32 [type](#)  
*Blocks type, contains magic number for synchronisation.*
- BUInt32 [length](#)  
*Packet length.*
- BUInt32 [packetOffset](#)  
*Offset to first packet header within block or 0 if no packet header within block.*

### 9.5.1 Member Data Documentation

#### 9.5.1.1 BUInt32 Bds::BdsDataBlockHeader::type

Blocks type, contains magic number for synchronisation.

#### 9.5.1.2 BUInt32 Bds::BdsDataBlockHeader::length

Packet length.

#### 9.5.1.3 BUInt32 Bds::BdsDataBlockHeader::packetOffset

Offset to first packet header within block or 0 if no packet header within block.

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

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h](#)

## 9.6 Bds::BdsDataPacket Class Reference

```
#include <BdsDataFileBds.h>
```

### Public Member Functions

- [BdsDataPacket \(\)](#)
- [~BdsDataPacket \(\)](#)
- void [clear \(\)](#)
- void [setChecksumAndLength \(\)](#)
- BError [validateChecksum \(\)](#)
- BError [setHeader \(BdsDataPacketHeader &header\)](#)
- BError [getHeader \(BdsDataPacketHeader &header\)](#)
- void [dump \(\)](#)

### 9.6.1 Constructor & Destructor Documentation

**9.6.1.1** [Bds::BdsDataPacket::BdsDataPacket \(\)](#)

**9.6.1.2** [Bds::BdsDataPacket::~~BdsDataPacket \(\)](#)

### 9.6.2 Member Function Documentation

**9.6.2.1** [void Bds::BdsDataPacket::clear \(\)](#)

**9.6.2.2** [void Bds::BdsDataPacket::setChecksumAndLength \(\)](#)

**9.6.2.3** [BError Bds::BdsDataPacket::validateChecksum \(\)](#)

**9.6.2.4** [BError Bds::BdsDataPacket::setHeader \(BdsDataPacketHeader & header\)](#)

**9.6.2.5** [BError Bds::BdsDataPacket::getHeader \(BdsDataPacketHeader & header\)](#)

**9.6.2.6** [void Bds::BdsDataPacket::dump \(\)](#)

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

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp](#)

## 9.7 Bds::BdsDataPacketHeader Struct Reference

```
#include <BdsDataFileBds.h>
```

### Public Attributes

- BUInt32 [type](#)  
*Packets type.*
- BUInt32 [length](#)  
*Length in bytes of packet.*
- BUInt32 [streamlet](#)  
*The streamlet id.*
- BUInt32 [sequence](#)  
*The streamlet packet sequence number.*
- BUInt32 [checksum](#)  
*Checksum of packet.*
- BTimeStamp [startTime](#)  
*The time of the first sample.*
- BTimeStamp [endTime](#)  
*The time of the last sample + 1.*

### 9.7.1 Member Data Documentation

#### 9.7.1.1 BUInt32 Bds::BdsDataPacketHeader::type

Packets type.

#### 9.7.1.2 BUInt32 Bds::BdsDataPacketHeader::length

Length in bytes of packet.

#### 9.7.1.3 BUInt32 Bds::BdsDataPacketHeader::streamlet

The streamlet id.

#### 9.7.1.4 BUInt32 Bds::BdsDataPacketHeader::sequence

The streamlet packet sequence number.

**9.7.1.5 BUInt32 Bds::BdsDataPacketHeader::checksum**

Checksum of packet.

**9.7.1.6 BTimeStamp Bds::BdsDataPacketHeader::startTime**

The time of the first sample.

**9.7.1.7 BTimeStamp Bds::BdsDataPacketHeader::endTime**

The time of the last sample + 1.

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

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h](#)

## 9.8 Bds::BdsDataStreamlet Class Reference

```
#include <BdsDataFileBds.h>
```

### Public Member Functions

- [BdsDataStreamlet \(\)](#)

### Public Attributes

- BUInt32 [packetNumber](#)
- BUInt64 [position](#)

### 9.8.1 Constructor & Destructor Documentation

**9.8.1.1** [Bds::BdsDataStreamlet::BdsDataStreamlet \(\)](#) [`inline`]

### 9.8.2 Member Data Documentation

**9.8.2.1** BUInt32 [Bds::BdsDataStreamlet::packetNumber](#)

**9.8.2.2** BUInt64 [Bds::BdsDataStreamlet::position](#)

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

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h](#)

## 9.9 Bds::Calibration Class Reference

This class defines a calibration setting.

```
#include <BdsD.h>
```

### Public Member Functions

- **Calibration** (BUInt32 pid=BUInt32(), BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BString pnetwork=BString(), BString pstation=BString(), BString pchannel=BString(), BFloat64 psamplingFrequency=BFloat64(), BFloat64 pcalibrationFrequency=BFloat64(), BFloat64 pcalibrationFactor=BFloat64(), BString pcalibrationUnits=BString())
- BString **getType** ()
- BError **setMembers** (BDictString &members)
- BError **getMembers** (BDictString &members)

### Public Attributes

- BUInt32 **id**  
*The ID.*
- BTimeStamp **startTime**  
*The Start Time.*
- BTimeStamp **endTime**  
*The End Time.*
- BString **network**  
*The Network/Organisation Name.*
- BString **station**  
*The station.*
- BString **channel**  
*The channel.*
- BFloat64 **samplingFrequency**  
*The sample rate used.*
- BFloat64 **calibrationFrequency**  
*The frequency that the CalibrationFactor value is valid for.*
- BFloat64 **calibrationFactor**  
*The scaling value to apply to the data to normalise to Nanometers. This is a measured value at the calibration frequency and is in Nanometers/Count.*
- BString **calibrationUnits**  
*The calibration units.*

## 9.9.1 Detailed Description

This class defines a calibration setting.

## 9.9.2 Constructor & Destructor Documentation

**9.9.2.1** **Bds::Calibration::Calibration** (**BUInt32** *pid* = **BUInt32** (), **BTimeStamp** *pstartTime* = **BTimeStamp** (), **BTimeStamp** *pendTime* = **BTimeStamp** (), **BString** *pnetwork* = **BString** (), **BString** *pstation* = **BString** (), **BString** *pchannel* = **BString** (), **BFloat64** *psamplingFrequency* = **BFloat64** (), **BFloat64** *pcalibrationFrequency* = **BFloat64** (), **BFloat64** *pcalibrationFactor* = **BFloat64** (), **BString** *pcalibrationUnits* = **BString** ())

## 9.9.3 Member Function Documentation

**9.9.3.1** **BString** **Bds::Calibration::getType** ()

**9.9.3.2** **BError** **Bds::Calibration::setMembers** (**BDictString** & *members*)

**9.9.3.3** **BError** **Bds::Calibration::getMembers** (**BDictString** & *members*)

## 9.9.4 Member Data Documentation

**9.9.4.1** **BUInt32** **Bds::Calibration::id**

The ID.

**9.9.4.2** **BTimeStamp** **Bds::Calibration::startTime**

The Start Time.

**9.9.4.3** **BTimeStamp** **Bds::Calibration::endTime**

The End Time.

**9.9.4.4** **BString** **Bds::Calibration::network**

The Network/Organisation Name.

**9.9.4.5** **BString** **Bds::Calibration::station**

The station.

**9.9.4.6** **BString** **Bds::Calibration::channel**

The channel.

#### 9.9.4.7 BFloat64 Bds::Calibration::samplingFrequency

The sample rate used.

#### 9.9.4.8 BFloat64 Bds::Calibration::calibrationFrequency

The frequency that the CalibrationFactor value is valid for.

#### 9.9.4.9 BFloat64 Bds::Calibration::calibrationFactor

The scaling value to apply to the data to normalise to Nanometers. This is a measured value at the calibration frequency and is in Nanometers/Count.

#### 9.9.4.10 BString Bds::Calibration::calibrationUnits

The calibration units.

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

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

## 9.10 Bds::Change Class Reference

This holds information on a change.

```
#include <BdsD.h>
```

### Public Member Functions

- [Change](#) (BUInt32 pid=BUInt32(), BUInt32 pchangeGroupId=BUInt32(), BTimeStamp ptime=BTimeStamp(), BString ptype=BString(), BString ptable=BString(), BUInt32 prowId=BUInt32())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*The unique id.*
- BUInt32 [changeGroupId](#)  
*The [Change](#) group ID.*
- BTimeStamp [time](#)  
*The Time the change was made.*
- BString [type](#)  
*The change type.*
- BString [table](#)  
*The database table affected.*
- BUInt32 [rowId](#)  
*The database row affected.*

#### 9.10.1 Detailed Description

This holds information on a change.

## 9.10.2 Constructor & Destructor Documentation

**9.10.2.1 Bds::Change::Change** (BUInt32 *pid* = BUInt32 (), BUInt32 *pchangeGroupId* = BUInt32 (), BTimeStamp *ptime* = BTimeStamp (), BString *ptype* = BString (), BString *ptable* = BString (), BUInt32 *provid* = BUInt32 ())

## 9.10.3 Member Function Documentation

**9.10.3.1 BString Bds::Change::getType** ()

**9.10.3.2 BError Bds::Change::setMembers** (BDictString & *members*)

**9.10.3.3 BError Bds::Change::getMembers** (BDictString & *members*)

## 9.10.4 Member Data Documentation

**9.10.4.1 BUInt32 Bds::Change::id**

The unique id.

**9.10.4.2 BUInt32 Bds::Change::changeGroupId**

The [Change](#) group ID.

**9.10.4.3 BTimeStamp Bds::Change::time**

The Time the change was made.

**9.10.4.4 BString Bds::Change::type**

The change type.

**9.10.4.5 BString Bds::Change::table**

The database table affected.

**9.10.4.6 BUInt32 Bds::Change::rowId**

The database row affected.

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

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

## 9.11 Bds::ChangeGroup Class Reference

This holds information on a change.

```
#include <BdsD.h>
```

### Public Member Functions

- [ChangeGroup](#) (BUInt32 pid=BUInt32(), BTimeStamp ptime=BTimeStamp(), BString ptype=BString(), BString puser=BString(), BString ptitle=BString(), BString pdescription=BString())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*The unique id.*
- BTimeStamp [time](#)  
*The Time the change was made.*
- BString [type](#)  
*The type of change.*
- BString [user](#)  
*The user who made the change.*
- BString [title](#)  
*The Changes title.*
- BString [description](#)  
*The Description of the change.*

#### 9.11.1 Detailed Description

This holds information on a change.

## 9.11.2 Constructor & Destructor Documentation

**9.11.2.1** **Bds::ChangeGroup::ChangeGroup** (**BUInt32** *pid* = **BUInt32** (), **BTimeStamp** *pTime* = **BTimeStamp** (), **BString** *pType* = **BString** (), **BString** *pUser* = **BString** (), **BString** *pTitle* = **BString** (), **BString** *pDescription* = **BString** ())

## 9.11.3 Member Function Documentation

**9.11.3.1** **BString** **Bds::ChangeGroup::getType** ()

**9.11.3.2** **BError** **Bds::ChangeGroup::setMembers** (**BDictString** & *members*)

**9.11.3.3** **BError** **Bds::ChangeGroup::getMembers** (**BDictString** & *members*)

## 9.11.4 Member Data Documentation

**9.11.4.1** **BUInt32** **Bds::ChangeGroup::id**

The unique id.

**9.11.4.2** **BTimeStamp** **Bds::ChangeGroup::time**

The Time the change was made.

**9.11.4.3** **BString** **Bds::ChangeGroup::type**

The type of change.

**9.11.4.4** **BString** **Bds::ChangeGroup::user**

The user who made the change.

**9.11.4.5** **BString** **Bds::ChangeGroup::title**

The Changes title.

**9.11.4.6** **BString** **Bds::ChangeGroup::description**

The Description of the change.

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

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

## 9.12 Bds::Channel Class Reference

This class defines a [Channel](#).

```
#include <BdsD.h>
```

### Public Member Functions

- [Channel](#) (BUInt32 pid=BUInt32(), BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BString pnetwork=BString(), BString pstation=BString(), BString pchannel=BString(), BString pchannelType=BString(), BString pchannelAux=BString(), BString pdescription=BString())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*ID.*
- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time the channel was available.*
- BString [network](#)  
*The Network Name.*
- BString [station](#)  
*The Stations name. (Do we want this in here and if so do we want, network, source etc).*
- BString [channel](#)  
*The channels name.*
- BString [channelType](#)  
*The channels type.*
- BString [channelAux](#)  
*The channels auxiliary identifier.*
- BString [description](#)  
*The channels description.*

### 9.12.1 Detailed Description

This class defines a [Channel](#).

## 9.12.2 Constructor & Destructor Documentation

**9.12.2.1 Bds::Channel::Channel (BUInt32 *pid* = BUInt32(), BTimeStamp *pstartTime* = BTimeStamp(), BTimeStamp *pendTime* = BTimeStamp(), BString *pnetwork* = BString(), BString *pstation* = BString(), BString *pchannel* = BString(), BString *pchannelType* = BString(), BString *pchannelAux* = BString(), BString *pdescription* = BString())**

## 9.12.3 Member Function Documentation

**9.12.3.1 BString Bds::Channel::getType ()**

**9.12.3.2 BError Bds::Channel::setMembers (BDictString & *members*)**

**9.12.3.3 BError Bds::Channel::getMembers (BDictString & *members*)**

## 9.12.4 Member Data Documentation

**9.12.4.1 BUInt32 Bds::Channel::id**

ID.

**9.12.4.2 BTimeStamp Bds::Channel::startTime**

The Start Time.

**9.12.4.3 BTimeStamp Bds::Channel::endTime**

The End Time the channel was available.

**9.12.4.4 BString Bds::Channel::network**

The [Network](#) Name.

**9.12.4.5 BString Bds::Channel::station**

The Stations name. (Do we want this in here and if so do we want, network, source etc).

**9.12.4.6 BString Bds::Channel::channel**

The channels name.

**9.12.4.7 BString Bds::Channel::channelType**

The channels type.

**9.12.4.8 BString Bds::Channel::channelAux**

The channels auxiliary identifier.

**9.12.4.9 BString Bds::Channel::description**

The channels description.

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

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

## 9.13 Bds::ChannelInfo Class Reference

```
#include <BdsD.h>
```

### Public Member Functions

- [ChannelInfo](#) ([Station](#) pstation=[Station](#)(), [Location](#) plocation=[Location](#)(), [Channel](#) pchannel=[Channel](#)(), [Digitiser](#) pdigitiser=[Digitiser](#)(), [Sensor](#) psensor=[Sensor](#)(), [SensorLocation](#) psensorLocation=[SensorLocation](#)(), [Calibration](#) pcalibration=[Calibration](#)(), [BList](#)< [Response](#) > presponses=[BList](#)< [Response](#) >())

### Public Attributes

- [Station](#) station  
*The [Station](#) info.*
- [Location](#) location  
*The [Station](#) location.*
- [Channel](#) channel  
*The [Channel](#) data.*
- [Digitiser](#) digitiser  
*The [Digitiser](#) in use.*
- [Sensor](#) sensor  
*The [Sensor](#) in use.*
- [SensorLocation](#) sensorLocation  
*The [Sensor](#)'s location.*
- [Calibration](#) calibration  
*The [Calibration](#) info.*
- [BList](#)< [Response](#) > responses  
*The list of frequency responses.*

### 9.13.1 Constructor & Destructor Documentation

- 9.13.1.1** [Bds::ChannelInfo::ChannelInfo](#) ([Station](#) pstation = [Station](#) (), [Location](#) plocation = [Location](#) (), [Channel](#) pchannel = [Channel](#) (), [Digitiser](#) pdigitiser = [Digitiser](#) (), [Sensor](#) psensor = [Sensor](#) (), [SensorLocation](#) psensorLocation = [SensorLocation](#) (), [Calibration](#) pcalibration = [Calibration](#) (), [BList](#)< [Response](#) > presponses = [BList](#)<[Response](#) > ())

### 9.13.2 Member Data Documentation

#### 9.13.2.1 Station [Bds::ChannelInfo::station](#)

The [Station](#) info.

**9.13.2.2 Location Bds::ChannelInfo::location**

The [Station](#) location.

**9.13.2.3 Channel Bds::ChannelInfo::channel**

The [Channel](#) data.

**9.13.2.4 Digitiser Bds::ChannelInfo::digitiser**

The [Digitiser](#) in use.

**9.13.2.5 Sensor Bds::ChannelInfo::sensor**

The [Sensor](#) in use.

**9.13.2.6 SensorLocation Bds::ChannelInfo::sensorLocation**

The Sensor's location.

**9.13.2.7 Calibration Bds::ChannelInfo::calibration**

The [Calibration](#) info.

**9.13.2.8 BList<Response > Bds::ChannelInfo::responses**

The list of frequency responses.

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

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

## 9.14 Bds::ChannelInstrument Class Reference

```
#include <BdsD.h>
```

### Public Member Functions

- [ChannelInstrument](#) (BUInt32 pid=BUInt32(), BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BUInt32 pchannelId=BUInt32(), BUInt32 pdigitiserId=BUInt32(), BUInt32 psensorId=BUInt32())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*ID.*
- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time the channel was available.*
- BUInt32 [channelId](#)  
*The channels Id.*
- BUInt32 [digitiserId](#)  
*The Digitiser in use.*
- BUInt32 [sensorId](#)  
*The sensor in use.*

## 9.14.1 Constructor & Destructor Documentation

**9.14.1.1** **Bds::ChannelInstrument::ChannelInstrument** (**BUInt32** *pid* = **BUInt32** (), **BTimeStamp** *pstartTime* = **BTimeStamp** (), **BTimeStamp** *pendTime* = **BTimeStamp** (), **BUInt32** *pchannelId* = **BUInt32** (), **BUInt32** *pdigitiserId* = **BUInt32** (), **BUInt32** *psensorId* = **BUInt32** ())

## 9.14.2 Member Function Documentation

**9.14.2.1** **BString** **Bds::ChannelInstrument::getType** ()

**9.14.2.2** **BError** **Bds::ChannelInstrument::setMembers** (**BDictString** & *members*)

**9.14.2.3** **BError** **Bds::ChannelInstrument::getMembers** (**BDictString** & *members*)

## 9.14.3 Member Data Documentation

**9.14.3.1** **BUInt32** **Bds::ChannelInstrument::id**

ID.

**9.14.3.2** **BTimeStamp** **Bds::ChannelInstrument::startTime**

The Start Time.

**9.14.3.3** **BTimeStamp** **Bds::ChannelInstrument::endTime**

The End Time the channel was available.

**9.14.3.4** **BUInt32** **Bds::ChannelInstrument::channelId**

The channels Id.

**9.14.3.5** **BUInt32** **Bds::ChannelInstrument::digitiserId**

The [Digitiser](#) in use.

**9.14.3.6** **BUInt32** **Bds::ChannelInstrument::sensorId**

The sensor in use.

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

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

## 9.15 Bds::DataAccess Class Reference

```
#include <BdsC.h>
```

### Public Member Functions

- [DataAccess](#) (BString name="")
- BError [connect](#) (BString user, BString password)  
*Provides user/password information for secure connection.*
- BError [setUser](#) (BString user, BString email)  
*Sets user to given name or email.*
- BError [setUserReal](#) ()  
*Sets user back to real user.*
- BError [getVersion](#) (BString &version)  
*Gets the software version.*
- BError [userGetGroups](#) (BList< BString > &groups)
- BError [networkGetList](#) (BList< [Network](#) > &networks)  
*Get list of groups the user belongs to.*
- BError [stationGetList](#) ([Selection](#) sel, BList< [Station](#) > &stations)  
*Get list of Stations.*
- BError [getSelectionInfo](#) ([SelectionGroup](#) group, [SelectionInfo](#) &selectionInfo)  
*Get selection info.*
- BError [getSelections](#) ([SelectionGroup](#) group, [Selection](#) selectionIn, [Selection](#) &selectionOut)  
*Get selection list.*
- BError [dataSearch](#) ([Selection](#) selection, [DataInfo](#) &dataInfo)  
*Search for data matching the given selection parameters.*
- BError [dataGetChannelInfo](#) ([DataInfo](#) dataInfo, BList< [ChannelInfo](#) > &channelInfos)  
*Return the channel MetaData in structured form.*
- BError [dataOpen](#) ([DataInfo](#) dataInfo, BString mode, BString format, [DataHandle](#) &dataHandle)  
*Open a data stream.*
- BError [dataGetInfo](#) ([DataHandle](#) dataHandle, [DataInfo](#) &dataInfo)  
*Get information on the data stream.*
- BError [dataGetWarnings](#) ([DataHandle](#) dataHandle, BList< BString > &warnings)  
*Get information on the data stream.*
- BError [dataSeekBlock](#) ([DataHandle](#) dataHandle, BUInt32 channel, BUInt32 segment, BTimeStamp time, BUInt32 &blockNumber)

*Searches for a data block matching the time given.*

- BError [dataGetBlock](#) ([DataHandle](#) dataHandle, BUInt32 channel, BUInt32 segment, BUInt32 block-Number, [DataBlock](#) &data)

*Return a block of data.*

- BError [dataClose](#) ([DataHandle](#) dataHandle)

*Close a stream.*

- BError [dataFormattedRead](#) ([DataHandle](#) dataHandle, BUInt32 number, BArray< BUInt8 > &data)

*Read the raw data from the stream.*

- BError [dataFormattedGetLength](#) ([DataHandle](#) dataHandle, BUInt64 &length)

*Read the raw data from the stream.*

- BError [noteUpdate](#) (BInt32 append, [Note](#) note, BUInt32 &id)

### 9.15.1 Detailed Description

This class defines a [TimePeriod](#) This class defines a [Network](#) organisation. Could have other information here like website, contacts etc This class defines an arrays channel This class defines a [Location](#). This will be the stations location. This class defines a Pole/Zero [Response](#) This class defines a Amplitude/Phase [Response](#) This class defines a [Response](#) This class defines a calibration setting This class defines a [Digitiser](#) This class defines a [Sensor](#) This class defines a [Sensor](#) location This class defines a [Channel](#) This class defines the set of data selection criteria This class defines information on a single channels set of data This defines a handle to a data stream/file This holds information on a user This holds information on a groups This holds information on data access groups This holds information on a change This holds information on a change This holds information on a [Note](#) This is the Data Access API interface

### 9.15.2 Constructor & Destructor Documentation

#### 9.15.2.1 Bds::DataAccess::DataAccess (BString *name* = "")

### 9.15.3 Member Function Documentation

#### 9.15.3.1 BError Bds::DataAccess::connect (BString *user*, BString *password*)

Provides user/password information for secure connection.

#### 9.15.3.2 BError Bds::DataAccess::setUser (BString *user*, BString *email*)

Sets user to given name or email.

#### 9.15.3.3 BError Bds::DataAccess::setUserReal ()

Sets user back to real user.

**9.15.3.4 BError Bds::DataAccess::getVersion (BString & *version*)**

Gets the software version.

**9.15.3.5 BError Bds::DataAccess::userGetGroups (BList< BString > & *groups*)****9.15.3.6 BError Bds::DataAccess::networkGetList (BList< Network > & *networks*)**

Get list of groups the user belongs to.

Get list of Networks

**9.15.3.7 BError Bds::DataAccess::stationGetList (Selection *sel*, BList< Station > & *stations*)**

Get list of Stations.

**9.15.3.8 BError Bds::DataAccess::getSelectionInfo (SelectionGroup *group*, SelectionInfo & *selectionInfo*)**

Get selection info.

**9.15.3.9 BError Bds::DataAccess::getSelections (SelectionGroup *group*, Selection *selectionIn*, Selection & *selectionOut*)**

Get selection list.

**9.15.3.10 BError Bds::DataAccess::dataSearch (Selection *selection*, DataInfo & *dataInfo*)**

Search for data matching the given selection parameters.

**9.15.3.11 BError Bds::DataAccess::dataGetChannelInfo (DataInfo *dataInfo*, BList< ChannelInfo > & *channelInfos*)**

Return the channel MetaData in structured form.

**9.15.3.12 BError Bds::DataAccess::dataOpen (DataInfo *dataInfo*, BString *mode*, BString *format*, DataHandle & *dataHandle*)**

Open a data stream.

**9.15.3.13 BError Bds::DataAccess::dataGetInfo (DataHandle *dataHandle*, DataInfo & *dataInfo*)**

Get information on the data stream.

**9.15.3.14 BError Bds::DataAccess::dataGetWarnings (DataHandle *dataHandle*, BList< BString > & *warnings*)**

Get information on the data stream.

**9.15.3.15 BError Bds::DataAccess::dataSeekBlock (DataHandle *dataHandle*, BUInt32 *channel*, BUInt32 *segment*, BTimeStamp *time*, BUInt32 & *blockNumber*)**

Searches for a data block matching the time given.

**9.15.3.16 BError Bds::DataAccess::dataGetBlock (DataHandle *dataHandle*, BUInt32 *channel*, BUInt32 *segment*, BUInt32 *blockNumber*, DataBlock & *data*)**

Return a block of data.

**9.15.3.17 BError Bds::DataAccess::dataClose (DataHandle *dataHandle*)**

Close a stream.

**9.15.3.18 BError Bds::DataAccess::dataFormattedRead (DataHandle *dataHandle*, BUInt32 *number*, BArray< BUInt8 > & *data*)**

Read the raw data from the stream.

**9.15.3.19 BError Bds::DataAccess::dataFormattedGetLength (DataHandle *dataHandle*, BUInt64 & *length*)**

Read the raw data from the stream.

**9.15.3.20 BError Bds::DataAccess::noteUpdate (BInt32 *append*, Note *note*, BUInt32 & *id*)**

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

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

## 9.16 Bds::DataAddAccess Class Reference

This is the DataAdd Access API interface.

```
#include <BdsC.h>
```

### Public Member Functions

- [DataAddAccess](#) (BString name="")
- BError [connect](#) (BString user, BString password)  
*Provides user/password information.*
- BError [setUser](#) (BString user, BString email)  
*Sets user to given name or email.*
- BError [setUserReal](#) ()  
*Sets user back to real user.*
- BError [getVersion](#) (BString &version)  
*Gets the software version.*
- BError [userGetGroups](#) (BList< BString > &groups)
- BError [networkGetList](#) (BList< [Network](#) > &networks)  
*Get list of groups the user belongs to.*
- BError [stationGetList](#) ([Selection](#) sel, BList< [Station](#) > &stations)  
*Get list of Stations.*
- BError [getSelectionInfo](#) ([SelectionGroup](#) group, [SelectionInfo](#) &selectionInfo)  
*Get selection info.*
- BError [getSelections](#) ([SelectionGroup](#) group, [Selection](#) selectionIn, [Selection](#) &selectionOut)  
*Get selection list.*
- BError [dataSearch](#) ([Selection](#) selection, [DataInfo](#) &dataInfo)  
*Search for data matching the given selection parameters.*
- BError [dataGetChannelInfo](#) ([DataInfo](#) dataInfo, BList< [ChannelInfo](#) > &channelInfos)  
*Return the channel MetaData in structured form.*
- BError [dataOpen](#) ([DataInfo](#) dataInfo, BString mode, BString format, [DataHandle](#) &dataHandle)  
*Open a data stream.*
- BError [dataGetInfo](#) ([DataHandle](#) dataHandle, [DataInfo](#) &dataInfo)  
*Get information on the data stream.*
- BError [dataGetWarnings](#) ([DataHandle](#) dataHandle, BList< BString > &warnings)  
*Get information on the data stream.*

- BError [dataSeekBlock](#) ([DataHandle](#) dataHandle, BUInt32 channel, BUInt32 segment, BTimeStamp time, BUInt32 &blockNumber)  
*Searches for a data block matching the time given.*
- BError [dataGetBlock](#) ([DataHandle](#) dataHandle, BUInt32 channel, BUInt32 segment, BUInt32 blockNumber, [DataBlock](#) &data)  
*Return a block of data.*
- BError [dataSetInfo](#) ([DataHandle](#) dataHandle, [DataInfo](#) dataInfo)  
*Set the info when writing to a stream.*
- BError [dataPutBlock](#) ([DataHandle](#) dataHandle, [DataBlock](#) data)  
*Send a block of data.*
- BError [dataClose](#) ([DataHandle](#) dataHandle)  
*Close a stream.*
- BError [dataFormattedRead](#) ([DataHandle](#) dataHandle, BUInt32 number, BArray< BUInt8 > &data)  
*Read the raw data from the stream.*
- BError [dataFormattedGetLength](#) ([DataHandle](#) dataHandle, BUInt64 &length)  
*The total length in bytes of the formatted data.*
- BError [noteUpdate](#) (BInt32 append, [Note](#) note, BUInt32 &id)

### 9.16.1 Detailed Description

This is the DataAdd Access API interface.

### 9.16.2 Constructor & Destructor Documentation

#### 9.16.2.1 Bds::DataAddAccess::DataAddAccess (BString *name* = "")

### 9.16.3 Member Function Documentation

#### 9.16.3.1 BError Bds::DataAddAccess::connect (BString *user*, BString *password*)

Provides user/password information.

#### 9.16.3.2 BError Bds::DataAddAccess::setUser (BString *user*, BString *email*)

Sets user to given name or email.

#### 9.16.3.3 BError Bds::DataAddAccess::setUserReal ()

Sets user back to real user.

**9.16.3.4 BError Bds::DataAddAccess::getVersion (BString & *version*)**

Gets the software version.

**9.16.3.5 BError Bds::DataAddAccess::userGetGroups (BList< BString > & *groups*)****9.16.3.6 BError Bds::DataAddAccess::networkGetList (BList< Network > & *networks*)**

Get list of groups the user belongs to.

Get list of Networks

**9.16.3.7 BError Bds::DataAddAccess::stationGetList (Selection *sel*, BList< Station > & *stations*)**

Get list of Stations.

**9.16.3.8 BError Bds::DataAddAccess::getSelectionInfo (SelectionGroup *group*, SelectionInfo & *selectionInfo*)**

Get selection info.

**9.16.3.9 BError Bds::DataAddAccess::getSelections (SelectionGroup *group*, Selection *selectionIn*, Selection & *selectionOut*)**

Get selection list.

**9.16.3.10 BError Bds::DataAddAccess::dataSearch (Selection *selection*, DataInfo & *dataInfo*)**

Search for data matching the given selection parameters.

**9.16.3.11 BError Bds::DataAddAccess::dataGetChannelInfo (DataInfo *dataInfo*, BList< ChannelInfo > & *channelInfos*)**

Return the channel MetaData in structured form.

**9.16.3.12 BError Bds::DataAddAccess::dataOpen (DataInfo *dataInfo*, BString *mode*, BString *format*, DataHandle & *dataHandle*)**

Open a data stream.

**9.16.3.13 BError Bds::DataAddAccess::dataGetInfo (DataHandle *dataHandle*, DataInfo & *dataInfo*)**

Get information on the data stream.

**9.16.3.14 BError Bds::DataAddAccess::dataGetWarnings (DataHandle *dataHandle*, BList< BString > & *warnings*)**

Get information on the data stream.

**9.16.3.15 BError Bds::DataAddAccess::dataSeekBlock (DataHandle *dataHandle*, BUInt32 *channel*, BUInt32 *segment*, BTimeStamp *time*, BUInt32 & *blockNumber*)**

Searches for a data block matching the time given.

**9.16.3.16 BError Bds::DataAddAccess::dataGetBlock (DataHandle *dataHandle*, BUInt32 *channel*, BUInt32 *segment*, BUInt32 *blockNumber*, DataBlock & *data*)**

Return a block of data.

**9.16.3.17 BError Bds::DataAddAccess::dataSetInfo (DataHandle *dataHandle*, DataInfo *dataInfo*)**

Set the info when writing to a stream.

**9.16.3.18 BError Bds::DataAddAccess::dataPutBlock (DataHandle *dataHandle*, DataBlock *data*)**

Send a block of data.

**9.16.3.19 BError Bds::DataAddAccess::dataClose (DataHandle *dataHandle*)**

Close a stream.

**9.16.3.20 BError Bds::DataAddAccess::dataFormattedRead (DataHandle *dataHandle*, BUInt32 *number*, BArray< BUInt8 > & *data*)**

Read the raw data from the stream.

**9.16.3.21 BError Bds::DataAddAccess::dataFormattedGetLength (DataHandle *dataHandle*, BUInt64 & *length*)**

The total length in bytes of the formatted data.

**9.16.3.22 BError Bds::DataAddAccess::noteUpdate (BInt32 *append*, Note *note*, BUInt32 & *id*)**

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

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

## 9.17 Bds::DataBlock Class Reference

```
#include <BdsD.h>
```

### Public Member Functions

- **DataBlock** (BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BUInt32 pchannelNumber=BUInt32(), BUInt32 psegmentNumber=BUInt32(), BArray< BArray< BFloat64 > > pchannelData=BArray< BArray< BFloat64 > >(), BDict< BString > pinfo=BDict< BString >())

### Public Attributes

- BTimeStamp **startTime**  
*The Start Time.*
- BTimeStamp **endTime**  
*The End Time the channel was available.*
- BUInt32 **channelNumber**  
*The first channel number.*
- BUInt32 **segmentNumber**  
*The segment number.*
- BArray< BArray< BFloat64 > > **channelData**  
*The raw channel data in a 2 dimensional array, orders as per channel information in dataInfo.*
- BDict< BString > **info**  
*Extra information on data.*

### 9.17.1 Constructor & Destructor Documentation

- 9.17.1.1 Bds::DataBlock::DataBlock (BTimeStamp pstartTime = BTimeStamp(), BTimeStamp pendTime = BTimeStamp(), BUInt32 pchannelNumber = BUInt32(), BUInt32 psegmentNumber = BUInt32(), BArray< BArray< BFloat64 > > pchannelData = BArray< BArray< BFloat64 > >(), BDict< BString > pinfo = BDict< BString >())**

### 9.17.2 Member Data Documentation

#### 9.17.2.1 BTimeStamp Bds::DataBlock::startTime

The Start Time.

#### 9.17.2.2 BTimeStamp Bds::DataBlock::endTime

The End Time the channel was available.

### 9.17.2.3 BUInt32 Bds::DataBlock::channelNumber

The first channel number.

### 9.17.2.4 BUInt32 Bds::DataBlock::segmentNumber

The segment number.

### 9.17.2.5 BArray<BArray<BFloat64 > > Bds::DataBlock::channelData

The raw channel data in a 2 dimensional array, orders as per channel information in dataInfo.

### 9.17.2.6 BDict<BString > Bds::DataBlock::info

Extra information on data.

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

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

## 9.18 Bds::DataChannel Class Reference

This class defines information on a single channels set of data.

```
#include <BdsD.h>
```

### Public Member Functions

- **DataChannel** (BUInt32 pid=BUInt32(), BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BString pnetwork=BString(), BString pstation=BString(), BString pchannel=BString(), BString psource=BString(), BUInt32 pnumBlocks=BUInt32(), BUInt64 pnumSamples=BUInt64(), BFloat64 psampleRate=BFloat64(), BUInt32 psampleFormat=BUInt32(), BUInt32 pdataFileId=BUInt32(), BUInt32 pdataFileChannel=BUInt32(), BDict< BString > pinfo=BDict< BString >())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*ID.*
- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time.*
- BString [network](#)  
*The [Network](#) Name.*
- BString [station](#)  
*The [Station](#) name.*
- BString [channel](#)  
*The [Channels](#) name.*
- BString [source](#)  
*The [Data Source](#).*
- BUInt32 [numBlocks](#)  
*The total number of blocks per channel if known, 0 otherwise.*
- BUInt64 [numSamples](#)  
*The total number of samples per channel if known, 0 otherwise.*
- BFloat64 [sampleRate](#)  
*The data's sample rate.*

- BUInt32 [sampleFormat](#)  
*The data sample format.*
- BUInt32 [dataFileId](#)  
*The Data File Id.*
- BUInt32 [dataFileChannel](#)  
*The Data File [Channel](#) number.*
- BDict< BString > [info](#)  
*Extra info on the channel.*

### 9.18.1 Detailed Description

This class defines information on a single channels set of data.

### 9.18.2 Constructor & Destructor Documentation

**9.18.2.1 Bds::DataChannel::DataChannel (BUInt32 *pid* = BUInt32(), BTimeStamp *pstartTime* = BTimeStamp(), BTimeStamp *pendTime* = BTimeStamp(), BString *pnetwork* = BString(), BString *pstation* = BString(), BString *pchannel* = BString(), BString *psource* = BString(), BUInt32 *pnumBlocks* = BUInt32(), BUInt64 *pnumSamples* = BUInt64(), BFloat64 *psampleRate* = BFloat64(), BUInt32 *psampleFormat* = BUInt32(), BUInt32 *pdataFileId* = BUInt32(), BUInt32 *pdataFileChannel* = BUInt32(), BDict< BString > *pinfo* = BDict<BString >())**

### 9.18.3 Member Function Documentation

**9.18.3.1 BString Bds::DataChannel::getType ()**

**9.18.3.2 BError Bds::DataChannel::setMembers (BDictString & *members*)**

**9.18.3.3 BError Bds::DataChannel::getMembers (BDictString & *members*)**

### 9.18.4 Member Data Documentation

**9.18.4.1 BUInt32 Bds::DataChannel::id**

ID.

**9.18.4.2 BTimeStamp Bds::DataChannel::startTime**

The Start Time.

**9.18.4.3 BTimeStamp Bds::DataChannel::endTime**

The End Time.

#### 9.18.4.4 BString Bds::DataChannel::network

The [Network](#) Name.

#### 9.18.4.5 BString Bds::DataChannel::station

The [Station](#) name.

#### 9.18.4.6 BString Bds::DataChannel::channel

The Channels name.

#### 9.18.4.7 BString Bds::DataChannel::source

The Data Source.

#### 9.18.4.8 BUInt32 Bds::DataChannel::numBlocks

The total number of blocks per channel if known, 0 otherwise.

#### 9.18.4.9 BUInt64 Bds::DataChannel::numSamples

The total number of samples per channel if known, 0 otherwise.

#### 9.18.4.10 BFloat64 Bds::DataChannel::sampleRate

The data's sample rate.

#### 9.18.4.11 BUInt32 Bds::DataChannel::sampleFormat

The data sample format.

#### 9.18.4.12 BUInt32 Bds::DataChannel::dataFileId

The Data File Id.

#### 9.18.4.13 BUInt32 Bds::DataChannel::dataFileChannel

The Data File [Channel](#) number.

#### 9.18.4.14 BDict<BString > Bds::DataChannel::info

Extra info on the channel.

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

- [BdsD.h](#)

- [BdsD.cc](#)

## 9.19 Bds::DataCollate Class Reference

This class defines the interface for generic data file access.

```
#include <BdsDataCollate.h>
```

### Public Member Functions

- [DataCollate \(\)](#)
- [~DataCollate \(\)](#)
- BError [addSource \(DataFile &dataFile, BUInt channel\)](#)
- BError [readData \(BUInt32 blockNumber, DataBlock &data\)](#)

### 9.19.1 Detailed Description

This class defines the interface for generic data file access.

### 9.19.2 Constructor & Destructor Documentation

#### 9.19.2.1 Bds::DataCollate::DataCollate ()

#### 9.19.2.2 Bds::DataCollate::~~DataCollate ()

### 9.19.3 Member Function Documentation

#### 9.19.3.1 BError Bds::DataCollate::addSource (DataFile & *dataFile*, BUInt *channel*)

#### 9.19.3.2 BError Bds::DataCollate::readData (BUInt32 *blockNumber*, DataBlock & *data*)

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

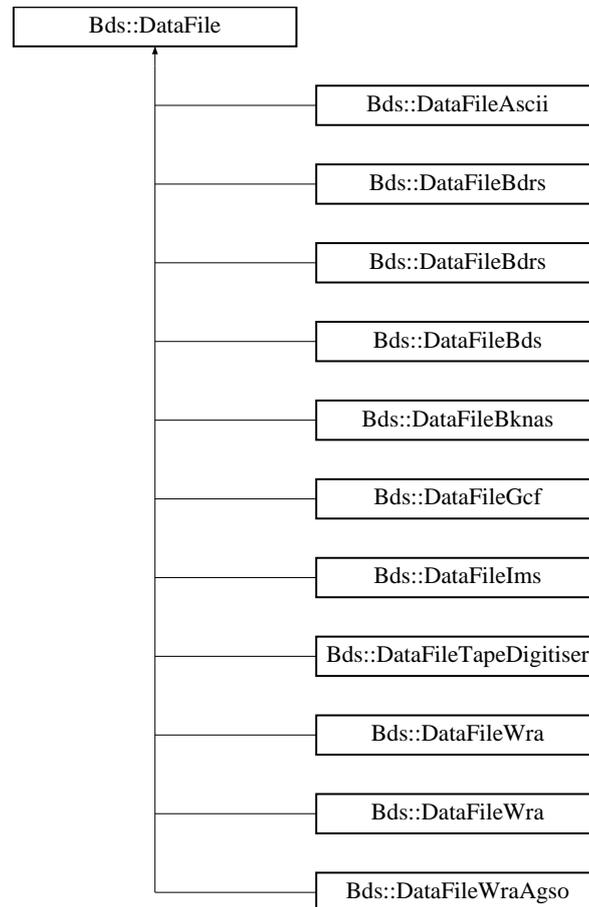
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp](#)

## 9.20 Bds::DataFile Class Reference

This class defines the interface for generic data file access.

```
#include <BdsDataFile.h>
```

Inheritance diagram for Bds::DataFile::



### Public Types

- enum [DataOrder](#) { [DataOrderUnknown](#), [DataOrderAll](#), [DataOrderSample](#), [DataOrderChannel](#) }
- enum [Features](#) { [FeatureNone](#) = 0x00, [FeatureCanWrite](#) = 0x01, [FeatureCanRead](#) = 0x02 }
- enum [OptionsList](#) { [OptionNone](#) = 0x00, [OptionValidate](#) = 0x01, [OptionFileNameProcess](#) = 0x02, [OptionValidateCorruptions](#) = 0x04 }
- typedef int [Options](#)

### Public Member Functions

- [DataFile](#) ()
  - virtual [~DataFile](#) ()
  - virtual void [init](#) ()
- Initialise.*

- virtual BError [open](#) (BString fileName, BString mode)  
*Open the file for read or write.*
- virtual void [close](#) ()  
*Close the file.*
- virtual BError [setFormat](#) (BString format)  
*Set the sub-format.*
- virtual [DataOrder](#) [getDataOrder](#) ()  
*Get the expected order of writing data, by sample or by channel.*
- virtual int [getFeatures](#) ()  
*Get bitmask of supported features.*
- virtual BError [setInfo](#) ([DataInfo](#) &dataInfo, BList< [ChannelInfo](#) > &channelInfos)  
*Set information on data for write.*
- virtual BError [start](#) (BUInt channel, BUInt segment)  
*Start next segment of data.*
- virtual BError [writeData](#) ([DataBlock](#) &data)  
*Write a block of data.*
- virtual BError [end](#) ()  
*End segment.*
- virtual BError [fileNameProcess](#) ()  
*Parse the file name for a date/time.*
- virtual BError [getFormat](#) (BString &format)  
*Get sub-format.*
- virtual BError [getInfo](#) ([DataInfo](#) &dataInfo, [Options](#) options, BList< BError > &errors)  
*Get info on data.*
- virtual BError [seekBlock](#) (BUInt32 channel, BTimeStamp time, BUInt32 &blockNumber)  
*Find requested block on given channel given a time.*
- virtual BError [readData](#) (BUInt32 channel, BUInt32 blockNumber, [DataBlock](#) &data)  
*Read a block.*

## Static Public Member Functions

- static BStringList [getFormats](#) ()  
*Get list of supported formats.*

## Protected Attributes

- BString [ofileName](#)
- BString [omode](#)
- BTimeStamp [ofilenameTime](#)
- BFile [ofile](#)
- BString [oformat](#)

### 9.20.1 Detailed Description

This class defines the interface for generic data file access.

### 9.20.2 Member Typedef Documentation

#### 9.20.2.1 typedef int Bds::DataFile::Options

### 9.20.3 Member Enumeration Documentation

#### 9.20.3.1 enum Bds::DataFile::DataOrder

Enumerator:

*DataOrderUnknown*  
*DataOrderAll*  
*DataOrderSample*  
*DataOrderChannel*

#### 9.20.3.2 enum Bds::DataFile::Features

Enumerator:

*FeatureNone*  
*FeatureCanWrite*  
*FeatureCanRead*

#### 9.20.3.3 enum Bds::DataFile::OptionsList

Enumerator:

*OptionNone*  
*OptionValidate*  
*OptionFileNameProcess*  
*OptionValidateCorruptions*

## 9.20.4 Constructor & Destructor Documentation

### 9.20.4.1 `Bds::DataFile::DataFile ()`

### 9.20.4.2 `Bds::DataFile::~~DataFile ()` [virtual]

## 9.20.5 Member Function Documentation

### 9.20.5.1 `void Bds::DataFile::init ()` [virtual]

Initialise.

### 9.20.5.2 `BError Bds::DataFile::open (BString fileName, BString mode)` [virtual]

Open the file for read or write.

Reimplemented in [Bds::DataFileAscii](#), [Bds::DataFileBdrs](#), [Bds::DataFileBds](#), [Bds::DataFileBknas](#), [Bds::DataFileIms](#), [Bds::DataFileTapeDigitiser](#), and [Bds::DataFileWra](#).

### 9.20.5.3 `void Bds::DataFile::close ()` [virtual]

Close the file.

Reimplemented in [Bds::DataFileBds](#), and [Bds::DataFileIms](#).

### 9.20.5.4 `BError Bds::DataFile::setFormat (BString format)` [virtual]

Set the sub-format.

Reimplemented in [Bds::DataFileBds](#), [Bds::DataFileWra](#), and [Bds::DataFileWra](#).

### 9.20.5.5 `DataFile::DataOrder Bds::DataFile::getDataOrder ()` [virtual]

Get the expected order of writing data, by sample or by channel.

Reimplemented in [Bds::DataFileBdrs](#), [Bds::DataFileBdrs](#), [Bds::DataFileBds](#), [Bds::DataFileIms](#), [Bds::DataFileWra](#), [Bds::DataFileWra](#), and [Bds::DataFileWraAgso](#).

### 9.20.5.6 `int Bds::DataFile::getFeatures ()` [virtual]

Get bitmask of supported features.

Reimplemented in [Bds::DataFileBdrs](#), [Bds::DataFileBdrs](#), [Bds::DataFileGcf](#), [Bds::DataFileIms](#), [Bds::DataFileWra](#), [Bds::DataFileWra](#), and [Bds::DataFileWraAgso](#).

### 9.20.5.7 `BError Bds::DataFile::setInfo (DataInfo & dataInfo, BList< ChannelInfo > & channelInfos)` [virtual]

Set information on data for write.

Reimplemented in [Bds::DataFileAscii](#), [Bds::DataFileBds](#), [Bds::DataFileBknas](#), and [Bds::DataFileIms](#).

**9.20.5.8 BError Bds::DataFile::start (BUInt *channel*, BUInt *segment*)** [virtual]

Start next segment of data.

Reimplemented in [Bds::DataFileImgs](#).

**9.20.5.9 BError Bds::DataFile::writeData (DataBlock & *data*)** [virtual]

Write a block of data.

Reimplemented in [Bds::DataFileAscii](#), [Bds::DataFileBds](#), [Bds::DataFileBknas](#), and [Bds::DataFileImgs](#).

**9.20.5.10 BError Bds::DataFile::end ()** [virtual]

End segment.

Reimplemented in [Bds::DataFileImgs](#).

**9.20.5.11 BError Bds::DataFile::fileNameProcess ()** [virtual]

Parse the file name for a date/time.

**9.20.5.12 BError Bds::DataFile::getFormat (BString & *format*)** [virtual]

Get sub-format.

**9.20.5.13 BError Bds::DataFile::getInfo (DataInfo & *dataInfo*, Options *options*, BList< BError > & *errors*)** [virtual]

Get info on data.

Reimplemented in [Bds::DataFileBdrs](#).

**9.20.5.14 BError Bds::DataFile::seekBlock (BUInt32 *channel*, BTimeStamp *time*, BUInt32 & *blockNumber*)** [virtual]

Find requested block on given channel given a time.

Reimplemented in [Bds::DataFileBds](#).

**9.20.5.15 BError Bds::DataFile::readData (BUInt32 *channel*, BUInt32 *blockNumber*, DataBlock & *data*)** [virtual]

Read a block.

Reimplemented in [Bds::DataFileBdrs](#), [Bds::DataFileBdrs](#), [Bds::DataFileBds](#), [Bds::DataFileGcf](#), [Bds::DataFileTapeDigitiser](#), [Bds::DataFileWra](#), [Bds::DataFileWra](#), and [Bds::DataFileWraAgso](#).

**9.20.5.16 BStringList Bds::DataFile::getFormats ()** [static]

Get list of supported formats.

Reimplemented in [Bds::DataFileAscii](#), [Bds::DataFileBdrs](#), [Bds::DataFileBdrs](#), [Bds::DataFileBds](#), [Bds::DataFileBknas](#), [Bds::DataFileGcf](#), [Bds::DataFileIms](#), [Bds::DataFileTapeDigitiser](#), [Bds::DataFileWra](#), [Bds::DataFileWra](#), and [Bds::DataFileWraAgso](#).

## 9.20.6 Member Data Documentation

**9.20.6.1** `BString Bds::DataFile::ofilename` [protected]

**9.20.6.2** `BString Bds::DataFile::omode` [protected]

**9.20.6.3** `BTimeStamp Bds::DataFile::ofilenameTime` [protected]

**9.20.6.4** `BFile Bds::DataFile::ofile` [protected]

**9.20.6.5** `BString Bds::DataFile::oformat` [protected]

Reimplemented in [Bds::DataFileBds](#), and [Bds::DataFileGcf](#).

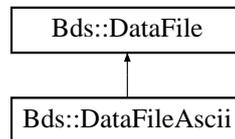
The documentation for this class was generated from the following files:

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp](#)

## 9.21 Bds::DataFileAscii Class Reference

```
#include <BdsDataFileAscii.h>
```

Inheritance diagram for Bds::DataFileAscii:



### Public Member Functions

- [DataFileAscii](#) ()
- BError [open](#) (BString fileName, BString mode)  
*Open the file for read or write.*
- BError [setInfo](#) ([DataInfo](#) &dataInfo, BList< [ChannelInfo](#) > &channelInfos)  
*Set information on data for write.*
- BError [writeData](#) ([DataBlock](#) &data)  
*Write a block of data.*

### Static Public Member Functions

- static BStringList [getFormats](#) ()  
*Get list of supported formats.*

### Private Attributes

- [DataInfo](#) odataInfo

### 9.21.1 Constructor & Destructor Documentation

#### 9.21.1.1 Bds::DataFileAscii::DataFileAscii ()

### 9.21.2 Member Function Documentation

#### 9.21.2.1 BError Bds::DataFileAscii::open (BString *fileName*, BString *mode*) [virtual]

Open the file for read or write.

Reimplemented from [Bds::DataFile](#).

### 9.21.2.2 BError Bds::DataFileAscii::setInfo (DataInfo & *dataInfo*, BList< ChannelInfo > & *channelInfos*) [virtual]

Set information on data for write.

Reimplemented from [Bds::DataFile](#).

### 9.21.2.3 BError Bds::DataFileAscii::writeData (DataBlock & *data*) [virtual]

Write a block of data.

Reimplemented from [Bds::DataFile](#).

### 9.21.2.4 BStringList Bds::DataFileAscii::getFormats () [static]

Get list of supported formats.

Reimplemented from [Bds::DataFile](#).

## 9.21.3 Member Data Documentation

### 9.21.3.1 DataInfo Bds::DataFileAscii::odataInfo [private]

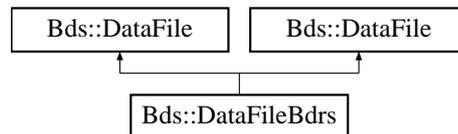
The documentation for this class was generated from the following files:

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp](#)

## 9.22 Bds::DataFileBdrs Class Reference

```
#include <BdsDataFileBdrs.h>
```

Inheritance diagram for Bds::DataFileBdrs::



### Public Member Functions

- [DataFileBdrs](#) ()
- [BError open](#) (BString fileName, BString mode)  
*Open the file for read or write.*
- [int getFeatures](#) ()  
*Get bitmask of supported features.*
- [DataOrder getDataOrder](#) ()  
*Get the expected order of writing data, by sample or by channel.*
- [BError getInfo](#) ([DataInfo](#) &dataInfo, int validate, BList< [BError](#) > &errors)  
*Get info on data.*
- [BError readData](#) (BUInt32 channel, BUInt32 blockNumber, [DataBlock](#) &data)  
*Read a block.*
- [DataFileBdrs](#) ()
- [int getFeatures](#) ()  
*Get bitmask of supported features.*
- [DataOrder getDataOrder](#) ()  
*Get the expected order of writing data, by sample or by channel.*
- [BError getInfo](#) ([DataInfo](#) &dataInfo, [Options](#) options, BList< [BError](#) > &errors)
- [BError readData](#) (BUInt32 channel, BUInt32 blockNumber, [DataBlock](#) &data)  
*Read a block.*

### Static Public Member Functions

- [static BStringList getFormats](#) ()  
*Get list of supported formats.*
- [static BStringList getFormats](#) ()  
*Get list of supported formats.*

## Private Member Functions

- BError [readBlock](#) (BUInt32 channel, BUInt64 pos, [DataBlock](#) &data)

## Private Attributes

- int [oyear](#)
- BTimeStamp [ofileNameDate](#)
- BUInt32 [oblockSize](#)
- double [osampleRate](#)
- [DataInfo](#) [odataInfo](#)
- int [omagic](#)
- int [oblockYear](#)
- BArray< BUInt64 > [oblockPositions](#)

## 9.22.1 Constructor & Destructor Documentation

**9.22.1.1** [Bds::DataFileBdrs::DataFileBdrs \(\)](#)

**9.22.1.2** [Bds::DataFileBdrs::DataFileBdrs \(\)](#)

## 9.22.2 Member Function Documentation

**9.22.2.1** [BError Bds::DataFileBdrs::open \(BString \*fileName\*, BString \*mode\*\)](#) [virtual]

Open the file for read or write.

Reimplemented from [Bds::DataFile](#).

**9.22.2.2** [int Bds::DataFileBdrs::getFeatures \(\)](#) [virtual]

Get bitmask of supported features.

Reimplemented from [Bds::DataFile](#).

**9.22.2.3** [DataFile::DataOrder Bds::DataFileBdrs::getDataOrder \(\)](#) [virtual]

Get the expected order of writing data, by sample or by channel.

Reimplemented from [Bds::DataFile](#).

**9.22.2.4** [BError Bds::DataFileBdrs::getInfo \(DataInfo & \*dataInfo\*, int \*options\*, BList< BError > & \*errors\*\)](#) [virtual]

Get info on data.

Reimplemented from [Bds::DataFile](#).

### 9.22.2.5 BError Bds::DataFileBdrs::readData (BUInt32 *channel*, BUInt32 *blockNumber*, DataBlock & *data*) [virtual]

Read a block.

Reimplemented from [Bds::DataFile](#).

### 9.22.2.6 BStringList Bds::DataFileBdrs::getFormats () [static]

Get list of supported formats.

Reimplemented from [Bds::DataFile](#).

### 9.22.2.7 int Bds::DataFileBdrs::getFeatures () [virtual]

Get bitmask of supported features.

Reimplemented from [Bds::DataFile](#).

### 9.22.2.8 DataOrder Bds::DataFileBdrs::getDataOrder () [virtual]

Get the expected order of writing data, by sample or by channel.

Reimplemented from [Bds::DataFile](#).

### 9.22.2.9 BError Bds::DataFileBdrs::getInfo (DataInfo & *dataInfo*, Options *options*, BList< BError > & *errors*)

### 9.22.2.10 BError Bds::DataFileBdrs::readData (BUInt32 *channel*, BUInt32 *blockNumber*, DataBlock & *data*) [virtual]

Read a block.

Reimplemented from [Bds::DataFile](#).

### 9.22.2.11 static BStringList Bds::DataFileBdrs::getFormats () [static]

Get list of supported formats.

Reimplemented from [Bds::DataFile](#).

**9.22.2.12 BError Bds::DataFileBdrs::readBlock (BUInt32 *channel*, BUInt64 *pos*, DataBlock & *data*)** [private]

### 9.22.3 Member Data Documentation

**9.22.3.1 int Bds::DataFileBdrs::oyear** [private]

**9.22.3.2 BTimeStamp Bds::DataFileBdrs::fileNameDate** [private]

**9.22.3.3 BUInt32 Bds::DataFileBdrs::oblockSize** [private]

**9.22.3.4 double Bds::DataFileBdrs::osampleRate** [private]

**9.22.3.5 DataInfo Bds::DataFileBdrs::odataInfo** [private]

**9.22.3.6 int Bds::DataFileBdrs::omagic** [private]

**9.22.3.7 int Bds::DataFileBdrs::oblockYear** [private]

**9.22.3.8 BArray<BUInt64> Bds::DataFileBdrs::oblockPositions** [private]

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

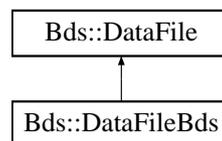
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs-1.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs-1.cpp](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra-1.cpp](#)

## 9.23 Bds::DataFileBds Class Reference

This class implements the BDS Data File/Stream access system.

```
#include <BdsDataFileBds.h>
```

Inheritance diagram for Bds::DataFileBds::



### Public Types

- enum { [StreamsMax](#) = 64 }
- enum { [DefaultBlockSize](#) = 65536 }
- enum [PackFormat](#) { [PackFormat\\_Unknown](#) = 0, [PackFormat\\_SM](#) = 1, [PackFormat\\_CM](#) = 2, [PackFormat\\_SM\\_CC](#) = 3 }

### Public Member Functions

- [DataFileBds](#) ()
- [~DataFileBds](#) ()
- BError [open](#) (BString fileName, BString mode)  
*Open the file for reading or writing.*
- void [close](#) ()  
*Close file.*
- BError [setFormat](#) (BString format)  
*Sets the sub-format.*
- BError [setInfo](#) ([DataInfo](#) &dataInfo, BList< [ChannelInfo](#) > &channelInfos)  
*Sets the information.*
- BError [writeData](#) ([DataBlock](#) &data)  
*Writes a data block to the file.*
- [DataOrder](#) [getDataOrder](#) ()  
*Get the expected order of writing data, by sample or by channel.*
- BError [getInfo](#) ([DataInfo](#) &dataInfo, [Options](#) options, BList< BError > &errors)  
*Get information on open file.*
- BError [seekBlock](#) (BUInt32 channel, BTimeStamp time, BUInt32 &blockNumber)  
*Find the block that contains the samples for the time requested.*
- BError [readData](#) (BUInt32 channel, BUInt32 blockNumber, [DataBlock](#) &data)

*Read the data block for the given channel or all channels if blockNumber is 0.*

- BError [setDiskBlockSize](#) (BUInt32 blockSize)  
*Sets up file/stream block size.*
- BUInt32 [getDiskBlockSize](#) ()  
*Returns the data block size in bytes.*

## Static Public Member Functions

- static BStringList [getFormats](#) ()  
*Get the names of the supported formats.*

## Private Member Functions

- void [clear](#) ()
- BError [setInfoRepeat](#) (BUInt32 repeat)  
*Sets the repeat period for the information packets.*
- BError [infoSet](#) (BUInt32 streamlet, BTimeStamp startTime, BTimeStamp endTime, BDictString &info)  
*Sets the information packet and writes this to the file/stream.*
- BError [infoGet](#) (BdsDataPacket &packet, BDictString &info)  
*Gets the information from an information packet.*
- BError [addInfoPacket](#) ()
- BError [dataAppend](#) (BUInt32 streamlet, DataBlock &data)  
*Appends a data block to the file.*
- BError [dataGet](#) (BUInt channel, BdsDataPacket &packet, DataBlock &data)  
*Gets the data from a data packet.*
- BError [packetRead](#) (BdsDataPacket &packet)  
*Reads a packet from the file.*
- BError [packetWrite](#) (BdsDataPacket &packet)  
*Writes a packet to the file.*
- BError [packetSeek](#) (BUInt32 streamlet, BUInt32 packetNumber, BdsDataPacket &packet)
- void [diskBlockReset](#) ()  
*Resets to disk block 0.*
- BError [diskBlockWrite](#) (void \*data, BUInt32 numBytes, int header=0)  
*Writes data to a block in the file.*
- BError [diskBlockWriteFlush](#) ()

*Flushes a semi-filled block to the file.*

- BError [diskBlockRead](#) (void \*data, BUInt32 numBytes, int header=0)  
*Reads data from a block.*
- BError [diskBlockSeek](#) (BUInt64 position)

### Private Attributes

- BString [oformat](#)  
*The format.*
- PackFormat [opackFormat](#)  
*The data packet format.*
- DataInfo [odataInfo](#)  
*Information on the channel data.*
- BList< [ChannelInfo](#) > [ochannelInfos](#)  
*Channel meta data information.*
- BUInt32 [oinfoRepeat](#)  
*How often to output info packet.*
- BDictString [oinfo](#)  
*Info packet output.*
- BdsDataPacketHeader [oinfoHeader](#)  
*Info packet header.*
- BUInt32 [odiskBlockSize](#)  
*The block size.*
- BdsDataBlock \* [odiskBlock](#)  
*Data block buffer.*
- BUInt64 [odiskPosition](#)  
*The disk block position.*
- BArray< [BdsDataStreamlet](#) > [ostreamlets](#)  
*Streamlet sequence numbers and positions.*
- BdsDataPacket [opacket](#)  
*Temporary packet for header/data packet generation.*

#### 9.23.1 Detailed Description

This class implements the BDS Data File/Stream access system.

## 9.23.2 Member Enumeration Documentation

### 9.23.2.1 anonymous enum

Enumerator:

*StreamsMax*

### 9.23.2.2 anonymous enum

Enumerator:

*DefaultBlockSize*

### 9.23.2.3 enum Bds::DataFileBds::PackFormat

Enumerator:

*PackFormat\_Unknown*

*PackFormat\_SM*

*PackFormat\_CM*

*PackFormat\_SM\_CC*

## 9.23.3 Constructor & Destructor Documentation

### 9.23.3.1 Bds::DataFileBds::DataFileBds ()

### 9.23.3.2 Bds::DataFileBds::~~DataFileBds ()

## 9.23.4 Member Function Documentation

### 9.23.4.1 BError Bds::DataFileBds::open (BString *fileName*, BString *mode*) [virtual]

Open the file for reading or writing.

Reimplemented from [Bds::DataFile](#).

### 9.23.4.2 void Bds::DataFileBds::close () [virtual]

Close file.

Reimplemented from [Bds::DataFile](#).

### 9.23.4.3 BError Bds::DataFileBds::setFormat (BString *format*) [virtual]

Sets the sub-format.

Reimplemented from [Bds::DataFile](#).

**9.23.4.4 BError Bds::DataFileBds::setInfo (DataInfo & *dataInfo*, BList< ChannelInfo > & *channelInfos*)** [virtual]

Sets the information.

Reimplemented from [Bds::DataFile](#).

**9.23.4.5 BError Bds::DataFileBds::writeData (DataBlock & *data*)** [virtual]

Writes a data block to the file.

Reimplemented from [Bds::DataFile](#).

**9.23.4.6 DataFile::DataOrder Bds::DataFileBds::getDataOrder ()** [virtual]

Get the expected order of writing data, by sample or by channel.

Reimplemented from [Bds::DataFile](#).

**9.23.4.7 BError Bds::DataFileBds::getInfo (DataInfo & *dataInfo*, Options *options*, BList< BError > & *errors*)**

Get information on open file.

**9.23.4.8 BError Bds::DataFileBds::seekBlock (BUInt32 *channel*, BTimeStamp *time*, BUInt32 & *blockNumber*)** [virtual]

Find the block that contains the samples for the time requested.

Reimplemented from [Bds::DataFile](#).

**9.23.4.9 BError Bds::DataFileBds::readData (BUInt32 *channel*, BUInt32 *blockNumber*, DataBlock & *data*)** [virtual]

Read the data block for the given channel or all channels if *blockNumber* is 0.

Reimplemented from [Bds::DataFile](#).

**9.23.4.10 BStringList Bds::DataFileBds::getFormats ()** [static]

Get the names of the supported formats.

Reimplemented from [Bds::DataFile](#).

**9.23.4.11 BError Bds::DataFileBds::setDiskBlockSize (BUInt32 *blockSize*)**

Sets up file/stream block size.

**9.23.4.12 uint32\_t Bds::DataFileBds::getDiskBlockSize ()**

Returns the data block size in bytes.

**9.23.4.13** void `Bds::DataFileBds::clear ()` [private]

**9.23.4.14** BError `Bds::DataFileBds::setInfoRepeat (BUInt32 repeat)` [private]

Sets the reapeat period for the information packets.

**9.23.4.15** BError `Bds::DataFileBds::infoSet (BUInt32 streamlet, BTimeStamp startTime, BTimeStamp endTime, BDictString & info)` [private]

Sets the information packet and writes this to the file/stream.

**9.23.4.16** BError `Bds::DataFileBds::infoGet (BdsDataPacket & packet, BDictString & info)` [private]

Gets the information from an information packet.

**9.23.4.17** BError `Bds::DataFileBds::addInfoPacket ()` [private]

**9.23.4.18** BError `Bds::DataFileBds::dataAppend (BUInt32 streamlet, DataBlock & data)` [private]

Appends a data block to the file.

**9.23.4.19** BError `Bds::DataFileBds::dataGet (BUInt channel, BdsDataPacket & packet, DataBlock & data)` [private]

Gets the data from a data packet.

**9.23.4.20** BError `Bds::DataFileBds::packetRead (BdsDataPacket & packet)` [private]

Reads a packet from the file.

**9.23.4.21** BError `Bds::DataFileBds::packetWrite (BdsDataPacket & packet)` [private]

Writes a packet to the file.

**9.23.4.22** BError `Bds::DataFileBds::packetSeek (BUInt32 streamlet, BUInt32 packetNumber, BdsDataPacket & packet)` [private]

**9.23.4.23** void `Bds::DataFileBds::diskBlockReset ()` [private]

Resets to disk block 0.

**9.23.4.24** BError `Bds::DataFileBds::diskBlockWrite (void * data, BUInt32 numBytes, int header = 0)` [private]

Writes data to a block in the file.

**9.23.4.25 BError Bds::DataFileBds::diskBlockWriteFlush ()** [private]

Flushes a semi-filled block to the file.

**9.23.4.26 BError Bds::DataFileBds::diskBlockRead (void \* data, BUInt32 numBytes, int header = 0)** [private]

Reads data from a block.

**9.23.4.27 BError Bds::DataFileBds::diskBlockSeek (BUInt64 position)** [private]**9.23.5 Member Data Documentation****9.23.5.1 BString Bds::DataFileBds::oformat** [private]

The format.

Reimplemented from [Bds::DataFile](#).

**9.23.5.2 PackFormat Bds::DataFileBds::opackFormat** [private]

The data packet format.

**9.23.5.3 DataInfo Bds::DataFileBds::odataInfo** [private]

Information on the channel data.

**9.23.5.4 BList<ChannelInfo> Bds::DataFileBds::ochannelInfos** [private]

[Channel](#) meta data information.

**9.23.5.5 BUInt32 Bds::DataFileBds::oinfoRepeat** [private]

How often to output info packet.

**9.23.5.6 BDictString Bds::DataFileBds::oinfo** [private]

Info packet output.

**9.23.5.7 BdsDataPacketHeader Bds::DataFileBds::oinfoHeader** [private]

Info packet header.

**9.23.5.8 BUInt32 Bds::DataFileBds::odiskBlockSize** [private]

The block size.

**9.23.5.9 BdsDataBlock\* Bds::DataFileBds::odiskBlock** [private]

Data block buffer.

**9.23.5.10 BUInt64 Bds::DataFileBds::odiskPosition** [private]

The disk block position.

**9.23.5.11 BArray<BdsDataStreamlet> Bds::DataFileBds::ostreamlets** [private]

Streamlet sequence numbers and positions.

**9.23.5.12 BdsDataPacket Bds::DataFileBds::opacket** [private]

Temporary packet for header/data packet generation.

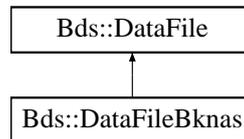
The documentation for this class was generated from the following files:

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp](#)

## 9.24 Bds::DataFileBknas Class Reference

```
#include <BdsDataFileBknas.h>
```

Inheritance diagram for Bds::DataFileBknas::



### Public Member Functions

- [DataFileBknas](#) ()
- BError [open](#) (BString fileName, BString mode)  
*Open the file for read or write.*
- BError [setInfo](#) ([DataInfo](#) &dataInfo, BList< [ChannelInfo](#) > &channelInfos)  
*Set information on data for write.*
- BError [writeData](#) ([DataBlock](#) &data)  
*Write a block of data.*

### Static Public Member Functions

- static BStringList [getFormats](#) ()  
*Get list of supported formats.*

### Private Attributes

- [DataInfo](#) odataInfo

### 9.24.1 Constructor & Destructor Documentation

#### 9.24.1.1 Bds::DataFileBknas::DataFileBknas ()

### 9.24.2 Member Function Documentation

#### 9.24.2.1 BError Bds::DataFileBknas::open (BString *fileName*, BString *mode*) [virtual]

Open the file for read or write.

Reimplemented from [Bds::DataFile](#).

#### 9.24.2.2 BError Bds::DataFileBknas::setInfo (DataInfo & *dataInfo*, BList< ChannelInfo > & *channelInfos*) [virtual]

Set information on data for write.

Reimplemented from [Bds::DataFile](#).

#### 9.24.2.3 BError Bds::DataFileBknas::writeData (DataBlock & *data*) [virtual]

Write a block of data.

Reimplemented from [Bds::DataFile](#).

#### 9.24.2.4 BStringList Bds::DataFileBknas::getFormats () [static]

Get list of supported formats.

Reimplemented from [Bds::DataFile](#).

### 9.24.3 Member Data Documentation

#### 9.24.3.1 DataInfo Bds::DataFileBknas::odataInfo [private]

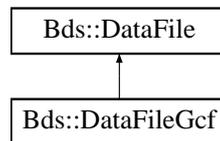
The documentation for this class was generated from the following files:

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp](#)

## 9.25 Bds::DataFileGcf Class Reference

```
#include <BdsDataFileGcf.h>
```

Inheritance diagram for Bds::DataFileGcf::



### Public Member Functions

- [DataFileGcf](#) ()
- int [getFeatures](#) ()  
*Get bitmask of supported features.*
- BError [getInfo](#) ([DataInfo](#) &dataInfo, [Options](#) options, BList< BError > &errors)
- BError [readData](#) (BUInt32 channel, BUInt32 blockNumber, [DataBlock](#) &data)  
*Read a block.*

### Static Public Member Functions

- static BStringList [getFormats](#) ()  
*Get list of supported formats.*

### Private Attributes

- BUInt32 [oblockSize](#)
- [DataInfo](#) [odataInfo](#)
- BString [osystemId](#)
- BString [ostreamId](#)
- int [offormat](#)
- int [ofilterCoef](#)
- double [osampleRate](#)

### 9.25.1 Constructor & Destructor Documentation

#### 9.25.1.1 Bds::DataFileGcf::DataFileGcf ()

### 9.25.2 Member Function Documentation

#### 9.25.2.1 int Bds::DataFileGcf::getFeatures () [virtual]

Get bitmask of supported features.

Reimplemented from [Bds::DataFile](#).

**9.25.2.2** `BError Bds::DataFileGcf::getInfo (DataInfo & dataInfo, Options options, BList< BError > & errors)`

**9.25.2.3** `BError Bds::DataFileGcf::readData (BUInt32 channel, BUInt32 blockNumber, DataBlock & data)` [virtual]

Read a block.

Reimplemented from [Bds::DataFile](#).

**9.25.2.4** `BStringList Bds::DataFileGcf::getFormats ()` [static]

Get list of supported formats.

Reimplemented from [Bds::DataFile](#).

### 9.25.3 Member Data Documentation

**9.25.3.1** `BUInt32 Bds::DataFileGcf::oblockSize` [private]

**9.25.3.2** `DataInfo Bds::DataFileGcf::odataInfo` [private]

**9.25.3.3** `BString Bds::DataFileGcf::osystemId` [private]

**9.25.3.4** `BString Bds::DataFileGcf::ostreamId` [private]

**9.25.3.5** `int Bds::DataFileGcf::oformat` [private]

Reimplemented from [Bds::DataFile](#).

**9.25.3.6** `int Bds::DataFileGcf::ofilterCoef` [private]

**9.25.3.7** `double Bds::DataFileGcf::osampleRate` [private]

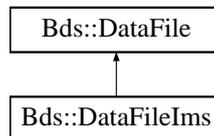
The documentation for this class was generated from the following files:

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp](#)

## 9.26 Bds::DataFileImgs Class Reference

```
#include <BdsDataFileImgs.h>
```

Inheritance diagram for Bds::DataFileImgs::



### Public Member Functions

- [DataFileImgs](#) ()
- BError [open](#) (BString fileName, BString mode)  
*Open the file for read or write.*
- void [close](#) ()  
*Close the file.*
- [DataOrder](#) [getDataOrder](#) ()  
*Get the expected order of writing data, by sample or by channel.*
- int [getFeatures](#) ()  
*Get bitmask of supported features.*
- BError [setInfo](#) ([DataInfo](#) &dataInfo, BList< [ChannelInfo](#) > &channelInfos)  
*Set information on data for write.*
- BError [start](#) (BUInt channel, BUInt segment)  
*Start next segment of data.*
- BError [writeData](#) ([DataBlock](#) &data)  
*Write a block of data.*
- BError [end](#) ()  
*End segment.*

### Static Public Member Functions

- static BStringList [getFormats](#) ()  
*Get list of supported formats.*

## Private Attributes

- [DataInfo](#) `odataInfo`
- [BList](#)< [ChannelInfo](#) > `ochannelInfos`
- [BUInt](#) `owriteChannel`
- [BInt32](#) `owriteChecksum`
- [BUInt](#) `owriteColumn`
- [BInt32](#) `oprevValue1`
- [BInt32](#) `oprevValue2`

## 9.26.1 Constructor & Destructor Documentation

### 9.26.1.1 `Bds::DataFileIms::DataFileIms ()`

## 9.26.2 Member Function Documentation

### 9.26.2.1 `BError Bds::DataFileIms::open (BString fileName, BString mode) [virtual]`

Open the file for read or write.

Reimplemented from [Bds::DataFile](#).

### 9.26.2.2 `void Bds::DataFileIms::close () [virtual]`

Close the file.

Reimplemented from [Bds::DataFile](#).

### 9.26.2.3 `DataFile::DataOrder Bds::DataFileIms::getDataOrder () [virtual]`

Get the expected order of writing data, by sample or by channel.

Reimplemented from [Bds::DataFile](#).

### 9.26.2.4 `int Bds::DataFileIms::getFeatures () [virtual]`

Get bitmask of supported features.

Reimplemented from [Bds::DataFile](#).

### 9.26.2.5 `BError Bds::DataFileIms::setInfo (DataInfo & dataInfo, BList< ChannelInfo > & channelInfos) [virtual]`

Set information on data for write.

Reimplemented from [Bds::DataFile](#).

### 9.26.2.6 `BError Bds::DataFileIms::start (BUInt channel, BUInt segment) [virtual]`

Start next segment of data.

Reimplemented from [Bds::DataFile](#).

**9.26.2.7 BError Bds::DataFileImgs::writeData (DataBlock & data) [virtual]**

Write a block of data.

Reimplemented from [Bds::DataFile](#).

**9.26.2.8 BError Bds::DataFileImgs::end () [virtual]**

End segment.

Reimplemented from [Bds::DataFile](#).

**9.26.2.9 BStringList Bds::DataFileImgs::getFormats () [static]**

Get list of supported formats.

Reimplemented from [Bds::DataFile](#).

**9.26.3 Member Data Documentation****9.26.3.1 DataInfo Bds::DataFileImgs::odataInfo [private]****9.26.3.2 BList<ChannelInfo> Bds::DataFileImgs::ochannelInfos [private]****9.26.3.3 BUInt Bds::DataFileImgs::owriteChannel [private]****9.26.3.4 BInt32 Bds::DataFileImgs::owriteChecksum [private]****9.26.3.5 BUInt Bds::DataFileImgs::owriteColumn [private]****9.26.3.6 BInt32 Bds::DataFileImgs::oprevValue1 [private]****9.26.3.7 BInt32 Bds::DataFileImgs::oprevValue2 [private]**

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

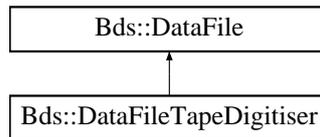
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileImgs.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileImgs.cpp](#)

## 9.27 Bds::DataFileTapeDigitiser Class Reference

This class implements the TapeDigitiser's file output conversion and storing system.

```
#include <BdsDataFileTapeDigitiser.h>
```

Inheritance diagram for Bds::DataFileTapeDigitiser::



### Public Member Functions

- [DataFileTapeDigitiser](#) ()
- BError [open](#) (BString fileName, BString mode)  
*Open the file for reading or writing.*
- BError [getInfo](#) (DataInfo &dataInfo, Options options, BList< BError > &errors)
- BError [readData](#) (BUInt32 channel, BUInt32 blockNumber, DataBlock &data)  
*Read a block.*

### Static Public Member Functions

- static BStringList [getFormats](#) ()  
*Get list of supported formats.*

### Private Member Functions

- uint32\_t [blockSize](#) ()  
*Returns the data block size in bytes.*
- uint32\_t [blockNumSamples](#) ()  
*Returns the number of samples of data in a block.*
- BError [readHeader](#) (BDictString &header)  
*Read the file header info.*
- uint32\_t [computeChecksum](#) (void \*data, int nBytes)

### Private Attributes

- BDictString [oheader](#)
- off64\_t [oheaderSize](#)
- uint32\_t [oblockSize](#)

- uint32\_t `oblockNumSamples`
- double `osampleRate`
- BBuffer `obuffer`

*Data block buffer.*

- BString `oversion`

## 9.27.1 Detailed Description

This class implements the TapeDigitiser's file output conversion and storing system.

## 9.27.2 Constructor & Destructor Documentation

### 9.27.2.1 Bds::DataFileTapeDigitiser::DataFileTapeDigitiser ()

## 9.27.3 Member Function Documentation

### 9.27.3.1 BError Bds::DataFileTapeDigitiser::open (BString *fileName*, BString *mode*) [virtual]

Open the file for reading or writing.

Reimplemented from [Bds::DataFile](#).

### 9.27.3.2 BError Bds::DataFileTapeDigitiser::getInfo (DataInfo & *dataInfo*, Options *options*, BList< BError > & *errors*)

### 9.27.3.3 BError Bds::DataFileTapeDigitiser::readData (BUInt32 *channel*, BUInt32 *blockNumber*, DataBlock & *data*) [virtual]

Read a block.

Reimplemented from [Bds::DataFile](#).

### 9.27.3.4 BStringList Bds::DataFileTapeDigitiser::getFormats () [static]

Get list of supported formats.

Reimplemented from [Bds::DataFile](#).

### 9.27.3.5 uint32\_t Bds::DataFileTapeDigitiser::blockSize () [private]

Returns the data block size in bytes.

### 9.27.3.6 uint32\_t Bds::DataFileTapeDigitiser::blockNumSamples () [private]

Returns the number of samples of data in a block.

**9.27.3.7 BError Bds::DataFileTapeDigitiser::readHeader (BDictString & *header*)** [private]

Read the file header info.

**9.27.3.8 uint32\_t Bds::DataFileTapeDigitiser::computeChecksum (void \* *data*, int *nBytes*)**  
[private]**9.27.4 Member Data Documentation****9.27.4.1 BDictString Bds::DataFileTapeDigitiser::oheader** [private]**9.27.4.2 off64\_t Bds::DataFileTapeDigitiser::oheaderSize** [private]**9.27.4.3 uint32\_t Bds::DataFileTapeDigitiser::oblockSize** [private]**9.27.4.4 uint32\_t Bds::DataFileTapeDigitiser::oblockNumSamples** [private]**9.27.4.5 double Bds::DataFileTapeDigitiser::osampleRate** [private]**9.27.4.6 BBuffer Bds::DataFileTapeDigitiser::obuffer** [private]

Data block buffer.

**9.27.4.7 BString Bds::DataFileTapeDigitiser::oversion** [private]

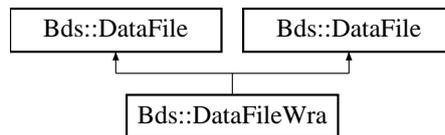
The documentation for this class was generated from the following files:

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp](#)

## 9.28 Bds::DataFileWra Class Reference

```
#include <BdsDataFileWra.h>
```

Inheritance diagram for Bds::DataFileWra::



### Public Member Functions

- [DataFileWra](#) ()
- BError [open](#) (BString fileName, BString mode)
  - Open the file for read or write.*
- BError [setFormat](#) (BString format)
  - Set the sub-format.*
- int [getFeatures](#) ()
  - Get bitmask of supported features.*
- [DataOrder](#) [getDataOrder](#) ()
  - Get the expected order of writing data, by sample or by channel.*
- BError [getInfo](#) ([DataInfo](#) &dataInfo, int validate, int fix, BList< BError > &errors)
- BError [readData](#) (BUInt32 channel, BUInt32 blockNumber, [DataBlock](#) &data)
  - Read a block.*
- [DataFileWra](#) ()
- BError [setFormat](#) (BString format)
  - Set the sub-format.*
- int [getFeatures](#) ()
  - Get bitmask of supported features.*
- [DataOrder](#) [getDataOrder](#) ()
  - Get the expected order of writing data, by sample or by channel.*
- BError [getInfo](#) ([DataInfo](#) &dataInfo, [Options](#) options, BList< BError > &errors)
- BError [readData](#) (BUInt32 channel, BUInt32 blockNumber, [DataBlock](#) &data)
  - Read a block.*

## Static Public Member Functions

- static BStringList [getFormats](#) ()  
*Get list of supported formats.*
- static BStringList [getFormats](#) ()  
*Get list of supported formats.*

## Private Member Functions

- BError [readBlock](#) (BUInt32 channel, BUInt64 pos, [DataBlock](#) &data)
- BError [readBlock](#) (BUInt32 channel, BUInt64 pos, [DataBlock](#) &data)

## Private Attributes

- int [oyear](#)
- int [oblockYear](#)
- BTimeStamp [ofilenameDate](#)
- BUInt32 [oblockSize](#)
- BUInt32 [onumChannels](#)
- double [osampleRate](#)
- [DataInfo](#) [odataInfo](#)
- BArray< BUInt64 > [oblockPositions](#)
- int [omagic](#)

## 9.28.1 Constructor & Destructor Documentation

**9.28.1.1** Bds::DataFileWra::DataFileWra ()

**9.28.1.2** Bds::DataFileWra::DataFileWra ()

## 9.28.2 Member Function Documentation

**9.28.2.1** BError Bds::DataFileWra::open (BString *fileName*, BString *mode*) [virtual]

Open the file for read or write.

Reimplemented from [Bds::DataFile](#).

**9.28.2.2** BError Bds::DataFileWra::setFormat (BString *format*) [virtual]

Set the sub-format.

Reimplemented from [Bds::DataFile](#).

**9.28.2.3** int Bds::DataFileWra::getFeatures () [virtual]

Get bitmask of supported features.

Reimplemented from [Bds::DataFile](#).

**9.28.2.4 DataFile::DataOrder Bds::DataFileWra::getDataOrder ()** [virtual]

Get the expected order of writing data, by sample or by channel.

Reimplemented from [Bds::DataFile](#).

**9.28.2.5 BError Bds::DataFileWra::getInfo (DataInfo & *dataInfo*, int *validate*, int *fix*, BList< BError > & *errors*)****9.28.2.6 BError Bds::DataFileWra::readData (BUInt32 *channel*, BUInt32 *blockNumber*, DataBlock & *data*)** [virtual]

Read a block.

Reimplemented from [Bds::DataFile](#).

**9.28.2.7 BStringList Bds::DataFileWra::getFormats ()** [static]

Get list of supported formats.

Reimplemented from [Bds::DataFile](#).

**9.28.2.8 BError Bds::DataFileWra::readBlock (BUInt32 *channel*, BUInt64 *pos*, DataBlock & *data*)** [private]**9.28.2.9 BError Bds::DataFileWra::setFormat (BString *format*)** [virtual]

Set the sub-format.

Reimplemented from [Bds::DataFile](#).

**9.28.2.10 int Bds::DataFileWra::getFeatures ()** [virtual]

Get bitmask of supported features.

Reimplemented from [Bds::DataFile](#).

**9.28.2.11 DataOrder Bds::DataFileWra::getDataOrder ()** [virtual]

Get the expected order of writing data, by sample or by channel.

Reimplemented from [Bds::DataFile](#).

**9.28.2.12 BError Bds::DataFileWra::getInfo (DataInfo & *dataInfo*, Options *options*, BList< BError > & *errors*)****9.28.2.13 BError Bds::DataFileWra::readData (BUInt32 *channel*, BUInt32 *blockNumber*, DataBlock & *data*)** [virtual]

Read a block.

Reimplemented from [Bds::DataFile](#).

#### 9.28.2.14 static BStringList Bds::DataFileWra::getFormats () [static]

Get list of supported formats.

Reimplemented from [Bds::DataFile](#).

#### 9.28.2.15 BError Bds::DataFileWra::readBlock (BUInt32 *channel*, BUInt64 *pos*, DataBlock & *data*) [private]

### 9.28.3 Member Data Documentation

#### 9.28.3.1 int Bds::DataFileWra::oyear [private]

#### 9.28.3.2 int Bds::DataFileWra::oblockYear [private]

#### 9.28.3.3 BTimeStamp Bds::DataFileWra::ofileNameDate [private]

#### 9.28.3.4 BUInt32 Bds::DataFileWra::oblockSize [private]

#### 9.28.3.5 BUInt32 Bds::DataFileWra::onumChannels [private]

#### 9.28.3.6 double Bds::DataFileWra::osampleRate [private]

#### 9.28.3.7 DataInfo Bds::DataFileWra::odataInfo [private]

#### 9.28.3.8 BArray< BUInt64 > Bds::DataFileWra::oblockPositions [private]

#### 9.28.3.9 int Bds::DataFileWra::omagic [private]

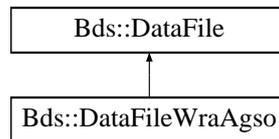
The documentation for this class was generated from the following files:

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra-1.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra-1.cpp](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp](#)

## 9.29 Bds::DataFileWraAgso Class Reference

```
#include <BdsDataFileWraAgso.h>
```

Inheritance diagram for Bds::DataFileWraAgso::



### Public Member Functions

- [DataFileWraAgso](#) ()
- [int getFeatures](#) ()  
*Get bitmask of supported features.*
- [DataOrder getDataOrder](#) ()  
*Get the expected order of writing data, by sample or by channel.*
- [BError getInfo](#) ([DataInfo](#) &dataInfo, [Options](#) options, [BList](#)< [BError](#) > &errors)
- [BError readData](#) ([BUInt32](#) channel, [BUInt32](#) blockNumber, [DataBlock](#) &data)  
*Read a block.*

### Static Public Member Functions

- [static BStringList getFormats](#) ()  
*Get list of supported formats.*

### Private Member Functions

- [BError readBlock](#) ([BUInt32](#) blockNumber, [BString](#) &channelheader, [DataBlock](#) &data)

### Private Attributes

- [int oyear](#)
- [int oblockYear](#)
- [BUInt32 oblockSize](#)
- [double osampleRate](#)
- [DataInfo odataInfo](#)
- [BArray](#)< [BUInt](#) > [ochannelStartBlocks](#)
- [BUInt oreadChannel](#)
- [BUInt oreadBlock](#)
- [BUInt ocurrentBlock](#)

## Static Private Attributes

- static int `ofieldWidths` [] = { 5, 10, 14, 7, 6, 7, 12, 10, 7, 7, 2, 3, 7, 11, 2, 31, 21, 11, 6, 3, 5, 0 }

## 9.29.1 Constructor & Destructor Documentation

### 9.29.1.1 `Bds::DataFileWraAgso::DataFileWraAgso ()`

## 9.29.2 Member Function Documentation

### 9.29.2.1 `int Bds::DataFileWraAgso::getFeatures ()` [virtual]

Get bitmask of supported features.

Reimplemented from [Bds::DataFile](#).

### 9.29.2.2 `DataFile::DataOrder Bds::DataFileWraAgso::getDataOrder ()` [virtual]

Get the expected order of writing data, by sample or by channel.

Reimplemented from [Bds::DataFile](#).

### 9.29.2.3 `BError Bds::DataFileWraAgso::getInfo (DataInfo & dataInfo, Options options, BList< BError > & errors)`

### 9.29.2.4 `BError Bds::DataFileWraAgso::readData (BUInt32 channel, BUInt32 blockNumber, DataBlock & data)` [virtual]

Read a block.

Reimplemented from [Bds::DataFile](#).

### 9.29.2.5 `BStringList Bds::DataFileWraAgso::getFormats ()` [static]

Get list of supported formats.

Reimplemented from [Bds::DataFile](#).

**9.29.2.6** **BError Bds::DataFileWraAgso::readBlock (BUInt32 *blockNumber*, BString & *channelheader*, DataBlock & *data*)** [private]

### 9.29.3 Member Data Documentation

**9.29.3.1** **int Bds::DataFileWraAgso::oyear** [private]

**9.29.3.2** **int Bds::DataFileWraAgso::oblockYear** [private]

**9.29.3.3** **BUInt32 Bds::DataFileWraAgso::oblockSize** [private]

**9.29.3.4** **double Bds::DataFileWraAgso::osampleRate** [private]

**9.29.3.5** **DataInfo Bds::DataFileWraAgso::odataInfo** [private]

**9.29.3.6** **BArray<BUInt> Bds::DataFileWraAgso::ochannelStartBlocks** [private]

**9.29.3.7** **BUInt Bds::DataFileWraAgso::oreadChannel** [private]

**9.29.3.8** **BUInt Bds::DataFileWraAgso::oreadBlock** [private]

**9.29.3.9** **BUInt Bds::DataFileWraAgso::ocurrentBlock** [private]

**9.29.3.10** **int Bds::DataFileWraAgso::ofieldWidths = { 5, 10, 14, 7, 6, 7, 12, 10, 7, 7, 2, 3, 7, 11, 2, 31, 21, 11, 6, 3, 5, 0 }** [static, private]

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

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp](#)

## 9.30 Bds::DataFormats Class Reference

This class defines the interface for generic data file access.

```
#include <BdsDataLib.h>
```

### Public Member Functions

- [DataFormats](#) ()
- [~DataFormats](#) ()
- BError [formatList](#) (BList< BStringList > &formats)
- BError [formatGet](#) (BString format, [DataFile](#) \*&dataFile)

### Protected Member Functions

- int [findString](#) (BStringList stringList, BString string)

### 9.30.1 Detailed Description

This class defines the interface for generic data file access.

### 9.30.2 Constructor & Destructor Documentation

**9.30.2.1** [Bds::DataFormats::DataFormats](#) ()

**9.30.2.2** [Bds::DataFormats::~~DataFormats](#) ()

### 9.30.3 Member Function Documentation

**9.30.3.1** BError [Bds::DataFormats::formatList](#) (BList< BStringList > & *formats*)

**9.30.3.2** BError [Bds::DataFormats::formatGet](#) (BString *format*, [DataFile](#) \*& *dataFile*)

**9.30.3.3** int [Bds::DataFormats::findString](#) (BStringList *stringList*, BString *string*)  
[protected]

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

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp](#)

## 9.31 Bds::DataHandle Class Reference

This defines a handle to a data stream/file.

```
#include <BdsD.h>
```

### Public Member Functions

- [DataHandle](#) (BUInt32 phandle=BUInt32(), BUInt32 pstartBlock=BUInt32())

### Public Attributes

- BUInt32 [handle](#)  
*Opaque file handle.*
- BUInt32 [startBlock](#)  
*The starting block containing the start time requested.*

#### 9.31.1 Detailed Description

This defines a handle to a data stream/file.

#### 9.31.2 Constructor & Destructor Documentation

**9.31.2.1 Bds::DataHandle::DataHandle (BUInt32 phandle = BUInt32 (), BUInt32 pstartBlock = BUInt32 ())**

#### 9.31.3 Member Data Documentation

##### 9.31.3.1 BUInt32 Bds::DataHandle::handle

Opaque file handle.

##### 9.31.3.2 BUInt32 Bds::DataHandle::startBlock

The starting block containing the start time requested.

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

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

## 9.32 Bds::DataInfo Class Reference

```
#include <BdsD.h>
```

### Public Member Functions

- **DataInfo** (BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BString parray=BString(), BString pdescription=BString(), BUInt32 psynchronous=BUInt32(), BArray< BArray< [DataChannel](#) > > pchannels=BArray< BArray< [DataChannel](#) > >(), BDict< BString > pinfo=BDict< BString >())

### Public Attributes

- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time.*
- BString [array](#)  
*The Seismic Array that all of the channels are from, if just one.*
- BString [description](#)  
*The Comment.*
- BUInt32 [synchronous](#)  
*The channels are synchronously sampled.*
- BArray< BArray< [DataChannel](#) > > [channels](#)  
*The Data channels. Each channel can have multiple segments of data.*
- BDict< BString > [info](#)  
*Extra info on the set of channels.*

### 9.32.1 Constructor & Destructor Documentation

- 9.32.1.1 Bds::DataInfo::DataInfo (BTimeStamp pstartTime = BTimeStamp(), BTimeStamp pendTime = BTimeStamp(), BString parray = BString(), BString pdescription = BString(), BUInt32 psynchronous = BUInt32(), BArray< BArray< [DataChannel](#) > > pchannels = BArray<BArray<[DataChannel](#) > >(), BDict< BString > pinfo = BDict<BString >())**

### 9.32.2 Member Data Documentation

#### 9.32.2.1 BTimeStamp Bds::DataInfo::startTime

The Start Time.

### 9.32.2.2 BTimeStamp Bds::DataInfo::endTime

The End Time.

### 9.32.2.3 BString Bds::DataInfo::array

The Seismic Array that all of the channels are from, if just one.

### 9.32.2.4 BString Bds::DataInfo::description

The Comment.

### 9.32.2.5 BUInt32 Bds::DataInfo::synchronous

The channels are synchronously sampled.

### 9.32.2.6 BArray<BArray<DataChannel > > Bds::DataInfo::channels

The Data channels. Each channel can have multiple segments of data.

### 9.32.2.7 BDict<BString > Bds::DataInfo::info

Extra info on the set of channels.

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

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

## 9.33 Bds::Digitiser Class Reference

This class defines a [Digitiser](#).

```
#include <BdsD.h>
```

### Public Member Functions

- [Digitiser](#) (BUInt32 pid=BUInt32(), BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BString pname=BString(), BString ptype=BString(), BString pserialNumber=BString(), BUInt32 pnumberChannels=BUInt32(), BFloat64 pbaseSamplingFrequency=BFloat64(), BFloat64 pinitialSamplingFrequency=BFloat64(), BFloat64 pgain=BFloat64())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*The ID.*
- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time the channel was available.*
- BString [name](#)  
*The Digitisers name.*
- BString [type](#)  
*The Digitisers type.*
- BString [serialNumber](#)  
*The digitisers's serial number.*
- BUInt32 [numberChannels](#)  
*The number of supported channels.*
- BFloat64 [baseSamplingFrequency](#)  
*The base sampling frequency.*
- BFloat64 [initialSamplingFrequency](#)  
*The initial pre-decimation sampling frequency.*
- BFloat64 [gain](#)  
*The overall gain of the digitiser at the manufacturers calibration frequency. (For information only).*

## 9.33.1 Detailed Description

This class defines a [Digitiser](#).

## 9.33.2 Constructor & Destructor Documentation

**9.33.2.1 Bds::Digitiser::Digitiser (BUInt32 *pid* = BUInt32 (), BTimeStamp *pstartTime* = BTimeStamp (), BTimeStamp *pendTime* = BTimeStamp (), BString *pname* = BString (), BString *pctype* = BString (), BString *pserialNumber* = BString (), BUInt32 *pnumberChannels* = BUInt32 (), BFloat64 *pbaseSamplingFrequency* = BFloat64 (), BFloat64 *pinitialSamplingFrequency* = BFloat64 (), BFloat64 *pgain* = BFloat64 ())**

## 9.33.3 Member Function Documentation

**9.33.3.1 BString Bds::Digitiser::getType ()**

**9.33.3.2 BError Bds::Digitiser::setMembers (BDictString & *members*)**

**9.33.3.3 BError Bds::Digitiser::getMembers (BDictString & *members*)**

## 9.33.4 Member Data Documentation

**9.33.4.1 BUInt32 Bds::Digitiser::id**

The ID.

**9.33.4.2 BTimeStamp Bds::Digitiser::startTime**

The Start Time.

**9.33.4.3 BTimeStamp Bds::Digitiser::endTime**

The End Time the channel was available.

**9.33.4.4 BString Bds::Digitiser::name**

The Digitisers name.

**9.33.4.5 BString Bds::Digitiser::type**

The Digitisers type.

**9.33.4.6 BString Bds::Digitiser::serialNumber**

The digitisers's serial number.

**9.33.4.7 BUInt32 Bds::Digitiser::numberChannels**

The number of supported channels.

**9.33.4.8 BFloat64 Bds::Digitiser::baseSamplingFrequency**

The base sampling frequency.

**9.33.4.9 BFloat64 Bds::Digitiser::initialSamplingFrequency**

The initial pre-decimation sampling frequency.

**9.33.4.10 BFloat64 Bds::Digitiser::gain**

The overall gain of the digitiser at the manufacturers calibration frequency. (For information only).

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

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

## 9.34 Bds::Fap Class Reference

This class defines a Amplitude/Phase [Response](#).

```
#include <BdsD.h>
```

### Public Member Functions

- [Fap](#) (BFloat64 pfrequency=BFloat64(), BFloat64 pamplitude=BFloat64(), BFloat64 pphase=BFloat64())

### Public Attributes

- BFloat64 [frequency](#)  
*The frequency.*
- BFloat64 [amplitude](#)  
*The Amplitude.*
- BFloat64 [phase](#)  
*The Phase.*

### 9.34.1 Detailed Description

This class defines a Amplitude/Phase [Response](#).

### 9.34.2 Constructor & Destructor Documentation

- 9.34.2.1 Bds::Fap::Fap (BFloat64 pfrequency = BFloat64(), BFloat64 pamplitude = BFloat64(), BFloat64 pphase = BFloat64())**

### 9.34.3 Member Data Documentation

#### 9.34.3.1 BFloat64 Bds::Fap::frequency

The frequency.

#### 9.34.3.2 BFloat64 Bds::Fap::amplitude

The Amplitude.

#### 9.34.3.3 BFloat64 Bds::Fap::phase

The Phase.

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

- [BdsD.h](#)

- [BdsD.cc](#)

## 9.35 Bds::FileDataHeaderTape\_1v1 Struct Reference

### Public Attributes

- [uint32\\_t type](#)  
*Type of file header.*
- [uint32\\_t headerLength](#)  
*Header length in bytes.*
- [uint32\\_t dataLength](#)  
*Data length in bytes.*
- [uint32\\_t dataType](#)  
*The type of a data sample.*
- [uint32\\_t numChannels](#)  
*The number of channels of data.*
- [uint32\\_t numSamples](#)  
*The number of samples of data per channel.*
- [BTimeStampMs startTime](#)  
*The time of the first sample.*
- [BTimeStampMs endTime](#)  
*The time of the last sample + 1.*
- [BTimeStampMs tapeLastVelaCode](#)  
*The last VELA code decoded.*
- [float tapeSignalLevelErrors](#)  
*The number of seconds the tape's FM signal was below a minimum level.*
- [float tapeSpeedVariationErrors](#)  
*The number of seconds the FM carrier was above a defined maximum deviation.*
- [float tapeVelaBitErrors](#)  
*The number of VELA timecode bit errors per second.*
- [float tapeFmSignalLevels](#) [24]  
*The FM signal levels for each channel.*

### 9.35.1 Member Data Documentation

#### 9.35.1.1 [uint32\\_t Bds::FileDataHeaderTape\\_1v1::type](#)

Type of file header.

**9.35.1.2 uint32\_t Bds::FileDataHeaderTape\_1v1::headerLength**

Header length in bytes.

**9.35.1.3 uint32\_t Bds::FileDataHeaderTape\_1v1::dataLength**

Data length in bytes.

**9.35.1.4 uint32\_t Bds::FileDataHeaderTape\_1v1::dataType**

The type of a data sample.

**9.35.1.5 uint32\_t Bds::FileDataHeaderTape\_1v1::numChannels**

The number of channels of data.

**9.35.1.6 uint32\_t Bds::FileDataHeaderTape\_1v1::numSamples**

The number of samples of data per channel.

**9.35.1.7 BTimeStampMs Bds::FileDataHeaderTape\_1v1::startTime**

The time of the first sample.

**9.35.1.8 BTimeStampMs Bds::FileDataHeaderTape\_1v1::endTime**

The time of the last sample + 1.

**9.35.1.9 BTimeStampMs Bds::FileDataHeaderTape\_1v1::tapeLastVelaCode**

The last VELA code decoded.

**9.35.1.10 float Bds::FileDataHeaderTape\_1v1::tapeSignalLevelErrors**

The number of seconds the tape's FM signal was below a minimum level.

**9.35.1.11 float Bds::FileDataHeaderTape\_1v1::tapeSpeedVariationErrors**

The number of seconds the FM carrier was above a defined maximum deviation.

**9.35.1.12 float Bds::FileDataHeaderTape\_1v1::tapeVelaBitErrors**

The number of VELA timecode bit errors per second.

**9.35.1.13 float Bds::FileDataHeaderTape\_1v1::tapeFmSignalLevels[24]**

The FM signal levels for each channel.

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

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp](#)

## 9.36 Bds::FileDataHeaderTape\_2v0 Struct Reference

### Public Attributes

- [uint32\\_t type](#)  
*Type of file header.*
- [uint32\\_t headerLength](#)  
*Header length in bytes.*
- [uint32\\_t dataLength](#)  
*Data length in bytes.*
- [uint32\\_t dataType](#)  
*The type of a data sample.*
- [uint32\\_t numChannels](#)  
*The number of channels of data.*
- [uint32\\_t numSamples](#)  
*The number of samples of data per channel.*
- [BTimeStampMs startTime](#)  
*The time of the first sample.*
- [BTimeStampMs endTime](#)  
*The time of the last sample + 1.*
- [BTimeStampMs tapeLastVelaCode](#)  
*The last VELA code decoded.*
- [float tapeSignalLevelErrors](#)  
*The number of seconds the tape's FM signal was below a minimum level.*
- [float tapeSpeedVariationErrors](#)  
*The number of seconds the FM carrier was above a defined maximum deviation.*
- [float tapeVelaBitErrors](#)  
*The number of VELA timecode bit errors per second.*
- [float tapeFmSignalLevels](#) [24]  
*The FM signal levels for each channel.*
- [uint32\\_t headerChecksum](#)  
*Checksum for header.*
- [uint32\\_t dataChecksum](#)  
*Checksum for data.*

## 9.36.1 Member Data Documentation

### 9.36.1.1 uint32\_t Bds::FileDataHeaderTape\_2v0::type

Type of file header.

### 9.36.1.2 uint32\_t Bds::FileDataHeaderTape\_2v0::headerLength

Header length in bytes.

### 9.36.1.3 uint32\_t Bds::FileDataHeaderTape\_2v0::dataLength

Data length in bytes.

### 9.36.1.4 uint32\_t Bds::FileDataHeaderTape\_2v0::dataType

The type of a data sample.

### 9.36.1.5 uint32\_t Bds::FileDataHeaderTape\_2v0::numChannels

The number of channels of data.

### 9.36.1.6 uint32\_t Bds::FileDataHeaderTape\_2v0::numSamples

The number of samples of data per channel.

### 9.36.1.7 BTimeStampMs Bds::FileDataHeaderTape\_2v0::startTime

The time of the first sample.

### 9.36.1.8 BTimeStampMs Bds::FileDataHeaderTape\_2v0::endTime

The time of the last sample + 1.

### 9.36.1.9 BTimeStampMs Bds::FileDataHeaderTape\_2v0::tapeLastVelaCode

The last VELA code decoded.

### 9.36.1.10 float Bds::FileDataHeaderTape\_2v0::tapeSignalLevelErrors

The number of seconds the tape's FM signal was below a minimum level.

### 9.36.1.11 float Bds::FileDataHeaderTape\_2v0::tapeSpeedVariationErrors

The number of seconds the FM carrier was above a defined maximum deviation.

**9.36.1.12 float Bds::FileDataHeaderTape\_2v0::tapeVelaBitErrors**

The number of VELA timecode bit errors per second.

**9.36.1.13 float Bds::FileDataHeaderTape\_2v0::tapeFmSignalLevels[24]**

The FM signal levels for each channel.

**9.36.1.14 uint32\_t Bds::FileDataHeaderTape\_2v0::headerChecksum**

Checksum for header.

**9.36.1.15 uint32\_t Bds::FileDataHeaderTape\_2v0::dataChecksum**

Checksum for data.

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

- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp](#)

## 9.37 Bds::Group Class Reference

This holds information on a groups.

```
#include <BdsD.h>
```

### Public Member Functions

- [Group](#) (BUInt32 pid=BUInt32(), BString pgroup=BString(), BString pdescription=BString())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*The unique id.*
- BString [group](#)  
*The Group name.*
- BString [description](#)  
*The Groups description.*

### 9.37.1 Detailed Description

This holds information on a groups.

### 9.37.2 Constructor & Destructor Documentation

**9.37.2.1 Bds::Group::Group (BUInt32 pid = BUInt32 (), BString pgroup = BString (), BString pdescription = BString ())**

### 9.37.3 Member Function Documentation

**9.37.3.1 BString Bds::Group::getType ()**

**9.37.3.2 BError Bds::Group::setMembers (BDictString & members)**

**9.37.3.3 BError Bds::Group::getMembers (BDictString & members)**

### 9.37.4 Member Data Documentation

**9.37.4.1 BUInt32 Bds::Group::id**

The unique id.

#### 9.37.4.2 BString Bds::Group::group

The [Group](#) name.

#### 9.37.4.3 BString Bds::Group::description

The Groups description.

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

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

## 9.38 Bds::ListRange Class Reference

This class defines a [TimePeriod](#).

```
#include <BdsD.h>
```

### Public Member Functions

- [ListRange](#) (BUInt32 pstart=BUInt32(), BUInt32 pnumber=BUInt32(), BInt32 preverse=BInt32())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [start](#)  
*The start position.*
- BUInt32 [number](#)  
*The number of items.*
- BInt32 [reverse](#)  
*List from end.*

### 9.38.1 Detailed Description

This class defines a [TimePeriod](#).

### 9.38.2 Constructor & Destructor Documentation

**9.38.2.1** `Bds::ListRange::ListRange (BUInt32 pstart = BUInt32 (), BUInt32 pnumber = BUInt32 (), BInt32 preverse = BInt32 ())`

### 9.38.3 Member Function Documentation

**9.38.3.1** `BString Bds::ListRange::getType ()`

**9.38.3.2** `BError Bds::ListRange::setMembers (BDictString & members)`

**9.38.3.3** `BError Bds::ListRange::getMembers (BDictString & members)`

### 9.38.4 Member Data Documentation

**9.38.4.1** `BUInt32 Bds::ListRange::start`

The start position.

#### 9.38.4.2 BUInt32 Bds::ListRange::number

The number of items.

#### 9.38.4.3 BInt32 Bds::ListRange::reverse

List from end.

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

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

## 9.39 Bds::Location Class Reference

This class defines a [Location](#). This will be the stations location.

```
#include <BdsD.h>
```

### Public Member Functions

- [Location](#) (BUInt32 pid=BUInt32(), BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BString pnetwork=BString(), BString pstation=BString(), BString pdatum=BString(), BFloat64 plongitude=BFloat64(), BFloat64 platitude=BFloat64(), BFloat64 pelevation=BFloat64(), BFloat64 parrayOffsetEast=BFloat64(), BFloat64 parrayOffsetNorth=BFloat64())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*The ID.*
- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time the channel was available.*
- BString [network](#)  
*The Network/Organisation Name.*
- BString [station](#)  
*The station this location is for.*
- BString [datum](#)  
*The locations Datum.*
- BFloat64 [longitude](#)  
*The longitude in degrees using the WGS84 datum.*
- BFloat64 [latitude](#)  
*The Latitude in degrees using the WGS84 datum.*
- BFloat64 [elevation](#)  
*The ground level elevation in meters from the WGS84 ellipsoid (Sea level).*
- BFloat64 [arrayOffsetEast](#)  
*The Array offset in in an array in an easterly direction.*
- BFloat64 [arrayOffsetNorth](#)  
*The Array offset in in an array in a northerly direction.*

### 9.39.1 Detailed Description

This class defines a [Location](#). This will be the stations location.

### 9.39.2 Constructor & Destructor Documentation

**9.39.2.1 Bds::Location::Location (BUInt32 *pid* = BUInt32 (), BTimeStamp *pstartTime* = BTimeStamp (), BTimeStamp *pendTime* = BTimeStamp (), BString *pnetwork* = BString (), BString *pstation* = BString (), BString *pdatum* = BString (), BFloat64 *plongitude* = BFloat64 (), BFloat64 *platitude* = BFloat64 (), BFloat64 *pelevation* = BFloat64 (), BFloat64 *parrayOffsetEast* = BFloat64 (), BFloat64 *parrayOffsetNorth* = BFloat64 ())**

### 9.39.3 Member Function Documentation

**9.39.3.1 BString Bds::Location::getType ()**

**9.39.3.2 BError Bds::Location::setMembers (BDictString & *members*)**

**9.39.3.3 BError Bds::Location::getMembers (BDictString & *members*)**

### 9.39.4 Member Data Documentation

**9.39.4.1 BUInt32 Bds::Location::id**

The ID.

**9.39.4.2 BTimeStamp Bds::Location::startTime**

The Start Time.

**9.39.4.3 BTimeStamp Bds::Location::endTime**

The End Time the channel was available.

**9.39.4.4 BString Bds::Location::network**

The Network/Organisation Name.

**9.39.4.5 BString Bds::Location::station**

The station this location is for.

**9.39.4.6 BString Bds::Location::datum**

The locations Datum.

**9.39.4.7 BFloat64 Bds::Location::longitude**

The longitude in degrees using the WGS84 datum.

**9.39.4.8 BFloat64 Bds::Location::latitude**

The Latitude in degrees using the WGS84 datum.

**9.39.4.9 BFloat64 Bds::Location::elevation**

The ground level elevation in meters from the WGS84 ellipsoid (Sea level).

**9.39.4.10 BFloat64 Bds::Location::arrayOffsetEast**

The Array offset in in an array in an easterly direction.

**9.39.4.11 BFloat64 Bds::Location::arrayOffsetNorth**

The Array offset in in an array in a northerly direction.

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

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

## 9.40 Bds::Network Class Reference

This class defines a [Network](#) organisation. Could have other information here like website, contacts etc.

```
#include <BdsD.h>
```

### Public Member Functions

- [Network](#) (BUInt32 pid=BUInt32(), BString pnetwork=BString(), BString pdescription=BString(), BList< BString > pstations=BList< BString >())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*ID.*
- BString [network](#)  
*The name.*
- BString [description](#)  
*The organisations description.*
- BList< BString > [stations](#)  
*The list of arrays/stations the [Network](#) uses.*

#### 9.40.1 Detailed Description

This class defines a [Network](#) organisation. Could have other information here like website, contacts etc.

## 9.40.2 Constructor & Destructor Documentation

**9.40.2.1** **Bds::Network::Network** (**BUInt32** *pid* = **BUInt32** (), **BString** *pnetwork* = **BString** (), **BString** *pdescription* = **BString** (), **BList**< **BString** > *pstations* = **BList**<**BString** > ())

## 9.40.3 Member Function Documentation

**9.40.3.1** **BString** **Bds::Network::getType** ()

**9.40.3.2** **BError** **Bds::Network::setMembers** (**BDictString** & *members*)

**9.40.3.3** **BError** **Bds::Network::getMembers** (**BDictString** & *members*)

## 9.40.4 Member Data Documentation

**9.40.4.1** **BUInt32** **Bds::Network::id**

ID.

**9.40.4.2** **BString** **Bds::Network::network**

The name.

**9.40.4.3** **BString** **Bds::Network::description**

The organisations description.

**9.40.4.4** **BList**<**BString** > **Bds::Network::stations**

The list of arrays/stations the [Network](#) uses.

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

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

## 9.41 Bds::Note Class Reference

This holds information on a [Note](#).

```
#include <BdsD.h>
```

### Public Member Functions

- [Note](#) (BUInt32 pid=BUInt32(), BTimeStamp ptime=BTimeStamp(), BString ptype=BString(), BString puser=BString(), BString ptitle=BString(), BString pdescription=BString())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*The unique id.*
- BTimeStamp [time](#)  
*The Time.*
- BString [type](#)  
*The Type.*
- BString [user](#)  
*The user.*
- BString [title](#)  
*The Changes title.*
- BString [description](#)  
*The Description of the change.*

#### 9.41.1 Detailed Description

This holds information on a [Note](#).

## 9.41.2 Constructor & Destructor Documentation

**9.41.2.1** **Bds::Note::Note** (**BUInt32** *pid* = **BUInt32** (), **BTimeStamp** *ptime* = **BTimeStamp** (), **BString** *pctype* = **BString** (), **BString** *puser* = **BString** (), **BString** *ptitle* = **BString** (), **BString** *pdescription* = **BString** ())

## 9.41.3 Member Function Documentation

**9.41.3.1** **BString** **Bds::Note::getType** ()

**9.41.3.2** **BError** **Bds::Note::setMembers** (**BDictString** & *members*)

**9.41.3.3** **BError** **Bds::Note::getMembers** (**BDictString** & *members*)

## 9.41.4 Member Data Documentation

**9.41.4.1** **BUInt32** **Bds::Note::id**

The unique id.

**9.41.4.2** **BTimeStamp** **Bds::Note::time**

The Time.

**9.41.4.3** **BString** **Bds::Note::type**

The Type.

**9.41.4.4** **BString** **Bds::Note::user**

The user.

**9.41.4.5** **BString** **Bds::Note::title**

The Changes title.

**9.41.4.6** **BString** **Bds::Note::description**

The Description of the change.

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

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

## 9.42 Bds::Point Class Reference

```
#include <BdsD.h>
```

### Public Member Functions

- [Point](#) (BFloat64 px=BFloat64(), BFloat64 py=BFloat64())

### Public Attributes

- BFloat64 [x](#)
- BFloat64 [y](#)

### 9.42.1 Constructor & Destructor Documentation

**9.42.1.1** [Bds::Point::Point](#) (BFloat64 *px* = BFloat64(), BFloat64 *py* = BFloat64())

### 9.42.2 Member Data Documentation

**9.42.2.1** BFloat64 [Bds::Point::x](#)

**9.42.2.2** BFloat64 [Bds::Point::y](#)

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

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

## 9.43 Bds::PoleZero Class Reference

This class defines a Pole/Zero [Response](#).

```
#include <BdsD.h>
```

### Public Member Functions

- [PoleZero](#) (BArray< [Point](#) > ppoles=BArray< [Point](#) >(), BArray< [Point](#) > pzeros=BArray< [Point](#) >())

### Public Attributes

- BArray< [Point](#) > [poles](#)  
*Poles.*
- BArray< [Point](#) > [zeros](#)  
*Zeros.*

### 9.43.1 Detailed Description

This class defines a Pole/Zero [Response](#).

### 9.43.2 Constructor & Destructor Documentation

- 9.43.2.1** [Bds::PoleZero::PoleZero](#) (BArray< [Point](#) > *ppoles* = BArray<[Point](#) >(), BArray< [Point](#) > *pzeros* = BArray<[Point](#) >())

### 9.43.3 Member Data Documentation

#### 9.43.3.1 BArray<Point > Bds::PoleZero::poles

Poles.

#### 9.43.3.2 BArray<Point > Bds::PoleZero::zeros

Zeros.

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

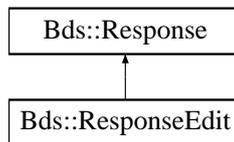
- [BdsD.h](#)
- [BdsD.cc](#)

## 9.44 Bds::Response Class Reference

This class defines a [Response](#).

```
#include <BdsD.h>
```

Inheritance diagram for Bds::Response::



### Public Member Functions

- [Response](#) (BUInt32 pid=BUInt32(), BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BString pnetwork=BString(), BString pstation=BString(), BString pchannel=BString(), BString pname=BString(), BString ptype=BString(), [PoleZero](#) ppoleZeros=[PoleZero](#)(), BArray< [Fap](#) > pfaps=BArray< [Fap](#) >(), BArray< BFloat64 > pfirCoefficients=BArray< BFloat64 >(), BFloat64 pgain=BFloat64(), BFloat64 pgainFrequency=BFloat64(), BFloat64 pdecimation=BFloat64(), BString psymmetry=BString())

### Public Attributes

- BUInt32 [id](#)  
*The ID.*
- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time the channel was available.*
- BString [network](#)  
*The Network/Organisation Name.*
- BString [station](#)  
*The station.*
- BString [channel](#)  
*The channel.*
- BString [name](#)  
*The response name. (Anti-Aliasing filter, [Digitiser](#), post filter etc).*
- BString [type](#)  
*The type of response ([PoleZero](#), [AmplitudePhase](#) or [FIR Coefficients](#)).*
- [PoleZero](#) [poleZeros](#)

*PoleZero, AmplitudePhase or FIR Coefficient data.*

- `BArray< Fap > faps`  
*The Frequency/Amplitude/Phase table.*
- `BArray< BFloat64 > firCoefficients`  
*The FIR filters coefficients.*
- `BFloat64 gain`  
*Overall gain at gain frequency. (For information) (Do we need this ?).*
- `BFloat64 gainFrequency`  
*Frequency that gain is valid for. (For information) (Do we need this ?).*
- `BFloat64 decimation`  
*Decimation performed post filter.*
- `BString symmetry`  
*Symmetry for FIR coefficients (A = asymmetric, B = symmetric[odd], C = symmetric[even]) ??*

### 9.44.1 Detailed Description

This class defines a [Response](#).

### 9.44.2 Constructor & Destructor Documentation

**9.44.2.1 Bds::Response::Response (BUInt32 pid = BUInt32 (), BTimeStamp pstartTime = BTimeStamp (), BTimeStamp pendTime = BTimeStamp (), BString pnetwork = BString (), BString pstation = BString (), BString pchannel = BString (), BString pname = BString (), BString ptype = BString (), PoleZero ppoleZeros = PoleZero (), BArray< Fap > pfaps = BArray< Fap > (), BArray< BFloat64 > pfirCoefficients = BArray< BFloat64 > (), BFloat64 pgain = BFloat64 (), BFloat64 pgainFrequency = BFloat64 (), BFloat64 pdecimation = BFloat64 (), BString psymmetry = BString ())**

### 9.44.3 Member Data Documentation

#### 9.44.3.1 BUInt32 Bds::Response::id

The ID.

#### 9.44.3.2 BTimeStamp Bds::Response::startTime

The Start Time.

#### 9.44.3.3 BTimeStamp Bds::Response::endTime

The End Time the channel was available.

#### 9.44.3.4 BString Bds::Response::network

The Network/Organisation Name.

#### 9.44.3.5 BString Bds::Response::station

The station.

#### 9.44.3.6 BString Bds::Response::channel

The channel.

#### 9.44.3.7 BString Bds::Response::name

The response name. (Anti-Aliasing filter, [Digitiser](#), post filter etc).

#### 9.44.3.8 BString Bds::Response::type

The type of response ([PoleZero](#), AmplitudePhase or FIR Coefficients).

#### 9.44.3.9 PoleZero Bds::Response::poleZeros

[PoleZero](#), AmplitudePhase or FIR Coefficient data.

#### 9.44.3.10 BArray<Fap > Bds::Response::faps

The Frequency/Amplitude/Phase table.

#### 9.44.3.11 BArray<BFloat64 > Bds::Response::firCoefficients

The FIR filters coefficients.

#### 9.44.3.12 BFloat64 Bds::Response::gain

Overall gain at gain frequency. (For information) (Do we need this ?).

#### 9.44.3.13 BFloat64 Bds::Response::gainFrequency

Frequency that gain is valid for. (For information) (Do we need this ?).

#### 9.44.3.14 BFloat64 Bds::Response::decimation

Decimation performed post filter.

**9.44.3.15 BString Bds::Response::symmetry**

Symmetry for FIR coefficients (A = asymmetric, B = symmetric[odd], C = symmetric[even]) ??

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

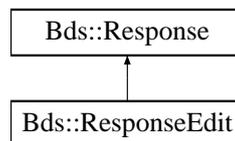
- [BdsD.h](#)
- [BdsD.cc](#)

## 9.45 Bds::ResponseEdit Class Reference

[Response](#) Editor object.

```
#include <BdsLib.h>
```

Inheritance diagram for Bds::ResponseEdit::



### Public Member Functions

- [ResponseEdit](#) (const [Response](#) &response)
- [~ResponseEdit](#) ()
- BString [getString](#) ()
- void [setString](#) (BString str)

### 9.45.1 Detailed Description

[Response](#) Editor object.

### 9.45.2 Constructor & Destructor Documentation

**9.45.2.1** Bds::ResponseEdit::ResponseEdit (const [Response](#) & *response*)

**9.45.2.2** Bds::ResponseEdit::~~ResponseEdit ()

### 9.45.3 Member Function Documentation

**9.45.3.1** BString Bds::ResponseEdit::getString ()

**9.45.3.2** void Bds::ResponseEdit::setString (BString *str*)

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

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

## 9.46 Bds::Selection Class Reference

```
#include <BdsD.h>
```

### Public Member Functions

- [Selection](#) ([ListRange](#) prange=[ListRange](#)(), [BTimeStamp](#) pstartTime=[BTimeStamp](#)(), [BTimeStamp](#) pendTime=[BTimeStamp](#)(), [BList](#)< [SelectionChannel](#) > pchannels=[BList](#)< [SelectionChannel](#) >(), [BUInt32](#) pchannelId=[BUInt32](#)(), [BUInt32](#) pdigitiserId=[BUInt32](#)(), [BUInt32](#) psensorId=[BUInt32](#)(), [BUInt32](#) psensorOldId=[BUInt32](#)())

### Public Attributes

- [ListRange](#) range  
*The range of data to return.*
- [BTimeStamp](#) startTime  
*The Start Time.*
- [BTimeStamp](#) endTime  
*The End Time.*
- [BList](#)< [SelectionChannel](#) > channels  
*The data channels to select.*
- [BUInt32](#) channelId  
*The Channel id.*
- [BUInt32](#) digitiserId  
*The Digitiser id.*
- [BUInt32](#) sensorId  
*The Sensor id.*
- [BUInt32](#) sensorOldId  
*The Sensor old id.*

## 9.46.1 Constructor & Destructor Documentation

**9.46.1.1** **Bds::Selection::Selection** (*ListRange prange* = ListRange (), *BTimeStamp pstartTime* = BTimeStamp (), *BTimeStamp pendTime* = BTimeStamp (), *BList< SelectionChannel > pchannels* = BList< SelectionChannel > (), *BUInt32 pchannelId* = BUInt32 (), *BUInt32 pdigitiserId* = BUInt32 (), *BUInt32 psensorId* = BUInt32 (), *BUInt32 psensorOldId* = BUInt32 ())

## 9.46.2 Member Data Documentation

### 9.46.2.1 ListRange Bds::Selection::range

The range of data to return.

### 9.46.2.2 BTimeStamp Bds::Selection::startTime

The Start Time.

### 9.46.2.3 BTimeStamp Bds::Selection::endTime

The End Time.

### 9.46.2.4 BList< SelectionChannel > Bds::Selection::channels

The data channels to select.

### 9.46.2.5 BUInt32 Bds::Selection::channelId

The [Channel](#) id.

### 9.46.2.6 BUInt32 Bds::Selection::digitiserId

The [Digitiser](#) id.

### 9.46.2.7 BUInt32 Bds::Selection::sensorId

The [Sensor](#) id.

### 9.46.2.8 BUInt32 Bds::Selection::sensorOldId

The [Sensor](#) old id.

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

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

## 9.47 Bds::SelectionChannel Class Reference

```
#include <BdsD.h>
```

### Public Member Functions

- [SelectionChannel](#) (BString pnetwork=BString(), BString pstation=BString(), BString pchannel=BString(), BString psource=BString())

### Public Attributes

- BString [network](#)
- BString [station](#)
- BString [channel](#)
- BString [source](#)

### 9.47.1 Constructor & Destructor Documentation

**9.47.1.1** [Bds::SelectionChannel::SelectionChannel](#) (BString *pnetwork* = BString(), BString *pstation* = BString(), BString *pchannel* = BString(), BString *psource* = BString())

### 9.47.2 Member Data Documentation

**9.47.2.1** BString [Bds::SelectionChannel::network](#)

**9.47.2.2** BString [Bds::SelectionChannel::station](#)

**9.47.2.3** BString [Bds::SelectionChannel::channel](#)

**9.47.2.4** BString [Bds::SelectionChannel::source](#)

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

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

## 9.48 Bds::SelectionInfo Class Reference

This class defines the set of data selection criteria.

```
#include <BdsD.h>
```

### Public Member Functions

- [SelectionInfo](#) (BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BList< BString > pnetworks=BList< BString >(), BList< BString > pstations=BList< BString >(), BList< BString > pchannels=BList< BString >(), BList< BString > psources=BList< BString >(), BUInt32 pnumDataChannels=BUInt32())

### Public Attributes

- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time.*
- BList< BString > [networks](#)  
*The list of [Network](#) Names.*
- BList< BString > [stations](#)  
*The list of Array or [Station](#) names.*
- BList< BString > [channels](#)  
*The list of Channels.*
- BList< BString > [sources](#)  
*The list of Data Sources.*
- BUInt32 [numDataChannels](#)  
*The number of sets of data in the system.*

### 9.48.1 Detailed Description

This class defines the set of data selection criteria.

## 9.48.2 Constructor & Destructor Documentation

**9.48.2.1 Bds::SelectionInfo::SelectionInfo** (*BTimeStamp pstartTime* = BTimeStamp(), *BTimeStamp pendTime* = BTimeStamp(), *BList< BString > pnetworks* = BList<BString >(), *BList< BString > pstations* = BList<BString >(), *BList< BString > pchannels* = BList<BString >(), *BList< BString > psources* = BList<BString >(), *BUInt32 pnumDataChannels* = BUInt32())

## 9.48.3 Member Data Documentation

### 9.48.3.1 BTimeStamp Bds::SelectionInfo::startTime

The Start Time.

### 9.48.3.2 BTimeStamp Bds::SelectionInfo::endTime

The End Time.

### 9.48.3.3 BList<BString > Bds::SelectionInfo::networks

The list of [Network](#) Names.

### 9.48.3.4 BList<BString > Bds::SelectionInfo::stations

The list of Array or [Station](#) names.

### 9.48.3.5 BList<BString > Bds::SelectionInfo::channels

The list of Channels.

### 9.48.3.6 BList<BString > Bds::SelectionInfo::sources

The list of Data Sources.

### 9.48.3.7 BUInt32 Bds::SelectionInfo::numDataChannels

The number of sets of data in the system.

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

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

## 9.49 Bds::Sensor Class Reference

This class defines a [Sensor](#).

```
#include <BdsD.h>
```

### Public Member Functions

- [Sensor](#) (BUInt32 pid=BUInt32(), BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BString pname=BString(), BString ptype=BString(), BString pserialNumber=BString(), BUInt32 pnumberChannels=BUInt32(), BString pgainUnits=BString(), BFloat64 pgain=BFloat64(), BUInt32 poldId=BUInt32())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*The ID.*
- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time.*
- BString [name](#)  
*The Sensors name.*
- BString [type](#)  
*The type of sensor. (Seismometer, Hydrophone etc).*
- BString [serialNumber](#)  
*The sensor's serial number. Only used when there is a unique physical sensor.*
- BUInt32 [numberChannels](#)  
*The number of supported channels.*
- BString [gainUnits](#)  
*The gain units.*
- BFloat64 [gain](#)  
*The overall gain of the sensor at the manufacturers calibration frequency. (For information only).*
- BUInt32 [oldId](#)  
*The Id from the old Autodrm database.*

## 9.49.1 Detailed Description

This class defines a [Sensor](#).

## 9.49.2 Constructor & Destructor Documentation

**9.49.2.1 Bds::Sensor::Sensor (BUInt32 *pid* = BUInt32 (), BTimeStamp *pstartTime* = BTimeStamp (), BTimeStamp *pendTime* = BTimeStamp (), BString *pname* = BString (), BString *pctype* = BString (), BString *pserialNumber* = BString (), BUInt32 *pnumberChannels* = BUInt32 (), BString *pgainUnits* = BString (), BFloat64 *pgain* = BFloat64 (), BUInt32 *poldId* = BUInt32 ())**

## 9.49.3 Member Function Documentation

**9.49.3.1 BString Bds::Sensor::getType ()**

**9.49.3.2 BError Bds::Sensor::setMembers (BDictString & *members*)**

**9.49.3.3 BError Bds::Sensor::getMembers (BDictString & *members*)**

## 9.49.4 Member Data Documentation

**9.49.4.1 BUInt32 Bds::Sensor::id**

The ID.

**9.49.4.2 BTimeStamp Bds::Sensor::startTime**

The Start Time.

**9.49.4.3 BTimeStamp Bds::Sensor::endTime**

The End Time.

**9.49.4.4 BString Bds::Sensor::name**

The Sensors name.

**9.49.4.5 BString Bds::Sensor::type**

The type of sensor. (Seismometer, Hydrophone etc).

**9.49.4.6 BString Bds::Sensor::serialNumber**

The sensor's serial number. Only used when there is a unique physical sensor.

**9.49.4.7 BUInt32 Bds::Sensor::numberChannels**

The number of supported channels.

**9.49.4.8 BString Bds::Sensor::gainUnits**

The gain units.

**9.49.4.9 BFloat64 Bds::Sensor::gain**

The overall gain of the sensor at the manufacturers calibration frequency. (For information only).

**9.49.4.10 BUInt32 Bds::Sensor::oldId**

The Id from the old Autodrm database.

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

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

## 9.50 Bds::SensorLocation Class Reference

This class defines a [Sensor](#) location.

```
#include <BdsD.h>
```

### Public Member Functions

- [SensorLocation](#) (BUInt32 pid=BUInt32(), BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp(), BUInt32 psensorId=BUInt32(), BFloat64 pdepth=BFloat64(), BFloat64 phorizontalAngle=BFloat64(), BFloat64 pverticalAngle=BFloat64())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*The ID.*
- BTimeStamp [startTime](#)  
*The Start Time.*
- BTimeStamp [endTime](#)  
*The End Time.*
- BUInt32 [sensorId](#)  
*The sensor ID.*
- BFloat64 [depth](#)  
*The depth of the sensor below ground level in meters.*
- BFloat64 [horizontalAngle](#)  
*The Sensors placement horizontal angle in degrees clockwise from north.*
- BFloat64 [verticalAngle](#)  
*The Sensors placement vertical angle in degrees degrees with zero = vertically up.*

#### 9.50.1 Detailed Description

This class defines a [Sensor](#) location.

## 9.50.2 Constructor & Destructor Documentation

**9.50.2.1** `Bds::SensorLocation::SensorLocation (BUInt32 pid = BUInt32 (), BTimeStamp pstartTime = BTimeStamp (), BTimeStamp pendTime = BTimeStamp (), BUInt32 psensorId = BUInt32 (), BFloat64 pdepth = BFloat64 (), BFloat64 phorizontalAngle = BFloat64 (), BFloat64 pverticalAngle = BFloat64 ())`

## 9.50.3 Member Function Documentation

**9.50.3.1** `BString Bds::SensorLocation::getType ()`

**9.50.3.2** `BError Bds::SensorLocation::setMembers (BDictString & members)`

**9.50.3.3** `BError Bds::SensorLocation::getMembers (BDictString & members)`

## 9.50.4 Member Data Documentation

**9.50.4.1** `BUInt32 Bds::SensorLocation::id`

The ID.

**9.50.4.2** `BTimeStamp Bds::SensorLocation::startTime`

The Start Time.

**9.50.4.3** `BTimeStamp Bds::SensorLocation::endTime`

The End Time.

**9.50.4.4** `BUInt32 Bds::SensorLocation::sensorId`

The sensor ID.

**9.50.4.5** `BFloat64 Bds::SensorLocation::depth`

The depth of the sensor below ground level in meters.

**9.50.4.6** `BFloat64 Bds::SensorLocation::horizontalAngle`

The Sensors placement horizontal angle in degrees clockwise from north.

**9.50.4.7** `BFloat64 Bds::SensorLocation::verticalAngle`

The Sensors placement vertical angle in degrees degrees with zero = vertically up.

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

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

## 9.51 Bds::Station Class Reference

```
#include <BdsD.h>
```

### Public Member Functions

- **Station** (BUInt32 pid=BUInt32(), BString pname=BString(), BString ptype=BString(), BString pdescription=BString(), BList< [ArrayChannel](#) > pchannels=BList< [ArrayChannel](#) >())

### Public Attributes

- BUInt32 **id**  
*ID.*
- BString **name**  
*The name.*
- BString **type**  
*Array or Station.*
- BString **description**  
*Description.*
- BList< [ArrayChannel](#) > **channels**  
*List of channels if an Array.*

### 9.51.1 Constructor & Destructor Documentation

- 9.51.1.1 Bds::Station::Station (BUInt32 pid = BUInt32(), BString pname = BString(), BString ptype = BString(), BString pdescription = BString(), BList< [ArrayChannel](#) > pchannels = BList< [ArrayChannel](#) >())**

### 9.51.2 Member Data Documentation

#### 9.51.2.1 BUInt32 Bds::Station::id

ID.

#### 9.51.2.2 BString Bds::Station::name

The name.

#### 9.51.2.3 BString Bds::Station::type

Array or [Station](#).

#### 9.51.2.4 BString Bds::Station::description

Description.

#### 9.51.2.5 BList<ArrayChannel > Bds::Station::channels

List of channels if an Array.

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

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

## 9.52 Bds::TimePeriod Class Reference

```
#include <BdsD.h>
```

### Public Member Functions

- [TimePeriod](#) (BTimeStamp pstartTime=BTimeStamp(), BTimeStamp pendTime=BTimeStamp())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BTimeStamp [startTime](#)  
*The Start time to the nearest us.*
- BTimeStamp [endTime](#)  
*The End time to the nearest us.*

### 9.52.1 Constructor & Destructor Documentation

**9.52.1.1** [Bds::TimePeriod::TimePeriod](#) (BTimeStamp *pstartTime* = BTimeStamp(), BTimeStamp *pendTime* = BTimeStamp())

### 9.52.2 Member Function Documentation

**9.52.2.1** [BString Bds::TimePeriod::getType](#) ()

**9.52.2.2** [BError Bds::TimePeriod::setMembers](#) (BDictString & *members*)

**9.52.2.3** [BError Bds::TimePeriod::getMembers](#) (BDictString & *members*)

### 9.52.3 Member Data Documentation

**9.52.3.1** [BTimeStamp Bds::TimePeriod::startTime](#)

The Start time to the nearest us.

**9.52.3.2** [BTimeStamp Bds::TimePeriod::endTime](#)

The End time to the nearest us.

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

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

## 9.53 Bds::User Class Reference

This holds information on a user.

```
#include <BdsD.h>
```

### Public Member Functions

- [User](#) (BUInt32 pid=BUInt32(), BString puser=BString(), BString ppassword=BString(), BString pname=BString(), BString pemail=BString(), BString ptelephone=BString(), BString pad-  
dress=BString(), BInt32 penabled=BInt32(), BList< BString > pgroups=BList< BString >())
- BString [getType](#) ()
- BError [setMembers](#) (BDictString &members)
- BError [getMembers](#) (BDictString &members)

### Public Attributes

- BUInt32 [id](#)  
*The unique user ID.*
- BString [user](#)  
*The User ID.*
- BString [password](#)  
*The Users password.*
- BString [name](#)  
*The Users full name.*
- BString [email](#)  
*The users email Address.*
- BString [telephone](#)  
*The Users telephone number.*
- BString [address](#)  
*The Users postal address.*
- BInt32 [enabled](#)  
*Whether the users account is enabled.*
- BList< BString > [groups](#)  
*The security groups the user belongs to.*

#### 9.53.1 Detailed Description

This holds information on a user.

## 9.53.2 Constructor & Destructor Documentation

**9.53.2.1** **Bds::User::User** (**BUInt32** *pid* = **BUInt32** (), **BString** *puser* = **BString** (), **BString** *ppassword* = **BString** (), **BString** *pname* = **BString** (), **BString** *pemail* = **BString** (), **BString** *ptelephone* = **BString** (), **BString** *paddress* = **BString** (), **BInt32** *penabled* = **BInt32** (), **BList**< **BString** > *pgroups* = **BList**<**BString** > ())

## 9.53.3 Member Function Documentation

**9.53.3.1** **BString** **Bds::User::getType** ()

**9.53.3.2** **BError** **Bds::User::setMembers** (**BDictString** & *members*)

**9.53.3.3** **BError** **Bds::User::getMembers** (**BDictString** & *members*)

## 9.53.4 Member Data Documentation

**9.53.4.1** **BUInt32** **Bds::User::id**

The unique user ID.

**9.53.4.2** **BString** **Bds::User::user**

The [User](#) ID.

**9.53.4.3** **BString** **Bds::User::password**

The Users password.

**9.53.4.4** **BString** **Bds::User::name**

The Users full name.

**9.53.4.5** **BString** **Bds::User::email**

The users email Address.

**9.53.4.6** **BString** **Bds::User::telephone**

The Users telephone number.

**9.53.4.7** **BString** **Bds::User::address**

The Users postal address.

**9.53.4.8** **BInt32** **Bds::User::enabled**

Whether the users account is enabled.

#### 9.53.4.9 `BList<BString> Bds::User::groups`

The security groups the user belongs to.

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

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

# Chapter 10

## File Documentation

### 10.1 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp File Reference

```
#include <BdsCompress.h>
```

#### Namespaces

- namespace [Bds](#)

#### Functions

- BError [Bds::bdsUnCompressCm8](#) (BUInt8 \*buffer, BUInt n, BArray< BInt32 > &data)

## 10.2 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.h File Reference

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

### Namespaces

- namespace [Bds](#)

### Functions

- BError [Bds::bdsUnCompressCm8](#) (BUInt8 \*buffer, BUInt n, BArray< BInt32 > &data)

## 10.3 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp File Reference

```
#include <BdsDataCollate.h>
```

### Namespaces

- namespace [Bds](#)

## 10.4 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h File Reference

```
#include <BString.h>
#include <BFile.h>
#include <BTimeStamp.h>
#include <BdsD.h>
#include <BdsDataFile.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataCollate](#)

*This class defines the interface for generic data file access.*

## 10.5 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp File Reference

```
#include <BdsDataFile.h>
```

### Namespaces

- namespace [Bds](#)

## 10.6 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h File Reference

```
#include <BString.h>
#include <BFile.h>
#include <BTimeStamp.h>
#include <BdsD.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFile](#)  
*This class defines the interface for generic data file access.*

## 10.7 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp File Reference

```
#include <BdsDataFileAscii.h>  
#include <BTimeStamp.h>
```

### Namespaces

- namespace [Bds](#)

## 10.8 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h **File Reference**

```
#include <BdsDataFile.h>
#include <BdsD.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFileAscii](#)

## 10.9 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs-1.cpp File Reference

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

### Namespaces

- namespace [Bds](#)

## 10.10 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs-1.h File Reference

```
#include <BdsDataFile.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFileBdrs](#)

## 10.11 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp File Reference

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

### Namespaces

- namespace [Bds](#)

## 10.12 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h File Reference

```
#include <BdsDataFile.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFileBdrs](#)

## 10.13 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <errno.h>
#include <byteswap.h>
#include <BdsLib.h>
#include <BdsDataLib.h>
#include <BdsDataFileBds.h>
#include <BBuffer.h>
#include <zlib.h>
#include <canada_compress.h>
```

### Namespaces

- namespace [Bds](#)

### Defines

- #define [LDEBUG](#) 0
- #define [dprintf](#)(fmt, a...)

### Functions

- BUInt32 [Bds::crc](#) (BUInt32 crc, void \*data, int numBytes)

### Variables

- const BString [Bds::BdsVersion](#) = "1.0.0"

### 10.13.1 Define Documentation

**10.13.1.1** [#define dprintf](#)(fmt, a...)

**10.13.1.2** [#define LDEBUG](#) 0

## 10.14 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h File Reference

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

### Namespaces

- namespace [Bds](#)

### Classes

- struct [Bds::BdsDataBlockHeader](#)
- struct [Bds::BdsDataBlock](#)
- struct [Bds::BdsDataPacketHeader](#)
- class [Bds::BdsDataPacket](#)
- class [Bds::BdsDataStreamlet](#)
- class [Bds::DataFileBds](#)

*This class implements the BDS Data File/Stream access system.*

### Defines

- #define [BdsDataFileBds\\_H](#) 1

### Enumerations

- enum [Bds::BdsDataType](#) { [Bds::BdsDataTypeBlock](#) = 0x42534442, [Bds::BdsDataTypeInfo](#) = 0x30534442, [Bds::BdsDataTypeData](#) = 0x31534442 }

#### 10.14.1 Define Documentation

##### 10.14.1.1 #define BdsDataFileBds\_H 1

## 10.15 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp File Reference

```
#include <BdsDataFileBknas.h>  
#include <BTimeStamp.h>
```

### Namespaces

- namespace [Bds](#)

## 10.16 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h File Reference

```
#include <BdsDataFile.h>
#include <BdsD.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFileBknas](#)

## 10.17 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp File Reference

```
#include <BdsDataFileGcf.h>
#include <arpa/inet.h>
#include <errno.h>
#include <gcf2.h>
```

### Namespaces

- namespace [Bds](#)

### Defines

- #define [DEBUG 0](#)

#### 10.17.1 Define Documentation

##### 10.17.1.1 #define [DEBUG 0](#)

## 10.18 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h **File Reference**

```
#include <BdsDataFile.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFileGcf](#)

## 10.19 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp File Reference

```
#include <BdsDataFileIms.h>
#include <BTimeStamp.h>
```

### Namespaces

- namespace [Bds](#)

### Functions

- void [Bds::dataCalculateDifference](#) (BInt32 &prevValue, BArray< BInt32 > &data)
- BInt32 [Bds::dataChecksum](#) (BInt32 checksum, BArray< BInt32 > &data)
- BError [Bds::dataCompressCm6](#) (int &prevValue1, int &prevValue2, BArray< BInt32 > &data, BString &d)
- void [Bds::dataConvert](#) (BArray< BFloat64 > &dataIn, BArray< BInt32 > &dataOut)

### Variables

- static char [Bds::cm6Table](#) [64]

## 10.20 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIm.s.h File Reference

```
#include <BdsDataFile.h>
#include <BdsD.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFileIm.s](#)

## 10.21 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <errno.h>
#include <BdsDataFileTapeDigitiser.h>
#include <BTimeStampMs.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- struct [Bds::FileDataHeaderTape\\_1v1](#)
- struct [Bds::FileDataHeaderTape\\_2v0](#)

### Enumerations

- enum [Bds::FileHeaderType](#) { [Bds::FileHeaderType\\_Standard](#) = 1, [Bds::FileHeaderType\\_-TapeDigitiser](#) = 10 }
- enum [Bds::FileSampleType](#) {  
[Bds::FileSampleType\\_Unknown](#), [Bds::FileSampleType\\_Float32](#), [Bds::FileSampleType\\_Float64](#),  
[Bds::FileSampleType\\_Int16](#),  
[Bds::FileSampleType\\_Int32](#) }

### Variables

- const double [Bds::Scale](#) = 16777216.0

## 10.22 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BFile.h>
#include <BEntry.h>
#include <BBuffer.h>
#include <BDict.h>
#include <BdsDataFile.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFileTapeDigitiser](#)

*This class implements the TapeDigitiser's file output conversion and storing system.*

### Defines

- #define [BdsDataFileTapeDigitiser\\_H](#) 1

#### 10.22.1 Define Documentation

##### 10.22.1.1 #define BdsDataFileTapeDigitiser\_H 1

## 10.23 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra-1.cpp File Reference

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

### Namespaces

- namespace [Bds](#)

## 10.24 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra-1.h File Reference

```
#include <BdsDataFile.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFileWra](#)

## 10.25 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp File Reference

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

### Namespaces

- namespace [Bds](#)

## 10.26 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h File Reference

```
#include <BdsDataFile.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFileWra](#)

## 10.27 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp File Reference

```
#include <BdsDataFileWraAgso.h>
#include <BdsCompress.h>
#include <BDebug.h>
#include <errno.h>
#include <math.h>
```

### Namespaces

- namespace [Bds](#)

### Functions

- static `BList< BString >` [parseStringFixedFields](#) (`BString s`, `int *fieldWidths`)

#### 10.27.1 Function Documentation

**10.27.1.1** `static BList<BString> parseStringFixedFields (BString s, int * fieldWidths)`  
[static]

## 10.28 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h File Reference

```
#include <BdsDataFile.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFileWraAgso](#)

## 10.29 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp File Reference

```
#include <BdsDataLib.h>
#include <BdsDataFileAscii.h>
#include <BdsDataFileBknas.h>
#include <BdsDataFileBdrs.h>
#include <BdsDataFileBds.h>
#include <BdsDataFileGcf.h>
#include <BdsDataFileIms.h>
#include <BdsDataFileTapeDigitiser.h>
#include <BdsDataFileWra.h>
#include <BdsDataFileWraAgso.h>
```

### Namespaces

- namespace [Bds](#)

### Variables

- DataFormats [Bds::dataFormats](#)

## 10.30 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h File Reference

```
#include <BdsDataFile.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataFormats](#)  
*This class defines the interface for generic data file access.*

## 10.31 /src/blacknest/bds/bds/bdsDataLib/canada\_compress.c File Reference

```
#include <stdlib.h>
#include "canada_compress.h"
```

### Defines

- #define `htonl(a)` (`a`)
- #define `ntohl(a)` (`a`)

### Functions

- static void `pack` (unsigned long `m`, unsigned long `*y`, unsigned char `*b`, int `*j`)
- static void `unpack` (uint32\_t `m`, uint32\_t `*y`, unsigned char `*b`, int `*j`)
- int `canada_compress` (unsigned char `*b`, unsigned long `*y`, int `*n`, int `m`, unsigned long `*v0`)
- int `canada_uncompress` (unsigned char `*b`, uint32\_t `*y`, int `*n`, int `samples`, uint32\_t `*v0`)

### Variables

- static int `corrupt` = 0

#### 10.31.1 Define Documentation

10.31.1.1 #define `htonl(a)` (`a`)

10.31.1.2 #define `ntohl(a)` (`a`)

#### 10.31.2 Function Documentation

10.31.2.1 int `canada_compress` (unsigned char `*b`, unsigned long `*y`, int `*n`, int `m`, unsigned long `*v0`)

10.31.2.2 int `canada_uncompress` (unsigned char `*b`, uint32\_t `*y`, int `*n`, int `samples`, uint32\_t `*v0`)

10.31.2.3 static void `pack` (unsigned long `m`, unsigned long `*y`, unsigned char `*b`, int `*j`)  
[static]

10.31.2.4 static void `unpack` (uint32\_t `m`, uint32\_t `*y`, unsigned char `*b`, int `*j`) [static]

#### 10.31.3 Variable Documentation

10.31.3.1 int `corrupt` = 0 [static]

## 10.32 /src/blacknest/bds/bds/bdsDataLib/canada\_compress.h File Reference

```
#include <arpa/inet.h>
```

### Defines

- #define `CANCOMP_ERR` -1
- #define `CANCOMP_SUCCESS` 0
- #define `CANCOMP_NOT_20` 1
- #define `CANCOMP_CORRUPT` 2
- #define `CANCOMP_EXCEED` 3

### Functions

- int `canada_uncompress` (unsigned char \**b*, uint32\_t \**y*, int \**n*, int *m*, uint32\_t \**v0*)
- int `canada_compress` (unsigned char \**b*, unsigned long \**y*, int \**n*, int *m*, unsigned long \**v0*)

#### 10.32.1 Define Documentation

10.32.1.1 #define `CANCOMP_CORRUPT` 2

10.32.1.2 #define `CANCOMP_ERR` -1

10.32.1.3 #define `CANCOMP_EXCEED` 3

10.32.1.4 #define `CANCOMP_NOT_20` 1

10.32.1.5 #define `CANCOMP_SUCCESS` 0

#### 10.32.2 Function Documentation

10.32.2.1 int `canada_compress` (unsigned char \* *b*, unsigned long \* *y*, int \* *n*, int *m*, unsigned long \* *v0*)

10.32.2.2 int `canada_uncompress` (unsigned char \* *b*, uint32\_t \* *y*, int \* *n*, int *m*, uint32\_t \* *v0*)

## 10.33 BdsC.cc File Reference

```
#include <BdsC.h>
```

### Namespaces

- namespace [Bds](#)

## 10.34 BdsC.h File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <Boap.h>
#include <BString.h>
#include <BList.h>
#include <BArray.h>
#include <BdsD.h>
```

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::DataAccess](#)
- class [Bds::DataAddAccess](#)  
*This is the DataAdd Access API interface.*
- class [Bds::AdminAccess](#)  
*This is the AdminAccess API interface.*

### Defines

- #define [BDSC\\_H](#) 1

### Variables

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

#### 10.34.1 Define Documentation

##### 10.34.1.1 #define BDSC\_H 1

## 10.35 BdsD.cc File Reference

```
#include <BdsD.h>
```

### Namespaces

- namespace [Bds](#)

## 10.36 BdsD.h File Reference

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

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::Point](#)
- class [Bds::TimePeriod](#)
- class [Bds::ListRange](#)  
*This class defines a [TimePeriod](#).*
- class [Bds::Network](#)  
*This class defines a [Network](#) organisation. Could have other information here like website, contacts etc.*
- class [Bds::ArrayChannel](#)  
*This class defines an arrays channel.*
- class [Bds::Station](#)
- class [Bds::Location](#)  
*This class defines a [Location](#). This will be the stations location.*
- class [Bds::PoleZero](#)  
*This class defines a [Pole/Zero Response](#).*
- class [Bds::Fap](#)  
*This class defines a [Amplitude/Phase Response](#).*
- class [Bds::Response](#)  
*This class defines a [Response](#).*
- class [Bds::Calibration](#)  
*This class defines a calibration setting.*
- class [Bds::Digitiser](#)  
*This class defines a [Digitiser](#).*
- class [Bds::Sensor](#)  
*This class defines a [Sensor](#).*
- class [Bds::SensorLocation](#)  
*This class defines a [Sensor location](#).*

- class [Bds::ChannelInstrument](#)
- class [Bds::Channel](#)  
*This class defines a [Channel](#).*
- class [Bds::SelectionInfo](#)  
*This class defines the set of data selection criteria.*
- class [Bds::SelectionChannel](#)
- class [Bds::Selection](#)
- class [Bds::ChannelInfo](#)
- class [Bds::DataChannel](#)  
*This class defines information on a single channels set of data.*
- class [Bds::DataInfo](#)
- class [Bds::DataHandle](#)  
*This defines a handle to a data stream/file.*
- class [Bds::DataBlock](#)
- class [Bds::User](#)  
*This holds information on a user.*
- class [Bds::Group](#)  
*This holds information on a groups.*
- class [Bds::AccessGroup](#)  
*This holds information on data access groups.*
- class [Bds::Change](#)  
*This holds information on a change.*
- class [Bds::ChangeGroup](#)  
*This holds information on a change.*
- class [Bds::Note](#)  
*This holds information on a [Note](#).*

## Defines

- #define [BDS\\_D\\_H](#) 1

## Enumerations

- enum [Bds::Errors](#) {  
[Bds::ErrorOk](#), [Bds::ErrorMisc](#), [Bds::ErrorWarning](#), [Bds::ErrorEndOfFile](#),  
[Bds::ErrorFile](#), [Bds::ErrorInit](#), [Bds::ErrorConfig](#), [Bds::ErrorResourceLimit](#),  
[Bds::ErrorParam](#), [Bds::ErrorNotImplemented](#), [Bds::ErrorComms](#), [Bds::ErrorTimeout](#),

---

```
Bds::ErrorValidate,    Bds::ErrorValidateMissingBlocks,    Bds::ErrorValidateTimeBackwards,
Bds::ErrorValidateFilenameTime,
Bds::ErrorValidateMetaData, Bds::ErrorValidateFix, Bds::ErrorFormat, Bds::ErrorTimeStamp,
Bds::ErrorEndOfData, Bds::ErrorNoData, Bds::ErrorDataPresent, Bds::ErrorAccessDenied }
• enum Bds::Priority { Bds::PriorityLow, Bds::PriorityNormal, Bds::PriorityHigh }
• enum Bds::SelectionGroup { Bds::SelectionGroupData, Bds::SelectionGroupMetaData }
• enum Bds::SampleFormat { Bds::SampleFormatUnknown, Bds::SampleFormatInt16,
Bds::SampleFormatInt32, Bds::SampleFormatFloat32 }
```

### 10.36.1 Define Documentation

#### 10.36.1.1 #define BSD\_H 1

## 10.37 BdsLib.cpp File Reference

```
#include <BdsLib.h>
```

### Namespaces

- namespace [Bds](#)

### Functions

- void [Bds::bdsChannelGetTypeAux](#) (BString name, BString &type, BString &aux)
- BString [Bds::bdsChannelGetName](#) (BString type, BString aux)
- BError [Bds::bdsDataInfoSetTimeRange](#) (DataInfo &dataInfo)
- BError [Bds::bdsDataInfoFromInfo](#) (BDictString info, DataInfo &dataInfo)  
*Convert info to DataInfo.*
- BError [Bds::bdsInfoFromDataInfo](#) (DataInfo &dataInfo, BDictString &info)
- void [Bds::bdsDumpDataInfo](#) (DataInfo &dataInfo)

## 10.38 BdsLib.h File Reference

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

### Namespaces

- namespace [Bds](#)

### Classes

- class [Bds::ResponseEdit](#)  
*Response Editor object.*

### Functions

- void [Bds::bdsChannelGetTypeAux](#) (BString name, BString &type, BString &aux)
- BString [Bds::bdsChannelGetName](#) (BString type, BString aux)
- BError [Bds::bdsDataInfoSetTimeRange](#) (DataInfo &dataInfo)
- BError [Bds::bdsDataInfoFromInfo](#) (BDictString info, DataInfo &dataInfo)  
*Convert info to DataInfo.*
- BError [Bds::bdsInfoFromDataInfo](#) (DataInfo &dataInfo, BDictString &info)
- void [Bds::bdsDumpDataInfo](#) (DataInfo &dataInfo)

## 10.39 BdsS.cc File Reference

```
#include <BdsC.h>
#include <BdsS.h>
```

### Namespaces

- namespace [Bds](#)

## 10.40 BdsT.cc File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <BdsT.h>
#include <Control.h>
```

## 10.41 overview.dox File Reference

# Index

- ~BdsDataPacket
  - Bds::BdsDataPacket, 42
- ~DataCollate
  - Bds::DataCollate, 74
- ~DataFile
  - Bds::DataFile, 78
- ~DataFileBds
  - Bds::DataFileBds, 90
- ~DataFormats
  - Bds::DataFormats, 112
- ~ResponseEdit
  - Bds::ResponseEdit, 144
- /src/blacknest/bds/bds/bdsDataLib/ Directory Reference, 17
- /src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp, 161
- /src/blacknest/bds/bds/bdsDataLib/BdsCompress.h, 162
- /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp, 163
- /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h, 164
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp, 165
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h, 166
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp, 167
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h, 168
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs-1.cpp, 169
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs-1.h, 170
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp, 171
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h, 172
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp, 173
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h, 174
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp, 175
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h, 176
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp, 177
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h, 178
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp, 179
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h, 180
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp, 181
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h, 182
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWr-1.cpp, 183
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWr-1.h, 184
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWr.cpp, 185
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWr.h, 186
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWrAgso.cpp, 187
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWrAgso.h, 188
- /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp, 189
- /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h, 190
- /src/blacknest/bds/bds/bdsDataLib/canada\_-compress.c, 191
- /src/blacknest/bds/bds/bdsDataLib/canada\_-compress.h, 192
- AccessGroup
  - Bds::AccessGroup, 28
- accessGroupDelete
  - Bds::AdminAccess, 33
- accessGroupGetList
  - Bds::AdminAccess, 33
- accessGroupUpdate
  - Bds::AdminAccess, 33
- InfoPacket
  - Bds::DataFileBds, 92
- address

- Bds::User, 159
- addSource
  - Bds::DataCollate, 74
- AdminAccess
  - Bds::AdminAccess, 32
- amplitude
  - Bds::Fap, 119
- apiVersion
  - Bds, 25
- array
  - Bds::DataInfo, 115
- ArrayChannel
  - Bds::ArrayChannel, 39
- arrayOffsetEast
  - Bds::Location, 133
- arrayOffsetNorth
  - Bds::Location, 133
- baseSamplingFrequency
  - Bds::Digitiser, 118
- Bds, 19
  - apiVersion, 25
  - bdsChannelGetName, 24
  - bdsChannelGetTypeAux, 24
  - bdsDataInfoFromInfo, 24
  - bdsDataInfoSetTimeRange, 24
  - BdsDataType, 22
  - BdsDataTypeBlock, 22
  - BdsDataTypeData, 22
  - BdsDataTypeInfo, 22
  - bdsDumpDataInfo, 25
  - bdsInfoFromDataInfo, 25
  - bdsUnCompressCm8, 25
  - BdsVersion, 25
  - cm6Table, 25
  - crc, 25
  - dataCalculateDifference, 25
  - dataChecksum, 25
  - dataCompressCm6, 25
  - dataConvert, 25
  - dataFormats, 25
  - ErrorAccessDenied, 23
  - ErrorComms, 23
  - ErrorConfig, 23
  - ErrorDataPresent, 23
  - ErrorEndOfData, 23
  - ErrorEndOfFile, 22
  - ErrorFile, 23
  - ErrorFormat, 23
  - ErrorInit, 23
  - ErrorMisc, 22
  - ErrorNoData, 23
  - ErrorNotImplemented, 23
  - ErrorOk, 22
  - ErrorParam, 23
  - ErrorResourceLimit, 23
  - Errors, 22
  - ErrorTimeout, 23
  - ErrorTimeStamp, 23
  - ErrorValidate, 23
  - ErrorValidateFilenameTime, 23
  - ErrorValidateFix, 23
  - ErrorValidateMetaData, 23
  - ErrorValidateMissingBlocks, 23
  - ErrorValidateTimeBackwards, 23
  - ErrorWarning, 22
  - FileHeaderType, 23
  - FileHeaderType\_Standard, 23
  - FileHeaderType\_TapeDigitiser, 23
  - FileSampleType, 23
  - FileSampleType\_Float32, 23
  - FileSampleType\_Float64, 23
  - FileSampleType\_Int16, 23
  - FileSampleType\_Int32, 23
  - FileSampleType\_Unknown, 23
  - Priority, 23
  - PriorityHigh, 23
  - PriorityLow, 23
  - PriorityNormal, 23
  - SampleFormat, 23
  - SampleFormatFloat32, 24
  - SampleFormatInt16, 24
  - SampleFormatInt32, 24
  - SampleFormatUnknown, 24
  - Scale, 25
  - SelectionGroup, 24
  - SelectionGroupData, 24
  - SelectionGroupMetaData, 24
- Bds::AccessGroup, 27
  - AccessGroup, 28
  - endTime, 28
  - getMembers, 28
  - getType, 28
  - group, 28
  - id, 28
  - network, 28
  - setMembers, 28
  - startTime, 28
  - station, 28
- Bds::AdminAccess, 29
  - accessGroupDelete, 33
  - accessGroupGetList, 33
  - accessGroupUpdate, 33
  - AdminAccess, 32
  - calibrationDelete, 36
  - calibrationGetList, 35
  - calibrationUpdate, 35
  - changeDelete, 38

- changeGetList, 38
- changeGetListNumber, 38
- changeGroupDelete, 38
- changeGroupEnd, 38
- changeGroupGetList, 38
- changeGroupStart, 37
- channelDelete, 34
- channelGet, 34
- channelGetList, 34
- channelInstrumentDelete, 35
- channelInstrumentGetList, 34
- channelInstrumentUpdate, 34
- channelUpdate, 34
- connect, 32
- dataClose, 37
- dataFormattedGetLength, 37
- dataFormattedRead, 37
- dataGetBlock, 36
- dataGetChannelInfo, 36
- dataGetInfo, 36
- dataGetWarnings, 36
- dataOpen, 36
- dataPutBlock, 37
- dataSearch, 36
- dataSeekBlock, 36
- dataSetInfo, 36
- digitiserDelete, 35
- digitiserGet, 35
- digitiserGetList, 35
- digitiserUpdate, 35
- getSelectionInfo, 33
- getSelections, 33
- getVersion, 32
- groupDelete, 33
- groupGetList, 33
- groupUpdate, 33
- locationDelete, 34
- locationGetList, 34
- locationUpdate, 34
- networkDelete, 34
- networkGetList, 33
- networkUpdate, 34
- noteDelete, 38
- noteGetList, 38
- noteUpdate, 38
- responseDelete, 36
- responseGetList, 36
- responseUpdate, 36
- sensorDelete, 35
- sensorGet, 35
- sensorGetList, 35
- sensorLocationDelete, 35
- sensorLocationGetList, 35
- sensorLocationUpdate, 35
- sensorUpdate, 35
- setUser, 32
- setUserReal, 32
- sqlQuery, 38
- stationDelete, 34
- stationGetList, 34
- stationUpdate, 34
- statisticsGet, 38
- userDelete, 33
- userGetGroups, 33
- userGetList, 32
- userUpdate, 32
- Bds::ArrayChannel, 39
  - ArrayChannel, 39
  - channel, 39
  - station, 39
- Bds::BdsDataBlock, 40
  - data, 40
  - header, 40
- Bds::BdsDataBlockHeader, 41
  - length, 41
  - packetOffset, 41
  - type, 41
- Bds::BdsDataPacket, 42
  - ~BdsDataPacket, 42
  - BdsDataPacket, 42
  - clear, 42
  - dump, 42
  - getHeader, 42
  - setChecksumAndLength, 42
  - setHeader, 42
  - validateChecksum, 42
- Bds::BdsDataPacketHeader, 43
  - checksum, 43
  - endTime, 44
  - length, 43
  - sequence, 43
  - startTime, 44
  - streamlet, 43
  - type, 43
- Bds::BdsDataStreamlet, 45
  - BdsDataStreamlet, 45
  - packetNumber, 45
  - position, 45
- Bds::Calibration, 46
  - Calibration, 47
  - calibrationFactor, 48
  - calibrationFrequency, 48
  - calibrationUnits, 48
  - channel, 47
  - endTime, 47
  - getMembers, 47
  - getType, 47
  - id, 47

- network, 47
- samplingFrequency, 47
- setMembers, 47
- startTime, 47
- station, 47
- Bds::Change, 49
  - Change, 50
  - changeGroupId, 50
  - getMembers, 50
  - getType, 50
  - id, 50
  - rowId, 50
  - setMembers, 50
  - table, 50
  - time, 50
  - type, 50
- Bds::ChangeGroup, 51
  - ChangeGroup, 52
  - description, 52
  - getMembers, 52
  - getType, 52
  - id, 52
  - setMembers, 52
  - time, 52
  - title, 52
  - type, 52
  - user, 52
- Bds::Channel, 53
  - Channel, 54
  - channel, 54
  - channelAux, 54
  - channelType, 54
  - description, 55
  - endTime, 54
  - getMembers, 54
  - getType, 54
  - id, 54
  - network, 54
  - setMembers, 54
  - startTime, 54
  - station, 54
- Bds::ChannelInfo, 56
  - calibration, 57
  - channel, 57
  - ChannelInfo, 56
  - digitiser, 57
  - location, 56
  - responses, 57
  - sensor, 57
  - sensorLocation, 57
  - station, 56
- Bds::ChannelInstrument, 58
  - channelId, 59
  - ChannelInstrument, 59
  - digitiserId, 59
  - endTime, 59
  - getMembers, 59
  - getType, 59
  - id, 59
  - sensorId, 59
  - setMembers, 59
  - startTime, 59
- Bds::DataAccess, 60
  - connect, 61
  - DataAccess, 61
  - dataClose, 63
  - dataFormattedGetLength, 63
  - dataFormattedRead, 63
  - dataGetBlock, 63
  - dataGetChannelInfo, 62
  - dataGetInfo, 62
  - dataGetWarnings, 62
  - dataOpen, 62
  - dataSearch, 62
  - dataSeekBlock, 62
  - getSelectionInfo, 62
  - getSelections, 62
  - getVersion, 61
  - networkGetList, 62
  - noteUpdate, 63
  - setUser, 61
  - setUserReal, 61
  - stationGetList, 62
  - userGetGroups, 62
- Bds::DataAddAccess, 64
  - connect, 65
  - DataAddAccess, 65
  - dataClose, 67
  - dataFormattedGetLength, 67
  - dataFormattedRead, 67
  - dataGetBlock, 67
  - dataGetChannelInfo, 66
  - dataGetInfo, 66
  - dataGetWarnings, 66
  - dataOpen, 66
  - dataPutBlock, 67
  - dataSearch, 66
  - dataSeekBlock, 67
  - dataSetInfo, 67
  - getSelectionInfo, 66
  - getSelections, 66
  - getVersion, 65
  - networkGetList, 66
  - noteUpdate, 67
  - setUser, 65
  - setUserReal, 65
  - stationGetList, 66
  - userGetGroups, 66

- Bds::DataBlock, 68
  - channelData, 69
  - channelNumber, 68
  - DataBlock, 68
  - endTime, 68
  - info, 69
  - segmentNumber, 69
  - startTime, 68
- Bds::DataChannel, 70
  - channel, 72
  - DataChannel, 71
  - dataFileChannel, 72
  - dataFileId, 72
  - endTime, 71
  - getMembers, 71
  - getType, 71
  - id, 71
  - info, 72
  - network, 71
  - numBlocks, 72
  - numSamples, 72
  - sampleFormat, 72
  - sampleRate, 72
  - setMembers, 71
  - source, 72
  - startTime, 71
  - station, 72
- Bds::DataCollate, 74
  - ~DataCollate, 74
  - addSource, 74
  - DataCollate, 74
  - readData, 74
- Bds::DataFile, 75
  - ~DataFile, 78
  - close, 78
  - DataFile, 78
  - DataOrder, 77
  - DataOrderAll, 77
  - DataOrderChannel, 77
  - DataOrderSample, 77
  - DataOrderUnknown, 77
  - end, 79
  - FeatureCanRead, 77
  - FeatureCanWrite, 77
  - FeatureNone, 77
  - Features, 77
  - fileNameProcess, 79
  - getDataOrder, 78
  - getFeatures, 78
  - getFormat, 79
  - getFormats, 79
  - getInfo, 79
  - init, 78
  - ofile, 80
  - ofileName, 80
  - ofileNameTime, 80
  - ofFormat, 80
  - omode, 80
  - open, 78
  - OptionFileNameProcess, 77
  - OptionNone, 77
  - Options, 77
  - OptionsList, 77
  - OptionValidate, 77
  - OptionValidateCorruptions, 77
  - readData, 79
  - seekBlock, 79
  - setFormat, 78
  - setInfo, 78
  - start, 78
  - writeData, 79
- Bds::DataFileAscii, 81
  - DataFileAscii, 81
  - getFormats, 82
  - odataInfo, 82
  - open, 81
  - setInfo, 81
  - writeData, 82
- Bds::DataFileBdrs, 83
  - DataFileBdrs, 84
  - getDataOrder, 84, 85
  - getFeatures, 84, 85
  - getFormats, 85
  - getInfo, 84, 85
  - oblockPositions, 86
  - oblockSize, 86
  - oblockYear, 86
  - odataInfo, 86
  - ofileNameDate, 86
  - omagic, 86
  - open, 84
  - osampleRate, 86
  - oyear, 86
  - readBlock, 85
  - readData, 84, 85
- Bds::DataFileBds, 87
  - ~DataFileBds, 90
  - addInfoPacket, 92
  - clear, 91
  - close, 90
  - dataAppend, 92
  - DataFileBds, 90
  - dataGet, 92
  - DefaultBlockSize, 90
  - diskBlockRead, 93
  - diskBlockReset, 92
  - diskBlockSeek, 93
  - diskBlockWrite, 92

- diskBlockWriteFlush, 92
- getDataOrder, 91
- getDiskBlockSize, 91
- getFormats, 91
- getInfo, 91
- infoGet, 92
- infoSet, 92
- ochannelInfos, 93
- odataInfo, 93
- odiskBlock, 93
- odiskBlockSize, 93
- odiskPosition, 94
- oformat, 93
- oinfo, 93
- oinfoHeader, 93
- oinfoRepeat, 93
- opacket, 94
- opackFormat, 93
- open, 90
- ostreamlets, 94
- packetRead, 92
- packetSeek, 92
- packetWrite, 92
- PackFormat, 90
- PackFormat\_CM, 90
- PackFormat\_SM, 90
- PackFormat\_SM\_CC, 90
- PackFormat\_Unknown, 90
- readData, 91
- seekBlock, 91
- setDiskBlockSize, 91
- setFormat, 90
- setInfo, 90
- setInfoRepeat, 92
- StreamsMax, 90
- writeData, 91
- Bds::DataFileBknas, 95
  - DataFileBknas, 95
  - getFormats, 96
  - odataInfo, 96
  - open, 95
  - setInfo, 95
  - writeData, 96
- Bds::DataFileGcf, 97
  - DataFileGcf, 97
  - getFeatures, 97
  - getFormats, 98
  - getInfo, 97
  - oblockSize, 98
  - odataInfo, 98
  - ofilterCoef, 98
  - oformat, 98
  - osampleRate, 98
  - ostreamId, 98
  - osystemId, 98
  - readData, 98
- Bds::DataFileIms, 99
  - close, 100
  - DataFileIms, 100
  - end, 101
  - getDataOrder, 100
  - getFeatures, 100
  - getFormats, 101
  - ochannelInfos, 101
  - odataInfo, 101
  - open, 100
  - oprevValue1, 101
  - oprevValue2, 101
  - owriteChannel, 101
  - owriteChecksum, 101
  - owriteColumn, 101
  - setInfo, 100
  - start, 100
  - writeData, 100
- Bds::DataFileTapeDigitiser, 102
  - blockNumSamples, 103
  - blockSize, 103
  - computeChecksum, 104
  - DataFileTapeDigitiser, 103
  - getFormats, 103
  - getInfo, 103
  - oblockNumSamples, 104
  - oblockSize, 104
  - obuffer, 104
  - oheader, 104
  - oheaderSize, 104
  - open, 103
  - osampleRate, 104
  - oversion, 104
  - readData, 103
  - readHeader, 103
- Bds::DataFileWra, 105
  - DataFileWra, 106
  - getDataOrder, 106, 107
  - getFeatures, 106, 107
  - getFormats, 107
  - getInfo, 107
  - oblockPositions, 108
  - oblockSize, 108
  - oblockYear, 108
  - odataInfo, 108
  - ofilenameDate, 108
  - omagic, 108
  - onumChannels, 108
  - open, 106
  - osampleRate, 108
  - oyear, 108
  - readBlock, 107, 108

- readData, 107
- setFormat, 106, 107
- Bds::DataFileWraAgso, 109
  - DataFileWraAgso, 110
  - getDataOrder, 110
  - getFeatures, 110
  - getFormats, 110
  - getInfo, 110
  - oblockSize, 111
  - oblockYear, 111
  - ochannelStartBlocks, 111
  - ocurrentBlock, 111
  - odataInfo, 111
  - ofieldWidths, 111
  - oreadBlock, 111
  - oreadChannel, 111
  - osampleRate, 111
  - oyear, 111
  - readBlock, 110
  - readData, 110
- Bds::DataFormats, 112
  - ~DataFormats, 112
  - DataFormats, 112
  - findString, 112
  - formatGet, 112
  - formatList, 112
- Bds::DataHandle, 113
  - DataHandle, 113
  - handle, 113
  - startBlock, 113
- Bds::DataInfo, 114
  - array, 115
  - channels, 115
  - DataInfo, 114
  - description, 115
  - endTime, 114
  - info, 115
  - startTime, 114
  - synchronous, 115
- Bds::Digitiser, 116
  - baseSamplingFrequency, 118
  - Digitiser, 117
  - endTime, 117
  - gain, 118
  - getMembers, 117
  - getType, 117
  - id, 117
  - initialSamplingFrequency, 118
  - name, 117
  - numberChannels, 117
  - serialNumber, 117
  - setMembers, 117
  - startTime, 117
  - type, 117
- Bds::Fap, 119
  - amplitude, 119
  - Fap, 119
  - frequency, 119
  - phase, 119
- Bds::FileDataHeaderTape\_1v1, 121
  - dataLength, 122
  - dataType, 122
  - endTime, 122
  - headerLength, 121
  - numChannels, 122
  - numSamples, 122
  - startTime, 122
  - tapeFmSignalLevels, 122
  - tapeLastVelaCode, 122
  - tapeSignalLevelErrors, 122
  - tapeSpeedVariationErrors, 122
  - tapeVelaBitErrors, 122
  - type, 121
- Bds::FileDataHeaderTape\_2v0, 124
  - dataCheckSum, 126
  - dataLength, 125
  - dataType, 125
  - endTime, 125
  - headerCheckSum, 126
  - headerLength, 125
  - numChannels, 125
  - numSamples, 125
  - startTime, 125
  - tapeFmSignalLevels, 126
  - tapeLastVelaCode, 125
  - tapeSignalLevelErrors, 125
  - tapeSpeedVariationErrors, 125
  - tapeVelaBitErrors, 125
  - type, 125
- Bds::Group, 127
  - description, 128
  - getMembers, 127
  - getType, 127
  - Group, 127
  - group, 127
  - id, 127
  - setMembers, 127
- Bds::ListRange, 129
  - getMembers, 129
  - getType, 129
  - ListRange, 129
  - number, 129
  - reverse, 130
  - setMembers, 129
  - start, 129
- Bds::Location, 131
  - arrayOffsetEast, 133
  - arrayOffsetNorth, 133

- datum, 132
- elevation, 133
- endTime, 132
- getMembers, 132
- getType, 132
- id, 132
- latitude, 133
- Location, 132
- longitude, 132
- network, 132
- setMembers, 132
- startTime, 132
- station, 132
- Bds::Network, 134
  - description, 135
  - getMembers, 135
  - getType, 135
  - id, 135
  - Network, 135
  - network, 135
  - setMembers, 135
  - stations, 135
- Bds::Note, 136
  - description, 137
  - getMembers, 137
  - getType, 137
  - id, 137
  - Note, 137
  - setMembers, 137
  - time, 137
  - title, 137
  - type, 137
  - user, 137
- Bds::Point, 138
  - Point, 138
  - x, 138
  - y, 138
- Bds::PoleZero, 139
  - poles, 139
  - PoleZero, 139
  - zeros, 139
- Bds::Response, 140
  - channel, 142
  - decimation, 142
  - endTime, 141
  - faps, 142
  - firCoefficients, 142
  - gain, 142
  - gainFrequency, 142
  - id, 141
  - name, 142
  - network, 141
  - poleZeros, 142
  - Response, 141
  - startTime, 141
  - station, 142
  - symmetry, 142
  - type, 142
- Bds::ResponseEdit, 144
  - ~ResponseEdit, 144
  - getString, 144
  - ResponseEdit, 144
  - setString, 144
- Bds::Selection, 145
  - channelId, 146
  - channels, 146
  - digitiserId, 146
  - endTime, 146
  - range, 146
  - Selection, 146
  - sensorId, 146
  - sensorOldId, 146
  - startTime, 146
- Bds::SelectionChannel, 147
  - channel, 147
  - network, 147
  - SelectionChannel, 147
  - source, 147
  - station, 147
- Bds::SelectionInfo, 148
  - channels, 149
  - endTime, 149
  - networks, 149
  - numDataChannels, 149
  - SelectionInfo, 149
  - sources, 149
  - startTime, 149
  - stations, 149
- Bds::Sensor, 150
  - endTime, 151
  - gain, 152
  - gainUnits, 152
  - getMembers, 151
  - getType, 151
  - id, 151
  - name, 151
  - numberChannels, 151
  - oldId, 152
  - Sensor, 151
  - serialNumber, 151
  - setMembers, 151
  - startTime, 151
  - type, 151
- Bds::SensorLocation, 153
  - depth, 154
  - endTime, 154
  - getMembers, 154
  - getType, 154

- horizontalAngle, 154
- id, 154
- sensorId, 154
- SensorLocation, 154
- setMembers, 154
- startTime, 154
- verticalAngle, 154
- Bds::Station, 155
  - channels, 156
  - description, 155
  - id, 155
  - name, 155
  - Station, 155
  - type, 155
- Bds::TimePeriod, 157
  - endTime, 157
  - getMembers, 157
  - getType, 157
  - setMembers, 157
  - startTime, 157
  - TimePeriod, 157
- Bds::User, 158
  - address, 159
  - email, 159
  - enabled, 159
  - getMembers, 159
  - getType, 159
  - groups, 159
  - id, 159
  - name, 159
  - password, 159
  - setMembers, 159
  - telephone, 159
  - User, 159
  - user, 159
- BdsC.cc, 193
- BdsC.h, 194
  - BDSC\_H, 194
- BDSC\_H
  - BdsC.h, 194
- bdsChannelGetName
  - Bds, 24
- bdsChannelGetTypeAux
  - Bds, 24
- BdsD.cc, 195
- BdsD.h, 196
  - BDSD\_H, 198
- BDSD\_H
  - BdsD.h, 198
- BdsDataFileBds.cpp
  - dprintf, 173
  - LDEBUG, 173
- BdsDataFileBds.h
  - BdsDataFileBds\_H, 174
- BdsDataFileBds\_H
  - BdsDataFileBds.h, 174
- BdsDataFileGcf.cpp
  - DEBUG, 177
- BdsDataFileTapeDigitiser.h
  - BdsDataFileTapeDigitiser\_H, 182
- BdsDataFileTapeDigitiser\_H
  - BdsDataFileTapeDigitiser.h, 182
- BdsDataFileWraAgso.cpp
  - parseStringFixedFields, 187
- bdsDataInfoFromInfo
  - Bds, 24
- bdsDataInfoSetTimeRange
  - Bds, 24
- BdsDataPacket
  - Bds::BdsDataPacket, 42
- BdsDataStreamlet
  - Bds::BdsDataStreamlet, 45
- BdsDataType
  - Bds, 22
- BdsDataTypeBlock
  - Bds, 22
- BdsDataTypeData
  - Bds, 22
- BdsDataTypeInfo
  - Bds, 22
- bdsDumpDataInfo
  - Bds, 25
- bdsInfoFromDataInfo
  - Bds, 25
- BdsLib.cpp, 199
- BdsLib.h, 200
- BdsS.cc, 201
- BdsT.cc, 202
- bdsUnCompressCm8
  - Bds, 25
- BdsVersion
  - Bds, 25
- blockNumSamples
  - Bds::DataFileTapeDigitiser, 103
- blockSize
  - Bds::DataFileTapeDigitiser, 103
- Calibration
  - Bds::Calibration, 47
- calibration
  - Bds::ChannelInfo, 57
- calibrationDelete
  - Bds::AdminAccess, 36
- calibrationFactor
  - Bds::Calibration, 48
- calibrationFrequency
  - Bds::Calibration, 48
- calibrationGetList

- Bds::AdminAccess, 35
- calibrationUnits
  - Bds::Calibration, 48
- calibrationUpdate
  - Bds::AdminAccess, 35
- canada\_compress
  - canada\_compress.c, 191
  - canada\_compress.h, 192
- canada\_compress.c
  - canada\_compress, 191
  - canada\_uncompress, 191
  - corrupt, 191
  - htonl, 191
  - ntohl, 191
  - pack, 191
  - unpack, 191
- canada\_compress.h
  - canada\_compress, 192
  - canada\_uncompress, 192
  - CANCOMP\_CORRUPT, 192
  - CANCOMP\_ERR, 192
  - CANCOMP\_EXCEED, 192
  - CANCOMP\_NOT\_20, 192
  - CANCOMP\_SUCCESS, 192
- canada\_uncompress
  - canada\_compress.c, 191
  - canada\_compress.h, 192
- CANCOMP\_CORRUPT
  - canada\_compress.h, 192
- CANCOMP\_ERR
  - canada\_compress.h, 192
- CANCOMP\_EXCEED
  - canada\_compress.h, 192
- CANCOMP\_NOT\_20
  - canada\_compress.h, 192
- CANCOMP\_SUCCESS
  - canada\_compress.h, 192
- Change
  - Bds::Change, 50
- changeDelete
  - Bds::AdminAccess, 38
- changeGetList
  - Bds::AdminAccess, 38
- changeGetListNumber
  - Bds::AdminAccess, 38
- ChangeGroup
  - Bds::ChangeGroup, 52
- changeGroupDelete
  - Bds::AdminAccess, 38
- changeGroupEnd
  - Bds::AdminAccess, 38
- changeGroupGetList
  - Bds::AdminAccess, 38
- changeGroupId
  - Bds::Change, 50
- changeGroupStart
  - Bds::AdminAccess, 37
- Channel
  - Bds::Channel, 54
- channel
  - Bds::ArrayChannel, 39
  - Bds::Calibration, 47
  - Bds::Channel, 54
  - Bds::ChannelInfo, 57
  - Bds::DataChannel, 72
  - Bds::Response, 142
  - Bds::SelectionChannel, 147
- channelAux
  - Bds::Channel, 54
- channelData
  - Bds::DataBlock, 69
- channelDelete
  - Bds::AdminAccess, 34
- channelGet
  - Bds::AdminAccess, 34
- channelGetList
  - Bds::AdminAccess, 34
- channelId
  - Bds::ChannelInstrument, 59
  - Bds::Selection, 146
- ChannelInfo
  - Bds::ChannelInfo, 56
- ChannelInstrument
  - Bds::ChannelInstrument, 59
- channelInstrumentDelete
  - Bds::AdminAccess, 35
- channelInstrumentGetList
  - Bds::AdminAccess, 34
- channelInstrumentUpdate
  - Bds::AdminAccess, 34
- channelNumber
  - Bds::DataBlock, 68
- channels
  - Bds::DataInfo, 115
  - Bds::Selection, 146
  - Bds::SelectionInfo, 149
  - Bds::Station, 156
- channelType
  - Bds::Channel, 54
- channelUpdate
  - Bds::AdminAccess, 34
- checksum
  - Bds::BdsDataPacketHeader, 43
- clear
  - Bds::BdsDataPacket, 42
  - Bds::DataFileBds, 91
- close
  - Bds::DataFile, 78

- Bds::DataFileBds, 90
- Bds::DataFileIms, 100
- cm6Table
  - Bds, 25
- computeChecksum
  - Bds::DataFileTapeDigitiser, 104
- connect
  - Bds::AdminAccess, 32
  - Bds::DataAccess, 61
  - Bds::DataAddAccess, 65
- corrupt
  - canada\_compress.c, 191
- crc
  - Bds, 25
- data
  - Bds::BdsDataBlock, 40
- DataAccess
  - Bds::DataAccess, 61
- DataAddAccess
  - Bds::DataAddAccess, 65
- dataAppend
  - Bds::DataFileBds, 92
- DataBlock
  - Bds::DataBlock, 68
- dataCalculateDifference
  - Bds, 25
- DataChannel
  - Bds::DataChannel, 71
- dataChecksum
  - Bds::FileDataHeaderTape\_2v0, 126
- dataChecksum
  - Bds, 25
- dataClose
  - Bds::AdminAccess, 37
  - Bds::DataAccess, 63
  - Bds::DataAddAccess, 67
- DataCollate
  - Bds::DataCollate, 74
- dataCompressCm6
  - Bds, 25
- dataConvert
  - Bds, 25
- DataFile
  - Bds::DataFile, 78
- DataFileAscii
  - Bds::DataFileAscii, 81
- DataFileBdrs
  - Bds::DataFileBdrs, 84
- DataFileBds
  - Bds::DataFileBds, 90
- DataFileBknas
  - Bds::DataFileBknas, 95
- dataFileChannel
  - Bds::DataChannel, 72
- DataFileGcf
  - Bds::DataFileGcf, 97
- dataFileId
  - Bds::DataChannel, 72
- DataFileIms
  - Bds::DataFileIms, 100
- DataFileTapeDigitiser
  - Bds::DataFileTapeDigitiser, 103
- DataFileWra
  - Bds::DataFileWra, 106
- DataFileWraAgso
  - Bds::DataFileWraAgso, 110
- DataFormats
  - Bds::DataFormats, 112
- dataFormats
  - Bds, 25
- dataFormattedGetLength
  - Bds::AdminAccess, 37
  - Bds::DataAccess, 63
  - Bds::DataAddAccess, 67
- dataFormattedRead
  - Bds::AdminAccess, 37
  - Bds::DataAccess, 63
  - Bds::DataAddAccess, 67
- dataGet
  - Bds::DataFileBds, 92
- dataGetBlock
  - Bds::AdminAccess, 36
  - Bds::DataAccess, 63
  - Bds::DataAddAccess, 67
- dataGetChannelInfo
  - Bds::AdminAccess, 36
  - Bds::DataAccess, 62
  - Bds::DataAddAccess, 66
- dataGetInfo
  - Bds::AdminAccess, 36
  - Bds::DataAccess, 62
  - Bds::DataAddAccess, 66
- dataGetWarnings
  - Bds::AdminAccess, 36
  - Bds::DataAccess, 62
  - Bds::DataAddAccess, 66
- DataHandle
  - Bds::DataHandle, 113
- DataInfo
  - Bds::DataInfo, 114
- dataLength
  - Bds::FileDataHeaderTape\_1v1, 122
  - Bds::FileDataHeaderTape\_2v0, 125
- dataOpen
  - Bds::AdminAccess, 36
  - Bds::DataAccess, 62
  - Bds::DataAddAccess, 66

- DataOrder
  - Bds::DataFile, 77
- DataOrderAll
  - Bds::DataFile, 77
- DataOrderChannel
  - Bds::DataFile, 77
- DataOrderSample
  - Bds::DataFile, 77
- DataOrderUnknown
  - Bds::DataFile, 77
- dataPutBlock
  - Bds::AdminAccess, 37
  - Bds::DataAddAccess, 67
- dataSearch
  - Bds::AdminAccess, 36
  - Bds::DataAccess, 62
  - Bds::DataAddAccess, 66
- dataSeekBlock
  - Bds::AdminAccess, 36
  - Bds::DataAccess, 62
  - Bds::DataAddAccess, 67
- dataSetInfo
  - Bds::AdminAccess, 36
  - Bds::DataAddAccess, 67
- dataType
  - Bds::FileDataHeaderTape\_1v1, 122
  - Bds::FileDataHeaderTape\_2v0, 125
- datum
  - Bds::Location, 132
- DEBUG
  - BdsDataFileGcf.cpp, 177
- decimation
  - Bds::Response, 142
- DefaultBlockSize
  - Bds::DataFileBds, 90
- depth
  - Bds::SensorLocation, 154
- description
  - Bds::ChangeGroup, 52
  - Bds::Channel, 55
  - Bds::DataInfo, 115
  - Bds::Group, 128
  - Bds::Network, 135
  - Bds::Note, 137
  - Bds::Station, 155
- Digitiser
  - Bds::Digitiser, 117
- digitiser
  - Bds::ChannelInfo, 57
- digitiserDelete
  - Bds::AdminAccess, 35
- digitiserGet
  - Bds::AdminAccess, 35
- digitiserGetList
  - Bds::AdminAccess, 35
- digitiserId
  - Bds::ChannelInstrument, 59
  - Bds::Selection, 146
- digitiserUpdate
  - Bds::AdminAccess, 35
- diskBlockRead
  - Bds::DataFileBds, 93
- diskBlockReset
  - Bds::DataFileBds, 92
- diskBlockSeek
  - Bds::DataFileBds, 93
- diskBlockWrite
  - Bds::DataFileBds, 92
- diskBlockWriteFlush
  - Bds::DataFileBds, 92
- dprintf
  - BdsDataFileBds.cpp, 173
- dump
  - Bds::BdsDataPacket, 42
- elevation
  - Bds::Location, 133
- email
  - Bds::User, 159
- enabled
  - Bds::User, 159
- end
  - Bds::DataFile, 79
  - Bds::DataFileIms, 101
- endTime
  - Bds::AccessGroup, 28
  - Bds::BdsDataPacketHeader, 44
  - Bds::Calibration, 47
  - Bds::Channel, 54
  - Bds::ChannelInstrument, 59
  - Bds::DataBlock, 68
  - Bds::DataChannel, 71
  - Bds::DataInfo, 114
  - Bds::Digitiser, 117
  - Bds::FileDataHeaderTape\_1v1, 122
  - Bds::FileDataHeaderTape\_2v0, 125
  - Bds::Location, 132
  - Bds::Response, 141
  - Bds::Selection, 146
  - Bds::SelectionInfo, 149
  - Bds::Sensor, 151
  - Bds::SensorLocation, 154
  - Bds::TimePeriod, 157
- ErrorAccessDenied
  - Bds, 23
- ErrorComms
  - Bds, 23
- ErrorConfig

- Bds, 23
- ErrorDataPresent
  - Bds, 23
- ErrorEndOfData
  - Bds, 23
- ErrorEndOfFile
  - Bds, 22
- ErrorFile
  - Bds, 23
- ErrorFormat
  - Bds, 23
- ErrorInit
  - Bds, 23
- ErrorMisc
  - Bds, 22
- ErrorNoData
  - Bds, 23
- ErrorNotImplemented
  - Bds, 23
- ErrorOk
  - Bds, 22
- ErrorParam
  - Bds, 23
- ErrorResourceLimit
  - Bds, 23
- Errors
  - Bds, 22
- ErrorTimeout
  - Bds, 23
- ErrorTimeStamp
  - Bds, 23
- ErrorValidate
  - Bds, 23
- ErrorValidateFilenameTime
  - Bds, 23
- ErrorValidateFix
  - Bds, 23
- ErrorValidateMetaData
  - Bds, 23
- ErrorValidateMissingBlocks
  - Bds, 23
- ErrorValidateTimeBackwards
  - Bds, 23
- ErrorWarning
  - Bds, 22
- Fap
  - Bds::Fap, 119
- faps
  - Bds::Response, 142
- FeatureCanRead
  - Bds::DataFile, 77
- FeatureCanWrite
  - Bds::DataFile, 77
- FeatureNone
  - Bds::DataFile, 77
- Features
  - Bds::DataFile, 77
- FileHeaderType
  - Bds, 23
- FileHeaderType\_Standard
  - Bds, 23
- FileHeaderType\_TapeDigitiser
  - Bds, 23
- fileNameProcess
  - Bds::DataFile, 79
- FileSampleType
  - Bds, 23
- FileSampleType\_Float32
  - Bds, 23
- FileSampleType\_Float64
  - Bds, 23
- FileSampleType\_Int16
  - Bds, 23
- FileSampleType\_Int32
  - Bds, 23
- FileSampleType\_Unknown
  - Bds, 23
- findString
  - Bds::DataFormats, 112
- firCoefficients
  - Bds::Response, 142
- formatGet
  - Bds::DataFormats, 112
- formatList
  - Bds::DataFormats, 112
- frequency
  - Bds::Fap, 119
- gain
  - Bds::Digitiser, 118
  - Bds::Response, 142
  - Bds::Sensor, 152
- gainFrequency
  - Bds::Response, 142
- gainUnits
  - Bds::Sensor, 152
- getDataOrder
  - Bds::DataFile, 78
  - Bds::DataFileBdrs, 84, 85
  - Bds::DataFileBds, 91
  - Bds::DataFileIms, 100
  - Bds::DataFileWra, 106, 107
  - Bds::DataFileWraAgso, 110
- getDiskBlockSize
  - Bds::DataFileBds, 91
- getFeatures
  - Bds::DataFile, 78

- Bds::DataFileBdrs, 84, 85
- Bds::DataFileGcf, 97
- Bds::DataFileIms, 100
- Bds::DataFileWra, 106, 107
- Bds::DataFileWraAgso, 110
- getFormat
  - Bds::DataFile, 79
- getFormats
  - Bds::DataFile, 79
  - Bds::DataFileAscii, 82
  - Bds::DataFileBdrs, 85
  - Bds::DataFileBds, 91
  - Bds::DataFileBknas, 96
  - Bds::DataFileGcf, 98
  - Bds::DataFileIms, 101
  - Bds::DataFileTapeDigitiser, 103
  - Bds::DataFileWra, 107
  - Bds::DataFileWraAgso, 110
- getHeader
  - Bds::BdsDataPacket, 42
- getInfo
  - Bds::DataFile, 79
  - Bds::DataFileBdrs, 84, 85
  - Bds::DataFileBds, 91
  - Bds::DataFileGcf, 97
  - Bds::DataFileTapeDigitiser, 103
  - Bds::DataFileWra, 107
  - Bds::DataFileWraAgso, 110
- getMembers
  - Bds::AccessGroup, 28
  - Bds::Calibration, 47
  - Bds::Change, 50
  - Bds::ChangeGroup, 52
  - Bds::Channel, 54
  - Bds::ChannelInstrument, 59
  - Bds::DataChannel, 71
  - Bds::Digitiser, 117
  - Bds::Group, 127
  - Bds::ListRange, 129
  - Bds::Location, 132
  - Bds::Network, 135
  - Bds::Note, 137
  - Bds::Sensor, 151
  - Bds::SensorLocation, 154
  - Bds::TimePeriod, 157
  - Bds::User, 159
- getSelectionInfo
  - Bds::AdminAccess, 33
  - Bds::DataAccess, 62
  - Bds::DataAddAccess, 66
- getSelections
  - Bds::AdminAccess, 33
  - Bds::DataAccess, 62
  - Bds::DataAddAccess, 66
- getString
  - Bds::ResponseEdit, 144
- getType
  - Bds::AccessGroup, 28
  - Bds::Calibration, 47
  - Bds::Change, 50
  - Bds::ChangeGroup, 52
  - Bds::Channel, 54
  - Bds::ChannelInstrument, 59
  - Bds::DataChannel, 71
  - Bds::Digitiser, 117
  - Bds::Group, 127
  - Bds::ListRange, 129
  - Bds::Location, 132
  - Bds::Network, 135
  - Bds::Note, 137
  - Bds::Sensor, 151
  - Bds::SensorLocation, 154
  - Bds::TimePeriod, 157
  - Bds::User, 159
- getVersion
  - Bds::AdminAccess, 32
  - Bds::DataAccess, 61
  - Bds::DataAddAccess, 65
- Group
  - Bds::Group, 127
- group
  - Bds::AccessGroup, 28
  - Bds::Group, 127
- groupDelete
  - Bds::AdminAccess, 33
- groupGetList
  - Bds::AdminAccess, 33
- groups
  - Bds::User, 159
- groupUpdate
  - Bds::AdminAccess, 33
- handle
  - Bds::DataHandle, 113
- header
  - Bds::BdsDataBlock, 40
- headerChecksum
  - Bds::FileDataHeaderTape\_2v0, 126
- headerLength
  - Bds::FileDataHeaderTape\_1v1, 121
  - Bds::FileDataHeaderTape\_2v0, 125
- horizontalAngle
  - Bds::SensorLocation, 154
- htonl
  - canada\_compress.c, 191
- id
  - Bds::AccessGroup, 28

- Bds::Calibration, [47](#)
- Bds::Change, [50](#)
- Bds::ChangeGroup, [52](#)
- Bds::Channel, [54](#)
- Bds::ChannelInstrument, [59](#)
- Bds::DataChannel, [71](#)
- Bds::Digitiser, [117](#)
- Bds::Group, [127](#)
- Bds::Location, [132](#)
- Bds::Network, [135](#)
- Bds::Note, [137](#)
- Bds::Response, [141](#)
- Bds::Sensor, [151](#)
- Bds::SensorLocation, [154](#)
- Bds::Station, [155](#)
- Bds::User, [159](#)
- info
  - Bds::DataBlock, [69](#)
  - Bds::DataChannel, [72](#)
  - Bds::DataInfo, [115](#)
- infoGet
  - Bds::DataFileBds, [92](#)
- infoSet
  - Bds::DataFileBds, [92](#)
- init
  - Bds::DataFile, [78](#)
- initialSamplingFrequency
  - Bds::Digitiser, [118](#)
- latitude
  - Bds::Location, [133](#)
- LDEBUG
  - BdsDataFileBds.cpp, [173](#)
- length
  - Bds::BdsDataBlockHeader, [41](#)
  - Bds::BdsDataPacketHeader, [43](#)
- ListRange
  - Bds::ListRange, [129](#)
- Location
  - Bds::Location, [132](#)
- location
  - Bds::ChannelInfo, [56](#)
- locationDelete
  - Bds::AdminAccess, [34](#)
- locationGetList
  - Bds::AdminAccess, [34](#)
- locationUpdate
  - Bds::AdminAccess, [34](#)
- longitude
  - Bds::Location, [132](#)
- name
  - Bds::Digitiser, [117](#)
  - Bds::Response, [142](#)
  - Bds::Sensor, [151](#)
  - Bds::Station, [155](#)
  - Bds::User, [159](#)
- Network
  - Bds::Network, [135](#)
- network
  - Bds::AccessGroup, [28](#)
  - Bds::Calibration, [47](#)
  - Bds::Channel, [54](#)
  - Bds::DataChannel, [71](#)
  - Bds::Location, [132](#)
  - Bds::Network, [135](#)
  - Bds::Response, [141](#)
  - Bds::SelectionChannel, [147](#)
- networkDelete
  - Bds::AdminAccess, [34](#)
- networkGetList
  - Bds::AdminAccess, [33](#)
  - Bds::DataAccess, [62](#)
  - Bds::DataAddAccess, [66](#)
- networks
  - Bds::SelectionInfo, [149](#)
- networkUpdate
  - Bds::AdminAccess, [34](#)
- Note
  - Bds::Note, [137](#)
- noteDelete
  - Bds::AdminAccess, [38](#)
- noteGetList
  - Bds::AdminAccess, [38](#)
- noteUpdate
  - Bds::AdminAccess, [38](#)
  - Bds::DataAccess, [63](#)
  - Bds::DataAddAccess, [67](#)
- ntohl
  - canada\_compress.c, [191](#)
- number
  - Bds::ListRange, [129](#)
- numberChannels
  - Bds::Digitiser, [117](#)
  - Bds::Sensor, [151](#)
- numBlocks
  - Bds::DataChannel, [72](#)
- numChannels
  - Bds::FileDataHeaderTape\_1v1, [122](#)
  - Bds::FileDataHeaderTape\_2v0, [125](#)
- numDataChannels
  - Bds::SelectionInfo, [149](#)
- numSamples
  - Bds::DataChannel, [72](#)
  - Bds::FileDataHeaderTape\_1v1, [122](#)
  - Bds::FileDataHeaderTape\_2v0, [125](#)
- oblockNumSamples

- Bds::DataFileTapeDigitiser, 104
- oblockPositions
  - Bds::DataFileBdrs, 86
  - Bds::DataFileWra, 108
- oblockSize
  - Bds::DataFileBdrs, 86
  - Bds::DataFileGcf, 98
  - Bds::DataFileTapeDigitiser, 104
  - Bds::DataFileWra, 108
  - Bds::DataFileWraAgso, 111
- oblockYear
  - Bds::DataFileBdrs, 86
  - Bds::DataFileWra, 108
  - Bds::DataFileWraAgso, 111
- obuffer
  - Bds::DataFileTapeDigitiser, 104
- ochannelInfos
  - Bds::DataFileBds, 93
  - Bds::DataFileIms, 101
- ochannelStartBlocks
  - Bds::DataFileWraAgso, 111
- ocurrentBlock
  - Bds::DataFileWraAgso, 111
- odataInfo
  - Bds::DataFileAscii, 82
  - Bds::DataFileBdrs, 86
  - Bds::DataFileBds, 93
  - Bds::DataFileBknas, 96
  - Bds::DataFileGcf, 98
  - Bds::DataFileIms, 101
  - Bds::DataFileWra, 108
  - Bds::DataFileWraAgso, 111
- odiskBlock
  - Bds::DataFileBds, 93
- odiskBlockSize
  - Bds::DataFileBds, 93
- odiskPosition
  - Bds::DataFileBds, 94
- ofieldWidths
  - Bds::DataFileWraAgso, 111
- ofile
  - Bds::DataFile, 80
- ofileName
  - Bds::DataFile, 80
- ofileNameDate
  - Bds::DataFileBdrs, 86
  - Bds::DataFileWra, 108
- ofileNameTime
  - Bds::DataFile, 80
- ofilterCoef
  - Bds::DataFileGcf, 98
- oformat
  - Bds::DataFile, 80
  - Bds::DataFileBds, 93
  - Bds::DataFileGcf, 98
- oheader
  - Bds::DataFileTapeDigitiser, 104
- oheaderSize
  - Bds::DataFileTapeDigitiser, 104
- oinfo
  - Bds::DataFileBds, 93
- oinfoHeader
  - Bds::DataFileBds, 93
- oinfoRepeat
  - Bds::DataFileBds, 93
- oldId
  - Bds::Sensor, 152
- omagic
  - Bds::DataFileBdrs, 86
  - Bds::DataFileWra, 108
- omode
  - Bds::DataFile, 80
- onumChannels
  - Bds::DataFileWra, 108
- opacket
  - Bds::DataFileBds, 94
- opackFormat
  - Bds::DataFileBds, 93
- open
  - Bds::DataFile, 78
  - Bds::DataFileAscii, 81
  - Bds::DataFileBdrs, 84
  - Bds::DataFileBds, 90
  - Bds::DataFileBknas, 95
  - Bds::DataFileIms, 100
  - Bds::DataFileTapeDigitiser, 103
  - Bds::DataFileWra, 106
- oprevValue1
  - Bds::DataFileIms, 101
- oprevValue2
  - Bds::DataFileIms, 101
- OptionFileNameProcess
  - Bds::DataFile, 77
- OptionNone
  - Bds::DataFile, 77
- Options
  - Bds::DataFile, 77
- OptionsList
  - Bds::DataFile, 77
- OptionValidate
  - Bds::DataFile, 77
- OptionValidateCorruptions
  - Bds::DataFile, 77
- oreadBlock
  - Bds::DataFileWraAgso, 111
- oreadChannel
  - Bds::DataFileWraAgso, 111
- osampleRate

- Bds::DataFileBdrs, 86
- Bds::DataFileGcf, 98
- Bds::DataFileTapeDigitiser, 104
- Bds::DataFileWra, 108
- Bds::DataFileWraAgso, 111
- ostreamId
  - Bds::DataFileGcf, 98
- ostreamlets
  - Bds::DataFileBds, 94
- osystemId
  - Bds::DataFileGcf, 98
- oversion
  - Bds::DataFileTapeDigitiser, 104
- overview.dox, 203
- overwriteChannel
  - Bds::DataFileIms, 101
- overwriteChecksum
  - Bds::DataFileIms, 101
- overwriteColumn
  - Bds::DataFileIms, 101
- oyear
  - Bds::DataFileBdrs, 86
  - Bds::DataFileWra, 108
  - Bds::DataFileWraAgso, 111
- pack
  - canada\_compress.c, 191
- packetNumber
  - Bds::BdsDataStreamlet, 45
- packetOffset
  - Bds::BdsDataBlockHeader, 41
- packetRead
  - Bds::DataFileBds, 92
- packetSeek
  - Bds::DataFileBds, 92
- packetWrite
  - Bds::DataFileBds, 92
- PackFormat
  - Bds::DataFileBds, 90
- PackFormat\_CM
  - Bds::DataFileBds, 90
- PackFormat\_SM
  - Bds::DataFileBds, 90
- PackFormat\_SM\_CC
  - Bds::DataFileBds, 90
- PackFormat\_Unknown
  - Bds::DataFileBds, 90
- parseStringFixedFields
  - BdsDataFileWraAgso.cpp, 187
- password
  - Bds::User, 159
- phase
  - Bds::Fap, 119
- Point
  - Bds::Point, 138
- poles
  - Bds::PoleZero, 139
- PoleZero
  - Bds::PoleZero, 139
- poleZeros
  - Bds::Response, 142
- position
  - Bds::BdsDataStreamlet, 45
- Priority
  - Bds, 23
- PriorityHigh
  - Bds, 23
- PriorityLow
  - Bds, 23
- PriorityNormal
  - Bds, 23
- range
  - Bds::Selection, 146
- readBlock
  - Bds::DataFileBdrs, 85
  - Bds::DataFileWra, 107, 108
  - Bds::DataFileWraAgso, 110
- readData
  - Bds::DataCollate, 74
  - Bds::DataFile, 79
  - Bds::DataFileBdrs, 84, 85
  - Bds::DataFileBds, 91
  - Bds::DataFileGcf, 98
  - Bds::DataFileTapeDigitiser, 103
  - Bds::DataFileWra, 107
  - Bds::DataFileWraAgso, 110
- readHeader
  - Bds::DataFileTapeDigitiser, 103
- Response
  - Bds::Response, 141
- responseDelete
  - Bds::AdminAccess, 36
- ResponseEdit
  - Bds::ResponseEdit, 144
- responseGetList
  - Bds::AdminAccess, 36
- responses
  - Bds::ChannelInfo, 57
- responseUpdate
  - Bds::AdminAccess, 36
- reverse
  - Bds::ListRange, 130
- rowId
  - Bds::Change, 50
- SampleFormat
  - Bds, 23

- sampleFormat
  - Bds::DataChannel, 72
- SampleFormatFloat32
  - Bds, 24
- SampleFormatInt16
  - Bds, 24
- SampleFormatInt32
  - Bds, 24
- SampleFormatUnknown
  - Bds, 24
- sampleRate
  - Bds::DataChannel, 72
- samplingFrequency
  - Bds::Calibration, 47
- Scale
  - Bds, 25
- seekBlock
  - Bds::DataFile, 79
  - Bds::DataFileBds, 91
- segmentNumber
  - Bds::DataBlock, 69
- Selection
  - Bds::Selection, 146
- SelectionChannel
  - Bds::SelectionChannel, 147
- SelectionGroup
  - Bds, 24
- SelectionGroupData
  - Bds, 24
- SelectionGroupMetaData
  - Bds, 24
- SelectionInfo
  - Bds::SelectionInfo, 149
- Sensor
  - Bds::Sensor, 151
- sensor
  - Bds::ChannelInfo, 57
- sensorDelete
  - Bds::AdminAccess, 35
- sensorGet
  - Bds::AdminAccess, 35
- sensorGetList
  - Bds::AdminAccess, 35
- sensorId
  - Bds::ChannelInstrument, 59
  - Bds::Selection, 146
  - Bds::SensorLocation, 154
- SensorLocation
  - Bds::SensorLocation, 154
- sensorLocation
  - Bds::ChannelInfo, 57
- sensorLocationDelete
  - Bds::AdminAccess, 35
- sensorLocationGetList
  - Bds::AdminAccess, 35
- sensorLocationUpdate
  - Bds::AdminAccess, 35
- sensorOldId
  - Bds::Selection, 146
- sensorUpdate
  - Bds::AdminAccess, 35
- sequence
  - Bds::BdsDataPacketHeader, 43
- serialNumber
  - Bds::Digitiser, 117
  - Bds::Sensor, 151
- setChecksumAndLength
  - Bds::BdsDataPacket, 42
- setDiskBlockSize
  - Bds::DataFileBds, 91
- setFormat
  - Bds::DataFile, 78
  - Bds::DataFileBds, 90
  - Bds::DataFileWra, 106, 107
- setHeader
  - Bds::BdsDataPacket, 42
- setInfo
  - Bds::DataFile, 78
  - Bds::DataFileAscii, 81
  - Bds::DataFileBds, 90
  - Bds::DataFileBknas, 95
  - Bds::DataFileIms, 100
- setInfoRepeat
  - Bds::DataFileBds, 92
- setMembers
  - Bds::AccessGroup, 28
  - Bds::Calibration, 47
  - Bds::Change, 50
  - Bds::ChangeGroup, 52
  - Bds::Channel, 54
  - Bds::ChannelInstrument, 59
  - Bds::DataChannel, 71
  - Bds::Digitiser, 117
  - Bds::Group, 127
  - Bds::ListRange, 129
  - Bds::Location, 132
  - Bds::Network, 135
  - Bds::Note, 137
  - Bds::Sensor, 151
  - Bds::SensorLocation, 154
  - Bds::TimePeriod, 157
  - Bds::User, 159
- setString
  - Bds::ResponseEdit, 144
- setUser
  - Bds::AdminAccess, 32
  - Bds::DataAccess, 61
  - Bds::DataAddAccess, 65

- setUserReal
  - Bds::AdminAccess, 32
  - Bds::DataAccess, 61
  - Bds::DataAddAccess, 65
- source
  - Bds::DataChannel, 72
  - Bds::SelectionChannel, 147
- sources
  - Bds::SelectionInfo, 149
- sqlQuery
  - Bds::AdminAccess, 38
- start
  - Bds::DataFile, 78
  - Bds::DataFileIms, 100
  - Bds::ListRange, 129
- startBlock
  - Bds::DataHandle, 113
- startTime
  - Bds::AccessGroup, 28
  - Bds::BdsDataPacketHeader, 44
  - Bds::Calibration, 47
  - Bds::Channel, 54
  - Bds::ChannelInstrument, 59
  - Bds::DataBlock, 68
  - Bds::DataChannel, 71
  - Bds::DataInfo, 114
  - Bds::Digitiser, 117
  - Bds::FileDataHeaderTape\_1v1, 122
  - Bds::FileDataHeaderTape\_2v0, 125
  - Bds::Location, 132
  - Bds::Response, 141
  - Bds::Selection, 146
  - Bds::SelectionInfo, 149
  - Bds::Sensor, 151
  - Bds::SensorLocation, 154
  - Bds::TimePeriod, 157
- Station
  - Bds::Station, 155
- station
  - Bds::AccessGroup, 28
  - Bds::ArrayChannel, 39
  - Bds::Calibration, 47
  - Bds::Channel, 54
  - Bds::ChannelInfo, 56
  - Bds::DataChannel, 72
  - Bds::Location, 132
  - Bds::Response, 142
  - Bds::SelectionChannel, 147
- stationDelete
  - Bds::AdminAccess, 34
- stationGetList
  - Bds::AdminAccess, 34
  - Bds::DataAccess, 62
  - Bds::DataAddAccess, 66
- stations
  - Bds::Network, 135
  - Bds::SelectionInfo, 149
- stationUpdate
  - Bds::AdminAccess, 34
- statisticsGet
  - Bds::AdminAccess, 38
- streamlet
  - Bds::BdsDataPacketHeader, 43
- StreamsMax
  - Bds::DataFileBds, 90
- symmetry
  - Bds::Response, 142
- synchronous
  - Bds::DataInfo, 115
- table
  - Bds::Change, 50
- tapeFmSignalLevels
  - Bds::FileDataHeaderTape\_1v1, 122
  - Bds::FileDataHeaderTape\_2v0, 126
- tapeLastVelaCode
  - Bds::FileDataHeaderTape\_1v1, 122
  - Bds::FileDataHeaderTape\_2v0, 125
- tapeSignalLevelErrors
  - Bds::FileDataHeaderTape\_1v1, 122
  - Bds::FileDataHeaderTape\_2v0, 125
- tapeSpeedVariationErrors
  - Bds::FileDataHeaderTape\_1v1, 122
  - Bds::FileDataHeaderTape\_2v0, 125
- tapeVelaBitErrors
  - Bds::FileDataHeaderTape\_1v1, 122
  - Bds::FileDataHeaderTape\_2v0, 125
- telephone
  - Bds::User, 159
- time
  - Bds::Change, 50
  - Bds::ChangeGroup, 52
  - Bds::Note, 137
- TimePeriod
  - Bds::TimePeriod, 157
- title
  - Bds::ChangeGroup, 52
  - Bds::Note, 137
- type
  - Bds::BdsDataBlockHeader, 41
  - Bds::BdsDataPacketHeader, 43
  - Bds::Change, 50
  - Bds::ChangeGroup, 52
  - Bds::Digitiser, 117
  - Bds::FileDataHeaderTape\_1v1, 121
  - Bds::FileDataHeaderTape\_2v0, 125
  - Bds::Note, 137
  - Bds::Response, 142

- Bds::Sensor, [151](#)
- Bds::Station, [155](#)
- unpack
  - canada\_compress.c, [191](#)
- User
  - Bds::User, [159](#)
- user
  - Bds::ChangeGroup, [52](#)
  - Bds::Note, [137](#)
  - Bds::User, [159](#)
- userDelete
  - Bds::AdminAccess, [33](#)
- userGetGroups
  - Bds::AdminAccess, [33](#)
  - Bds::DataAccess, [62](#)
  - Bds::DataAddAccess, [66](#)
- userGetList
  - Bds::AdminAccess, [32](#)
- userUpdate
  - Bds::AdminAccess, [32](#)
- validateChecksum
  - Bds::BdsDataPacket, [42](#)
- verticalAngle
  - Bds::SensorLocation, [154](#)
- writeData
  - Bds::DataFile, [79](#)
  - Bds::DataFileAscii, [82](#)
  - Bds::DataFileBds, [91](#)
  - Bds::DataFileBknas, [96](#)
  - Bds::DataFileIms, [100](#)
- x
  - Bds::Point, [138](#)
- y
  - Bds::Point, [138](#)
- zeros
  - Bds::PoleZero, [139](#)