

BdsApi

2.2.0

Generated by Doxygen 1.8.15







<b>1 Main Page</b>	<b>1</b>
1.1 Introduction	1
1.2 Overview	2
1.3 C++ Examples	2
1.4 Python Examples	5
<b>2 Namespace Index</b>	<b>9</b>
2.1 Namespace List	9
<b>3 Hierarchical Index</b>	<b>11</b>
3.1 Class Hierarchy	11
<b>4 Class Index</b>	<b>13</b>
4.1 Class List	13
<b>5 File Index</b>	<b>17</b>
5.1 File List	17
<b>6 Namespace Documentation</b>	<b>19</b>
6.1 Bds Namespace Reference	19
6.1.1 Enumeration Type Documentation	25
6.1.1.1 AvailType	25
6.1.1.2 BdsDataType	25
6.1.1.3 DataFlags	26
6.1.1.4 Errors	26
6.1.1.5 FileHeaderType	27
6.1.1.6 FileSampleType	27
6.1.1.7 Mode	27
6.1.1.8 Priority	28
6.1.1.9 SampleFormat	28
6.1.1.10 SelectionGroup	28
6.1.2 Function Documentation	28
6.1.2.1 bdsChannelGetName()	29
6.1.2.2 bdsChannelGetTypeAux()	29
6.1.2.3 bdsDataChannelInfo()	29
6.1.2.4 bdsDataChannelOverallResponse()	29
6.1.2.5 bdsDataChannelRef() [1/2]	30
6.1.2.6 bdsDataChannelRef() [2/2]	30
6.1.2.7 bdsDataFileSeedLogError()	30
6.1.2.8 bdsDataFileSeedLogWarning()	30
6.1.2.9 bdsDataInfoFromInfo()	30
6.1.2.10 bdsDataInfoMergeFlatten()	30
6.1.2.11 bdsDataInfoSetTimeRange()	31
6.1.2.12 bdsDumpChannelInfos()	31



6.1.2.13 bdsDumpData()	31
6.1.2.14 bdsDumpDataInfo()	31
6.1.2.15 bdsDumpPoleZeros()	31
6.1.2.16 bdsDumpSelection()	32
6.1.2.17 bdsFileNameExpand() [1/2]	32
6.1.2.18 bdsFileNameExpand() [2/2]	32
6.1.2.19 bdsInfoFromDataInfo()	32
6.1.2.20 bdsPoleZeroGain()	32
6.1.2.21 bdsPoleZeroGainPhase()	33
6.1.2.22 bdsPoleZeroToFap()	33
6.1.2.23 bdsSelectionChannelInfo()	33
6.1.2.24 bdsStationAlias()	33
6.1.2.25 bdsUnCompressCm8()	33
6.1.2.26 bdsUnCompressSteim1()	34
6.1.2.27 crc()	34
6.1.2.28 crc64()	34
6.1.2.29 crcInit()	34
6.1.2.30 dataCalculateDifference()	34
6.1.2.31 dataCalculateUnDifference()	34
6.1.2.32 dataChecksum()	35
6.1.2.33 dataCompressCm6()	35
6.1.2.34 dataConvert() [1/3]	35
6.1.2.35 dataConvert() [2/3]	35
6.1.2.36 dataConvert() [3/3]	35
6.1.2.37 dataDeCompressCm6()	35
6.1.2.38 duplicateDump()	36
6.1.2.39 fileNameTime()	36
6.1.2.40 fixedString()	36
6.1.2.41 fixedWidthValue()	36
6.1.2.42 getHexString()	36
6.1.2.43 nullString()	36
6.1.2.44 record_handler()	37
6.1.2.45 removeCR()	37
6.1.2.46 seedTime()	37
6.1.2.47 seedTimeString()	37
6.1.2.48 stringFormat()	37
6.1.3 Variable Documentation	37
6.1.3.1 apiVersion	37
6.1.3.2 BdsDataFileVersion	38
6.1.3.3 ChannelAuxLen	38
6.1.3.4 ChannelTypeLen	38
6.1.3.5 cm6Table	38



6.1.3.6 cm6TableRev . . . . .	38
6.1.3.7 crcInitDone . . . . .	39
6.1.3.8 crcVec . . . . .	39
6.1.3.9 dataFormats . . . . .	39
6.1.3.10 NetworkNameLen . . . . .	39
6.1.3.11 Scale . . . . .	39
6.1.3.12 SourceLen . . . . .	39
6.1.3.13 StationNameLen . . . . .	39
<b>7 Class Documentation</b>	<b>41</b>
7.1 Bds::AccessGroup Class Reference . . . . .	41
7.1.1 Detailed Description . . . . .	42
7.1.2 Constructor & Destructor Documentation . . . . .	42
7.1.2.1 AccessGroup() . . . . .	42
7.1.3 Member Function Documentation . . . . .	42
7.1.3.1 getMember() . . . . .	42
7.1.3.2 getMembers() . . . . .	42
7.1.3.3 getType() . . . . .	42
7.1.3.4 setMember() . . . . .	43
7.1.3.5 setMembers() . . . . .	43
7.1.4 Member Data Documentation . . . . .	43
7.1.4.1 endTime . . . . .	43
7.1.4.2 group . . . . .	43
7.1.4.3 id . . . . .	43
7.1.4.4 network . . . . .	44
7.1.4.5 startTime . . . . .	44
7.1.4.6 station . . . . .	44
7.2 Bds::AdminAccess Class Reference . . . . .	44
7.2.1 Detailed Description . . . . .	49
7.2.2 Constructor & Destructor Documentation . . . . .	49
7.2.2.1 AdminAccess() . . . . .	49
7.2.3 Member Function Documentation . . . . .	49
7.2.3.1 accessGroupDelete() . . . . .	49
7.2.3.2 accessGroupGetList() . . . . .	49
7.2.3.3 accessGroupUpdate() . . . . .	50
7.2.3.4 calibrationDelete() . . . . .	50
7.2.3.5 calibrationGetList() . . . . .	50
7.2.3.6 calibrationUpdate() . . . . .	50
7.2.3.7 changeDelete() . . . . .	50
7.2.3.8 changeGetList() . . . . .	51
7.2.3.9 changeGetListNumber() . . . . .	51
7.2.3.10 changeGroupDelete() . . . . .	51



7.2.3.11	changeGroupEnd()	51
7.2.3.12	changeGroupGetList()	51
7.2.3.13	changeGroupStart()	52
7.2.3.14	channelDelete()	52
7.2.3.15	channelGet()	52
7.2.3.16	channelGetList()	52
7.2.3.17	channelInstrumentDelete()	52
7.2.3.18	channelInstrumentGetList()	53
7.2.3.19	channelInstrumentUpdate()	53
7.2.3.20	channelUpdate()	53
7.2.3.21	clean()	53
7.2.3.22	connect()	53
7.2.3.23	dataAvailability()	54
7.2.3.24	databaseBackup()	54
7.2.3.25	databaseRestore()	54
7.2.3.26	dataChannelDelete()	54
7.2.3.27	dataChannelGetList()	54
7.2.3.28	dataChannelUpdate()	55
7.2.3.29	dataClose()	55
7.2.3.30	dataFileDelete()	55
7.2.3.31	dataFileGetList()	55
7.2.3.32	dataFileUpdate()	55
7.2.3.33	dataFormatGetList()	56
7.2.3.34	dataFormattedGetLength()	56
7.2.3.35	dataFormattedRead()	56
7.2.3.36	dataGetBlock()	56
7.2.3.37	dataGetChannelInfo()	56
7.2.3.38	dataGetInfo()	57
7.2.3.39	dataGetNotes()	57
7.2.3.40	dataGetWarnings()	57
7.2.3.41	dataOpen()	57
7.2.3.42	dataPutBlock()	58
7.2.3.43	dataSearch()	58
7.2.3.44	dataSeekBlock()	58
7.2.3.45	dataSetInfo()	58
7.2.3.46	digitiserDelete()	58
7.2.3.47	digitiserGet()	59
7.2.3.48	digitiserGetList()	59
7.2.3.49	digitiserUpdate()	59
7.2.3.50	getSelectionInfo()	59
7.2.3.51	getSelections()	59
7.2.3.52	getVersion()	60



7.2.3.53 groupDelete()	60
7.2.3.54 groupGetList()	60
7.2.3.55 groupUpdate()	60
7.2.3.56 locationDelete()	60
7.2.3.57 locationGetList()	61
7.2.3.58 locationUpdate()	61
7.2.3.59 logAppend()	61
7.2.3.60 logDelete()	61
7.2.3.61 logGetList()	61
7.2.3.62 logUpdate()	62
7.2.3.63 modeSet()	62
7.2.3.64 modeSnapshotPause()	62
7.2.3.65 networkDelete()	62
7.2.3.66 networkGetList()	62
7.2.3.67 networkUpdate()	63
7.2.3.68 noteDelete()	63
7.2.3.69 noteGetList()	63
7.2.3.70 noteReadDocument()	63
7.2.3.71 noteUpdate()	63
7.2.3.72 noteWriteDocument()	64
7.2.3.73 responseDelete()	64
7.2.3.74 responseGetList()	64
7.2.3.75 responseUpdate()	64
7.2.3.76 sensorDelete()	64
7.2.3.77 sensorGet()	65
7.2.3.78 sensorGetList()	65
7.2.3.79 sensorUpdate()	65
7.2.3.80 setUser()	65
7.2.3.81 setUserReal()	65
7.2.3.82 sourceDelete()	66
7.2.3.83 sourceGetList()	66
7.2.3.84 sourcePriorityDelete()	66
7.2.3.85 sourcePriorityGetList()	66
7.2.3.86 sourcePriorityUpdate()	66
7.2.3.87 sourceUpdate()	66
7.2.3.88 sqlQuery()	67
7.2.3.89 stationDelete()	67
7.2.3.90 stationGetList()	67
7.2.3.91 stationUpdate()	67
7.2.3.92 statisticsGet()	67
7.2.3.93 transactionEnd()	68
7.2.3.94 transactionStart()	68



7.2.3.95 userDelete()	68
7.2.3.96 userGet()	68
7.2.3.97 userGetFromId()	68
7.2.3.98 userGetGroups()	68
7.2.3.99 userGetList()	69
7.2.3.100 userSet()	69
7.2.3.101 userUpdate()	69
7.2.3.102 validateUser()	69
7.3 Bds::ArrayChannel Class Reference	69
7.3.1 Detailed Description	70
7.3.2 Constructor & Destructor Documentation	70
7.3.2.1 ArrayChannel()	70
7.3.3 Member Data Documentation	70
7.3.3.1 channel	70
7.3.3.2 station	70
7.4 Bds::BdsDataBlock Struct Reference	71
7.4.1 Detailed Description	71
7.4.2 Member Data Documentation	71
7.4.2.1 data	71
7.4.2.2 header	71
7.5 Bds::BdsDataBlockHeader Struct Reference	71
7.5.1 Detailed Description	72
7.5.2 Member Data Documentation	72
7.5.2.1 length	72
7.5.2.2 packetOffset	72
7.5.2.3 type	72
7.6 Bds::BdsDataBlockPos Class Reference	73
7.6.1 Detailed Description	73
7.6.2 Constructor & Destructor Documentation	73
7.6.2.1 BdsDataBlockPos()	73
7.6.3 Member Function Documentation	73
7.6.3.1 operator<()	74
7.6.4 Member Data Documentation	74
7.6.4.1 channel	74
7.6.4.2 endTime	74
7.6.4.3 numChannels	74
7.6.4.4 numSamples	74
7.6.4.5 position	74
7.6.4.6 segment	74
7.6.4.7 startTime	75
7.7 Bds::BdsDataPacket Class Reference	75
7.7.1 Detailed Description	75



7.7.2 Constructor & Destructor Documentation	75
7.7.2.1 BdsDataPacket()	76
7.7.2.2 ~BdsDataPacket()	76
7.7.3 Member Function Documentation	76
7.7.3.1 clear()	76
7.7.3.2 dump()	76
7.7.3.3 getHeader()	76
7.7.3.4 reset()	76
7.7.3.5 setChecksumAndLength()	76
7.7.3.6 setHeader()	77
7.7.3.7 validateChecksum()	77
7.8 Bds::BdsDataPacketHeader Struct Reference	77
7.8.1 Detailed Description	77
7.8.2 Member Data Documentation	77
7.8.2.1 checksum	78
7.8.2.2 endTime	78
7.8.2.3 length	78
7.8.2.4 sequence	78
7.8.2.5 startTime	78
7.8.2.6 streamlet	78
7.8.2.7 type	79
7.9 Bds::BdsDataSegment Class Reference	79
7.9.1 Detailed Description	79
7.9.2 Constructor & Destructor Documentation	79
7.9.2.1 BdsDataSegment()	79
7.9.3 Member Function Documentation	79
7.9.3.1 operator<()	80
7.9.4 Member Data Documentation	80
7.9.4.1 blocks	80
7.9.4.2 endTime	80
7.9.4.3 numBlocks	80
7.9.4.4 numSamples	80
7.9.4.5 sampleRate	80
7.9.4.6 startTime	80
7.10 Bds::BdsDataStreamlet Class Reference	81
7.10.1 Detailed Description	81
7.10.2 Constructor & Destructor Documentation	81
7.10.2.1 BdsDataStreamlet()	81
7.10.3 Member Data Documentation	81
7.10.3.1 blocks	81
7.10.3.2 channel	82
7.10.3.3 numChannels	82



7.10.3.4 packetNumber . . . . .	82
7.10.3.5 position . . . . .	82
7.10.3.6 segments . . . . .	82
7.11 Bds::BdsSeedType Class Reference . . . . .	82
7.11.1 Detailed Description . . . . .	83
7.11.2 Constructor & Destructor Documentation . . . . .	83
7.11.2.1 BdsSeedType() . . . . .	83
7.11.3 Member Function Documentation . . . . .	83
7.11.3.1 appendDouble() . . . . .	83
7.11.3.2 appendExp() . . . . .	83
7.11.3.3 appendInt() . . . . .	84
7.11.3.4 appendString() . . . . .	84
7.11.3.5 appendStringVariable() . . . . .	84
7.11.3.6 getDouble() . . . . .	84
7.11.3.7 getInt() . . . . .	84
7.11.3.8 getString() . . . . .	84
7.11.3.9 getStringVariable() . . . . .	85
7.11.3.10 getUInt() . . . . .	85
7.12 Bds::Calibration Class Reference . . . . .	85
7.12.1 Detailed Description . . . . .	86
7.12.2 Constructor & Destructor Documentation . . . . .	86
7.12.2.1 Calibration() . . . . .	87
7.12.3 Member Function Documentation . . . . .	87
7.12.3.1 getMember() . . . . .	87
7.12.3.2 getMembers() . . . . .	87
7.12.3.3 getType() . . . . .	87
7.12.3.4 setMember() . . . . .	88
7.12.3.5 setMembers() . . . . .	88
7.12.4 Member Data Documentation . . . . .	88
7.12.4.1 calibrationFactor . . . . .	88
7.12.4.2 calibrationFrequency . . . . .	88
7.12.4.3 calibrationUnits . . . . .	88
7.12.4.4 channel . . . . .	89
7.12.4.5 depth . . . . .	89
7.12.4.6 endTime . . . . .	89
7.12.4.7 horizontalAngle . . . . .	89
7.12.4.8 id . . . . .	89
7.12.4.9 name . . . . .	89
7.12.4.10 network . . . . .	90
7.12.4.11 samplingFrequency . . . . .	90
7.12.4.12 source . . . . .	90
7.12.4.13 startTime . . . . .	90



7.12.4.14 station	90
7.12.4.15 verticalAngle	90
7.13 Bds::CdChannel_1v0 Struct Reference	91
7.13.1 Detailed Description	91
7.13.2 Member Data Documentation	91
7.13.2.1 auth	91
7.13.2.2 calibrationFactor	91
7.13.2.3 calibrationPeriod	91
7.13.2.4 channel	92
7.13.2.5 compress	92
7.13.2.6 name	92
7.13.2.7 spare0	92
7.13.2.8 spare1	92
7.14 Bds::CdDataChannel Class Reference	92
7.14.1 Detailed Description	93
7.14.2 Member Data Documentation	93
7.14.2.1 channel	93
7.14.2.2 data	93
7.14.2.3 dataSize	93
7.14.2.4 mode	93
7.14.2.5 numSamples	93
7.14.2.6 period	93
7.14.2.7 startTime	94
7.14.2.8 station	94
7.14.2.9 status	94
7.15 Bds::CdDataFormatFrame_1v0 Struct Reference	94
7.15.1 Detailed Description	94
7.15.2 Member Data Documentation	94
7.15.2.1 channels	95
7.15.2.2 frameLength	95
7.15.2.3 frameType	95
7.15.2.4 maxFrameLength	95
7.15.2.5 numChannels	95
7.15.2.6 period	95
7.16 Bds::CdFlag Class Reference	95
7.16.1 Detailed Description	96
7.16.2 Constructor & Destructor Documentation	96
7.16.2.1 CdFlag()	96
7.16.3 Member Data Documentation	96
7.16.3.1 dead	96
7.16.3.2 zeroed	96
7.17 Bds::CdPacketData Class Reference	96



7.17.1 Detailed Description	97
7.17.2 Member Data Documentation	97
7.17.2.1 auth	97
7.17.2.2 authKey	97
7.17.2.3 authSize	97
7.17.2.4 channels	97
7.17.2.5 crc	98
7.17.2.6 creator	98
7.17.2.7 destination	98
7.17.2.8 frameType	98
7.17.2.9 numChannels	98
7.17.2.10 period	98
7.17.2.11 sequenceNum	98
7.17.2.12 series	98
7.17.2.13 startTime	99
7.17.2.14 trailerOffset	99
7.18 Bds::Change Class Reference	99
7.18.1 Detailed Description	100
7.18.2 Constructor & Destructor Documentation	100
7.18.2.1 Change()	100
7.18.3 Member Function Documentation	100
7.18.3.1 getMember()	100
7.18.3.2 getMembers()	100
7.18.3.3 getType()	101
7.18.3.4 setMember()	101
7.18.3.5 setMembers()	101
7.18.4 Member Data Documentation	101
7.18.4.1 changeGroupId	101
7.18.4.2 id	101
7.18.4.3 rowId	101
7.18.4.4 table	102
7.18.4.5 time	102
7.18.4.6 type	102
7.19 Bds::ChangeGroup Class Reference	102
7.19.1 Detailed Description	103
7.19.2 Constructor & Destructor Documentation	103
7.19.2.1 ChangeGroup()	103
7.19.3 Member Function Documentation	103
7.19.3.1 getMember()	103
7.19.3.2 getMembers()	104
7.19.3.3 getType()	104
7.19.3.4 setMember()	104



7.19.3.5 setMembers()	104
7.19.4 Member Data Documentation	104
7.19.4.1 description	104
7.19.4.2 id	105
7.19.4.3 time	105
7.19.4.4 title	105
7.19.4.5 type	105
7.19.4.6 user	105
7.20 Bds::Channel Class Reference	106
7.20.1 Detailed Description	107
7.20.2 Constructor & Destructor Documentation	107
7.20.2.1 Channel()	107
7.20.3 Member Function Documentation	107
7.20.3.1 getMember()	107
7.20.3.2 getMembers()	107
7.20.3.3 getType()	108
7.20.3.4 setMember()	108
7.20.3.5 setMembers()	108
7.20.4 Member Data Documentation	108
7.20.4.1 channel	108
7.20.4.2 channelAux	108
7.20.4.3 channelType	108
7.20.4.4 dataType	109
7.20.4.5 description	109
7.20.4.6 endTime	109
7.20.4.7 id	109
7.20.4.8 network	109
7.20.4.9 startTime	109
7.20.4.10 station	110
7.21 Bds::ChannelInfo Class Reference	110
7.21.1 Detailed Description	111
7.21.2 Constructor & Destructor Documentation	111
7.21.2.1 ChannelInfo()	111
7.21.3 Member Data Documentation	111
7.21.3.1 calibration	111
7.21.3.2 channel	111
7.21.3.3 dataType	112
7.21.3.4 digitiser	112
7.21.3.5 endTime	112
7.21.3.6 location	112
7.21.3.7 responses	112
7.21.3.8 sensor	112



---

7.21.3.9 source	113
7.21.3.10 startTime	113
7.21.3.11 station	113
7.22 Bds::ChannellInfos Class Reference	113
7.22.1 Detailed Description	113
7.22.2 Constructor & Destructor Documentation	114
7.22.2.1 ChannellInfos()	114
7.22.3 Member Data Documentation	114
7.22.3.1 channels	114
7.23 Bds::ChannellInstrument Class Reference	114
7.23.1 Detailed Description	115
7.23.2 Constructor & Destructor Documentation	115
7.23.2.1 ChannellInstrument()	115
7.23.3 Member Function Documentation	115
7.23.3.1 getMember()	116
7.23.3.2 getMembers()	116
7.23.3.3 getType()	116
7.23.3.4 setMember()	116
7.23.3.5 setMembers()	116
7.23.4 Member Data Documentation	116
7.23.4.1 channelId	117
7.23.4.2 digitiserId	117
7.23.4.3 endTime	117
7.23.4.4 id	117
7.23.4.5 sensorId	117
7.23.4.6 source	117
7.23.4.7 startTime	118
7.24 Bds::ChannelName Class Reference	118
7.24.1 Detailed Description	118
7.24.2 Constructor & Destructor Documentation	118
7.24.2.1 ChannelName()	119
7.24.3 Member Data Documentation	119
7.24.3.1 channel	119
7.24.3.2 network	119
7.24.3.3 source	119
7.24.3.4 station	119
7.25 Bds::CleanOptions Class Reference	120
7.25.1 Detailed Description	120
7.25.2 Constructor & Destructor Documentation	120
7.25.2.1 CleanOptions()	120
7.25.3 Member Data Documentation	120
7.25.3.1 changes	120

---



7.25.3.2 deletedFiles . . . . .	121
7.25.3.3 logs . . . . .	121
7.26 Bds::CompressSteim1 Class Reference . . . . .	121
7.26.1 Detailed Description . . . . .	121
7.26.2 Constructor & Destructor Documentation . . . . .	121
7.26.2.1 CompressSteim1() . . . . .	121
7.26.3 Member Function Documentation . . . . .	122
7.26.3.1 clear() . . . . .	122
7.26.3.2 setByteOrder() . . . . .	122
7.26.3.3 unCompress() . . . . .	122
7.27 Bds::DataAccess Class Reference . . . . .	122
7.27.1 Detailed Description . . . . .	125
7.27.2 Constructor & Destructor Documentation . . . . .	125
7.27.2.1 DataAccess() . . . . .	125
7.27.3 Member Function Documentation . . . . .	125
7.27.3.1 calibrationGetList() . . . . .	125
7.27.3.2 channelGetList() . . . . .	125
7.27.3.3 channelInstrumentGetList() . . . . .	126
7.27.3.4 clean() . . . . .	126
7.27.3.5 connect() . . . . .	126
7.27.3.6 dataAvailability() . . . . .	126
7.27.3.7 databaseBackup() . . . . .	126
7.27.3.8 dataChannelGetList() . . . . .	127
7.27.3.9 dataClose() . . . . .	127
7.27.3.10 dataFileGetList() . . . . .	127
7.27.3.11 dataFormatGetList() . . . . .	127
7.27.3.12 dataFormattedGetLength() . . . . .	127
7.27.3.13 dataFormattedRead() . . . . .	128
7.27.3.14 dataGetBlock() . . . . .	128
7.27.3.15 dataGetChannelInfo() . . . . .	128
7.27.3.16 dataGetInfo() . . . . .	128
7.27.3.17 dataGetNotes() . . . . .	128
7.27.3.18 dataGetWarnings() . . . . .	129
7.27.3.19 dataOpen() . . . . .	129
7.27.3.20 dataSearch() . . . . .	129
7.27.3.21 dataSeekBlock() . . . . .	129
7.27.3.22 digitiserGet() . . . . .	130
7.27.3.23 digitiserGetList() . . . . .	130
7.27.3.24 getSelectionInfo() . . . . .	130
7.27.3.25 getSelections() . . . . .	130
7.27.3.26 getVersion() . . . . .	130
7.27.3.27 groupGetList() . . . . .	131



7.27.3.28 locationGetList()	131
7.27.3.29 logAppend()	131
7.27.3.30 logUpdate()	131
7.27.3.31 modeSet()	131
7.27.3.32 modeSnapshotPause()	132
7.27.3.33 networkGetList()	132
7.27.3.34 noteGetList()	132
7.27.3.35 noteReadDocument()	132
7.27.3.36 noteUpdate()	132
7.27.3.37 noteWriteDocument()	133
7.27.3.38 responseGetList()	133
7.27.3.39 sensorGet()	133
7.27.3.40 sensorGetList()	133
7.27.3.41 setUser()	133
7.27.3.42 setUserReal()	134
7.27.3.43 sourceGetList()	134
7.27.3.44 sourcePriorityGetList()	134
7.27.3.45 stationGetList()	134
7.27.3.46 statisticsGet()	134
7.27.3.47 userGet()	134
7.27.3.48 userGetFromId()	135
7.27.3.49 userGetGroups()	135
7.27.3.50 userSet()	135
7.27.3.51 validateUser()	135
7.28 Bds::DataAddAccess Class Reference	135
7.28.1 Detailed Description	138
7.28.2 Constructor & Destructor Documentation	138
7.28.2.1 DataAddAccess()	138
7.28.3 Member Function Documentation	138
7.28.3.1 calibrationGetList()	138
7.28.3.2 channelGetList()	139
7.28.3.3 channelInstrumentGetList()	139
7.28.3.4 clean()	139
7.28.3.5 connect()	139
7.28.3.6 dataAvailability()	139
7.28.3.7 databaseBackup()	140
7.28.3.8 dataChannelGetList()	140
7.28.3.9 dataClose()	140
7.28.3.10 dataFileGetList()	140
7.28.3.11 dataFormatGetList()	140
7.28.3.12 dataFormattedGetLength()	141
7.28.3.13 dataFormattedRead()	141



7.28.3.14 dataGetBlock()	141
7.28.3.15 dataGetChannelInfo()	141
7.28.3.16 dataGetInfo()	141
7.28.3.17 dataGetNotes()	142
7.28.3.18 dataGetWarnings()	142
7.28.3.19 dataOpen()	142
7.28.3.20 dataPutBlock()	142
7.28.3.21 dataSearch()	143
7.28.3.22 dataSeekBlock()	143
7.28.3.23 dataSetInfo()	143
7.28.3.24 digitiserGet()	143
7.28.3.25 digitiserGetList()	143
7.28.3.26 getSelectionInfo()	144
7.28.3.27 getSelections()	144
7.28.3.28 getVersion()	144
7.28.3.29 groupGetList()	144
7.28.3.30 locationGetList()	144
7.28.3.31 logAppend()	145
7.28.3.32 logUpdate()	145
7.28.3.33 modeSet()	145
7.28.3.34 modeSnapshotPause()	145
7.28.3.35 networkGetList()	145
7.28.3.36 noteGetList()	146
7.28.3.37 noteReadDocument()	146
7.28.3.38 noteUpdate()	146
7.28.3.39 noteWriteDocument()	146
7.28.3.40 responseGetList()	146
7.28.3.41 sensorGet()	147
7.28.3.42 sensorGetList()	147
7.28.3.43 setUser()	147
7.28.3.44 setUserReal()	147
7.28.3.45 sourceGetList()	147
7.28.3.46 sourcePriorityGetList()	148
7.28.3.47 stationGetList()	148
7.28.3.48 statisticsGet()	148
7.28.3.49 userGet()	148
7.28.3.50 userGetFromId()	148
7.28.3.51 userGetGroups()	149
7.28.3.52 userSet()	149
7.28.3.53 validateUser()	149
7.29 Bds::DataAvail Class Reference	149
7.29.1 Detailed Description	150



7.29.2 Constructor & Destructor Documentation	150
7.29.2.1 DataAvail()	150
7.29.3 Member Data Documentation	150
7.29.3.1 availType	150
7.29.3.2 endTime	150
7.29.3.3 startTime	150
7.30 Bds::DataAvailChan Class Reference	151
7.30.1 Detailed Description	151
7.30.2 Constructor & Destructor Documentation	151
7.30.2.1 DataAvailChan()	152
7.30.3 Member Data Documentation	152
7.30.3.1 channel	152
7.30.3.2 endTime	152
7.30.3.3 network	152
7.30.3.4 segments	152
7.30.3.5 source	153
7.30.3.6 startTime	153
7.30.3.7 station	153
7.31 Bds::DataBlock Class Reference	153
7.31.1 Detailed Description	154
7.31.2 Constructor & Destructor Documentation	154
7.31.2.1 DataBlock()	154
7.31.3 Member Data Documentation	154
7.31.3.1 channelData	154
7.31.3.2 channelNumber	154
7.31.3.3 endTime	155
7.31.3.4 info	155
7.31.3.5 segmentNumber	155
7.31.3.6 startTime	155
7.32 Bds::DataBlockPos Class Reference	155
7.32.1 Detailed Description	156
7.32.2 Constructor & Destructor Documentation	156
7.32.2.1 DataBlockPos()	156
7.32.3 Member Function Documentation	156
7.32.3.1 operator<()	156
7.32.4 Member Data Documentation	156
7.32.4.1 endTime	156
7.32.4.2 numSamples	157
7.32.4.3 order	157
7.32.4.4 position	157
7.32.4.5 ref	157
7.32.4.6 startTime	157



7.33 Bds::DataChannel Class Reference	157
7.33.1 Detailed Description	159
7.33.2 Constructor & Destructor Documentation	159
7.33.2.1 DataChannel()	159
7.33.3 Member Function Documentation	159
7.33.3.1 getMember()	159
7.33.3.2 getMembers()	160
7.33.3.3 getType()	160
7.33.3.4 setMember()	160
7.33.3.5 setMembers()	160
7.33.4 Member Data Documentation	160
7.33.4.1 channel	160
7.33.4.2 dataFileChannel	161
7.33.4.3 dataFileId	161
7.33.4.4 endTime	161
7.33.4.5 id	161
7.33.4.6 importFilename	161
7.33.4.7 importFormat	161
7.33.4.8 importStartTime	162
7.33.4.9 info	162
7.33.4.10 network	162
7.33.4.11 numBlocks	162
7.33.4.12 numSamples	162
7.33.4.13 sampleFormat	162
7.33.4.14 sampleRate	163
7.33.4.15 source	163
7.33.4.16 startTime	163
7.33.4.17 station	163
7.34 Bds::DataCollate Class Reference	163
7.34.1 Detailed Description	164
7.34.2 Constructor & Destructor Documentation	164
7.34.2.1 DataCollate()	164
7.34.2.2 ~DataCollate()	164
7.34.3 Member Function Documentation	164
7.34.3.1 addSource()	164
7.34.3.2 readData()	164
7.35 Bds::DataError Class Reference	165
7.35.1 Detailed Description	166
7.35.2 Constructor & Destructor Documentation	166
7.35.2.1 DataError() [1/2]	166
7.35.2.2 DataError() [2/2]	166
7.35.3 Member Function Documentation	166



7.35.3.1	<a href="#">getErrorNumber()</a>	166
7.35.3.2	<a href="#">getString()</a>	167
7.35.3.3	<a href="#">getTitle()</a>	167
7.35.3.4	<a href="#">mergeDataInfo()</a>	167
7.35.3.5	<a href="#">num()</a>	167
7.35.3.6	<a href="#">operator int()</a>	167
7.35.3.7	<a href="#">set()</a>	167
7.35.3.8	<a href="#">setString()</a>	168
7.35.3.9	<a href="#">setStringUser()</a>	168
7.35.3.10	<a href="#">str()</a>	168
7.35.4	<a href="#">Member Data Documentation</a>	168
7.35.4.1	<a href="#">ochannel</a>	168
7.35.4.2	<a href="#">odescription</a>	168
7.35.4.3	<a href="#">oendTime</a>	169
7.35.4.4	<a href="#">oerrorNumber</a>	169
7.35.4.5	<a href="#">ofilename</a>	169
7.35.4.6	<a href="#">onetwork</a>	169
7.35.4.7	<a href="#">osource</a>	169
7.35.4.8	<a href="#">ostartTime</a>	169
7.35.4.9	<a href="#">ostation</a>	170
7.35.4.10	<a href="#">otitle</a>	170
7.35.4.11	<a href="#">ouser</a>	170
7.36	<a href="#">Bds::DataFile Class Reference</a>	170
7.36.1	<a href="#">Detailed Description</a>	173
7.36.2	<a href="#">Member Enumeration Documentation</a>	173
7.36.2.1	<a href="#">DataOrder</a>	173
7.36.2.2	<a href="#">Features</a>	173
7.36.2.3	<a href="#">ReadOptionsList</a>	173
7.36.2.4	<a href="#">WriteOptionsList</a>	174
7.36.3	<a href="#">Constructor &amp; Destructor Documentation</a>	174
7.36.3.1	<a href="#">DataFile()</a>	174
7.36.3.2	<a href="#">~DataFile()</a>	174
7.36.4	<a href="#">Member Function Documentation</a>	174
7.36.4.1	<a href="#">close()</a>	174
7.36.4.2	<a href="#">dataErrorFixup()</a>	175
7.36.4.3	<a href="#">duplicateCheck()</a>	175
7.36.4.4	<a href="#">end()</a>	175
7.36.4.5	<a href="#">fileNameProcess()</a>	175
7.36.4.6	<a href="#">flush()</a>	175
7.36.4.7	<a href="#">getDataOrder()</a>	176
7.36.4.8	<a href="#">getFeatures()</a>	176
7.36.4.9	<a href="#">getFileName()</a>	176



7.36.4.10	<a href="#">getFilePosition()</a>	176
7.36.4.11	<a href="#">getFormat()</a>	176
7.36.4.12	<a href="#">getFormats()</a>	177
7.36.4.13	<a href="#">getInfo()</a>	177
7.36.4.14	<a href="#">getMetaData()</a>	177
7.36.4.15	<a href="#">init()</a>	177
7.36.4.16	<a href="#">open()</a>	177
7.36.4.17	<a href="#">readData()</a>	178
7.36.4.18	<a href="#">seekBlock()</a>	178
7.36.4.19	<a href="#">setFormat()</a>	178
7.36.4.20	<a href="#">setInfo()</a>	178
7.36.4.21	<a href="#">start()</a>	179
7.36.4.22	<a href="#">timeCompare()</a>	179
7.36.4.23	<a href="#">writeData()</a>	179
7.36.5	<a href="#">Member Data Documentation</a>	179
7.36.5.1	<a href="#">ofile</a>	179
7.36.5.2	<a href="#">ofileName</a>	179
7.36.5.3	<a href="#">ofileNameTime</a>	180
7.36.5.4	<a href="#">offormat</a>	180
7.36.5.5	<a href="#">omode</a>	180
7.37	<a href="#">Bds::DataFileAd22 Class Reference</a>	180
7.37.1	<a href="#">Detailed Description</a>	181
7.37.2	<a href="#">Constructor &amp; Destructor Documentation</a>	181
7.37.2.1	<a href="#">DataFileAd22()</a>	181
7.37.3	<a href="#">Member Function Documentation</a>	181
7.37.3.1	<a href="#">getDataOrder()</a>	181
7.37.3.2	<a href="#">getFeatures()</a>	181
7.37.3.3	<a href="#">getFormats()</a>	182
7.37.3.4	<a href="#">getInfo()</a>	182
7.37.3.5	<a href="#">readData()</a>	182
7.38	<a href="#">Bds::DataFileAscii Class Reference</a>	182
7.38.1	<a href="#">Detailed Description</a>	183
7.38.2	<a href="#">Constructor &amp; Destructor Documentation</a>	183
7.38.2.1	<a href="#">DataFileAscii()</a>	183
7.38.3	<a href="#">Member Function Documentation</a>	183
7.38.3.1	<a href="#">end()</a>	184
7.38.3.2	<a href="#">getDataOrder()</a>	184
7.38.3.3	<a href="#">getFeatures()</a>	184
7.38.3.4	<a href="#">getFormats()</a>	184
7.38.3.5	<a href="#">open()</a>	184
7.38.3.6	<a href="#">setFormat()</a>	185
7.38.3.7	<a href="#">setInfo()</a>	185



7.38.3.8 start()	185
7.38.3.9 writeData()	185
7.39 Bds::DataFileBdrs Class Reference	186
7.39.1 Detailed Description	186
7.39.2 Constructor & Destructor Documentation	186
7.39.2.1 DataFileBdrs()	186
7.39.3 Member Function Documentation	187
7.39.3.1 getDataOrder()	187
7.39.3.2 getFeatures()	187
7.39.3.3 getFormats()	187
7.39.3.4 getInfo()	187
7.39.3.5 readData()	188
7.40 Bds::DataFileBds Class Reference	188
7.40.1 Detailed Description	189
7.40.2 Member Enumeration Documentation	189
7.40.2.1 anonymous enum	189
7.40.2.2 anonymous enum	190
7.40.2.3 PackFormat	190
7.40.3 Constructor & Destructor Documentation	190
7.40.3.1 DataFileBds()	190
7.40.3.2 ~DataFileBds()	190
7.40.4 Member Function Documentation	190
7.40.4.1 close()	190
7.40.4.2 flush()	191
7.40.4.3 getDataOrder()	191
7.40.4.4 getDiskBlockSize()	191
7.40.4.5 getFormats()	191
7.40.4.6 getInfo()	191
7.40.4.7 open()	192
7.40.4.8 readData()	192
7.40.4.9 seekBlock()	192
7.40.4.10 setDiskBlockSize()	192
7.40.4.11 setFormat()	193
7.40.4.12 setInfo()	193
7.40.4.13 writeData()	193
7.41 Bds::DataFileBkns Class Reference	193
7.41.1 Detailed Description	194
7.41.2 Constructor & Destructor Documentation	194
7.41.2.1 DataFileBkns()	194
7.41.3 Member Function Documentation	194
7.41.3.1 getFormats()	194
7.41.3.2 open()	195



7.41.3.3 setInfo()	195
7.41.3.4 writeData()	195
7.42 Bds::DataFileCd Class Reference	195
7.42.1 Detailed Description	196
7.42.2 Constructor & Destructor Documentation	196
7.42.2.1 DataFileCd()	196
7.42.3 Member Function Documentation	196
7.42.3.1 getDataOrder()	196
7.42.3.2 getFeatures()	197
7.42.3.3 getFormats()	197
7.42.3.4 getInfo()	197
7.42.3.5 readData()	197
7.43 Bds::DataFileCss Class Reference	198
7.43.1 Detailed Description	198
7.43.2 Constructor & Destructor Documentation	198
7.43.2.1 DataFileCss()	198
7.43.3 Member Function Documentation	199
7.43.3.1 getDataOrder()	199
7.43.3.2 getFeatures()	199
7.43.3.3 getFormats()	199
7.43.3.4 getInfo()	199
7.43.3.5 readData()	200
7.44 Bds::DataFileCssData Class Reference	200
7.44.1 Detailed Description	201
7.44.2 Constructor & Destructor Documentation	201
7.44.2.1 DataFileCssData()	201
7.44.2.2 ~DataFileCssData()	201
7.44.3 Member Function Documentation	201
7.44.3.1 set()	201
7.44.4 Member Data Documentation	201
7.44.4.1 calibrationFactor	202
7.44.4.2 calibrationFreq	202
7.44.4.3 chan	202
7.44.4.4 chanid	202
7.44.4.5 clip	202
7.44.4.6 commId	202
7.44.4.7 datatype	202
7.44.4.8 dirName	202
7.44.4.9 endTime	203
7.44.4.10 file	203
7.44.4.11 fileName	203
7.44.4.12 fileOffset	203



7.44.4.13 instType . . . . .	203
7.44.4.14 jdate . . . . .	203
7.44.4.15 loadDate . . . . .	203
7.44.4.16 nsamp . . . . .	203
7.44.4.17 sampleBigEndian . . . . .	204
7.44.4.18 sampleFormat . . . . .	204
7.44.4.19 sampleRate . . . . .	204
7.44.4.20 sampleSize . . . . .	204
7.44.4.21 segtype . . . . .	204
7.44.4.22 sta . . . . .	204
7.44.4.23 startTime . . . . .	204
7.44.4.24 wfid . . . . .	205
7.45 Bds::DataFileGcf Class Reference . . . . .	205
7.45.1 Detailed Description . . . . .	205
7.45.2 Constructor & Destructor Documentation . . . . .	206
7.45.2.1 DataFileGcf() . . . . .	206
7.45.3 Member Function Documentation . . . . .	206
7.45.3.1 getDataOrder() . . . . .	206
7.45.3.2 getFeatures() . . . . .	206
7.45.3.3 getFormats() . . . . .	206
7.45.3.4 getInfo() . . . . .	206
7.45.3.5 readData() . . . . .	207
7.46 Bds::DataFileIms Class Reference . . . . .	207
7.46.1 Detailed Description . . . . .	208
7.46.2 Constructor & Destructor Documentation . . . . .	208
7.46.2.1 DataFileIms() . . . . .	208
7.46.3 Member Function Documentation . . . . .	208
7.46.3.1 close() . . . . .	208
7.46.3.2 end() . . . . .	208
7.46.3.3 getDataOrder() . . . . .	209
7.46.3.4 getFeatures() . . . . .	209
7.46.3.5 getFormats() . . . . .	209
7.46.3.6 getMetaData() . . . . .	209
7.46.3.7 open() . . . . .	209
7.46.3.8 setInfo() . . . . .	210
7.46.3.9 start() . . . . .	210
7.46.3.10 writeData() . . . . .	210
7.47 Bds::DataFileInfo Class Reference . . . . .	210
7.47.1 Detailed Description . . . . .	211
7.47.2 Constructor & Destructor Documentation . . . . .	211
7.47.2.1 DataFileInfo() . . . . .	212
7.47.3 Member Function Documentation . . . . .	212



7.47.3.1 getMember()	212
7.47.3.2 getMembers()	212
7.47.3.3 getType()	212
7.47.3.4 setMember()	212
7.47.3.5 setMembers()	213
7.47.4 Member Data Documentation	213
7.47.4.1 comment	213
7.47.4.2 endTime	213
7.47.4.3 format	213
7.47.4.4 id	213
7.47.4.5 importTime	213
7.47.4.6 importUserId	214
7.47.4.7 location	214
7.47.4.8 startTime	214
7.47.4.9 state	214
7.47.4.10 url	214
7.48 Bds::DataFileLac Class Reference	215
7.48.1 Detailed Description	215
7.48.2 Constructor & Destructor Documentation	215
7.48.2.1 DataFileLac()	215
7.48.3 Member Function Documentation	216
7.48.3.1 getDataOrder()	216
7.48.3.2 getFeatures()	216
7.48.3.3 getFormats()	216
7.48.3.4 getInfo()	216
7.48.3.5 readData()	217
7.49 Bds::DataFileLog Class Reference	217
7.49.1 Detailed Description	218
7.49.2 Constructor & Destructor Documentation	218
7.49.2.1 DataFileLog()	218
7.49.3 Member Function Documentation	218
7.49.3.1 end()	218
7.49.3.2 getDataOrder()	218
7.49.3.3 getFeatures()	219
7.49.3.4 getFormats()	219
7.49.3.5 getInfo()	219
7.49.3.6 open()	219
7.49.3.7 readData()	219
7.49.3.8 setFormat()	220
7.49.3.9 setInfo()	220
7.49.3.10 start()	220
7.49.3.11 writeData()	220



7.50 Bds::DataFileOptions Class Reference	221
7.50.1 Detailed Description	221
7.50.2 Constructor & Destructor Documentation	221
7.50.2.1 DataFileOptions()	221
7.50.3 Member Function Documentation	221
7.50.3.1 operator int()	221
7.50.3.2 operator"  =()	221
7.50.4 Member Data Documentation	222
7.50.4.1 oignoreBlockList	222
7.50.4.2 ooptionList	222
7.51 Bds::DataFileResponse Class Reference	222
7.51.1 Detailed Description	223
7.51.2 Constructor & Destructor Documentation	223
7.51.2.1 DataFileResponse()	223
7.51.3 Member Function Documentation	223
7.51.3.1 getFeatures()	223
7.51.3.2 getFormats()	223
7.51.3.3 getMetaData()	223
7.51.3.4 setInfo()	224
7.52 Bds::DataFileSac Class Reference	224
7.52.1 Detailed Description	224
7.52.2 Constructor & Destructor Documentation	225
7.52.2.1 DataFileSac()	225
7.52.3 Member Function Documentation	225
7.52.3.1 getFeatures()	225
7.52.3.2 getFormats()	225
7.52.3.3 setInfo()	225
7.53 Bds::DataFileSeed Class Reference	226
7.53.1 Detailed Description	227
7.53.2 Constructor & Destructor Documentation	227
7.53.2.1 DataFileSeed()	227
7.53.2.2 ~DataFileSeed()	227
7.53.3 Member Function Documentation	227
7.53.3.1 close()	227
7.53.3.2 end()	227
7.53.3.3 getDataOrder()	228
7.53.3.4 getFeatures()	228
7.53.3.5 getFormats()	228
7.53.3.6 getInfo()	228
7.53.3.7 msrFileWrite()	228
7.53.3.8 readData()	229
7.53.3.9 setFormat()	229



7.53.3.10 setInfo()	229
7.53.3.11 start()	229
7.53.3.12 writeData()	230
7.53.4 Member Data Documentation	230
7.53.4.1 omsrErr	230
7.53.4.2 onoLock	230
7.54 Bds::DataFileTapeDigitiser Class Reference	230
7.54.1 Detailed Description	231
7.54.2 Constructor & Destructor Documentation	231
7.54.2.1 DataFileTapeDigitiser()	231
7.54.3 Member Function Documentation	231
7.54.3.1 getFormats()	231
7.54.3.2 getInfo()	231
7.54.3.3 open()	232
7.54.3.4 readData()	232
7.55 Bds::DataFileWra Class Reference	232
7.55.1 Detailed Description	233
7.55.2 Constructor & Destructor Documentation	233
7.55.2.1 DataFileWra()	233
7.55.3 Member Function Documentation	233
7.55.3.1 getDataOrder()	233
7.55.3.2 getFeatures()	233
7.55.3.3 getFormats()	234
7.55.3.4 getInfo()	234
7.55.3.5 readData()	234
7.55.3.6 setFormat()	234
7.56 Bds::DataFileWraAgso Class Reference	235
7.56.1 Detailed Description	235
7.56.2 Constructor & Destructor Documentation	235
7.56.2.1 DataFileWraAgso()	235
7.56.3 Member Function Documentation	236
7.56.3.1 getDataOrder()	236
7.56.3.2 getFeatures()	236
7.56.3.3 getFormats()	236
7.56.3.4 getInfo()	236
7.56.3.5 readData()	237
7.57 Bds::DataFormat Class Reference	237
7.57.1 Detailed Description	238
7.57.2 Constructor & Destructor Documentation	238
7.57.2.1 DataFormat()	238
7.57.3 Member Data Documentation	238
7.57.3.1 dataRead	238



7.57.3.2 dataWrite	238
7.57.3.3 description	238
7.57.3.4 extension	239
7.57.3.5 metaDataRead	239
7.57.3.6 metaDataWrite	239
7.57.3.7 names	239
7.58 Bds::DataFormats Class Reference	239
7.58.1 Detailed Description	240
7.58.2 Constructor & Destructor Documentation	240
7.58.2.1 DataFormats()	240
7.58.2.2 ~DataFormats()	240
7.58.3 Member Function Documentation	240
7.58.3.1 findFormat()	240
7.58.3.2 formatGet()	240
7.58.3.3 formatList()	241
7.59 Bds::DataHandle Class Reference	241
7.59.1 Detailed Description	241
7.59.2 Constructor & Destructor Documentation	241
7.59.2.1 DataHandle()	241
7.59.3 Member Data Documentation	241
7.59.3.1 dataFileId	242
7.59.3.2 handle	242
7.60 Bds::DataInfo Class Reference	242
7.60.1 Detailed Description	243
7.60.2 Constructor & Destructor Documentation	243
7.60.2.1 DataInfo()	243
7.60.3 Member Data Documentation	243
7.60.3.1 array	243
7.60.3.2 channels	243
7.60.3.3 description	244
7.60.3.4 endTime	244
7.60.3.5 info	244
7.60.3.6 infoExtra	244
7.60.3.7 startTime	244
7.60.3.8 synchronous	244
7.60.3.9 warnings	245
7.61 Bds::Digitiser Class Reference	245
7.61.1 Detailed Description	246
7.61.2 Constructor & Destructor Documentation	246
7.61.2.1 Digitiser()	246
7.61.3 Member Function Documentation	246
7.61.3.1 getMember()	246



7.61.3.2 getMembers()	247
7.61.3.3 getType()	247
7.61.3.4 setMember()	247
7.61.3.5 setMembers()	247
7.61.4 Member Data Documentation	247
7.61.4.1 baseSamplingFrequency	247
7.61.4.2 endTime	248
7.61.4.3 gain	248
7.61.4.4 id	248
7.61.4.5 initialSamplingFrequency	248
7.61.4.6 name	248
7.61.4.7 numberChannels	248
7.61.4.8 serialNumber	249
7.61.4.9 shared	249
7.61.4.10 startTime	249
7.61.4.11 type	249
7.62 Bds::Fap Class Reference	249
7.62.1 Detailed Description	250
7.62.2 Constructor & Destructor Documentation	250
7.62.2.1 Fap()	250
7.62.3 Member Data Documentation	250
7.62.3.1 amplitude	250
7.62.3.2 frequency	250
7.62.3.3 phase	251
7.63 Bds::Fir Class Reference	251
7.63.1 Detailed Description	251
7.63.2 Constructor & Destructor Documentation	251
7.63.2.1 Fir()	251
7.63.3 Member Data Documentation	252
7.63.3.1 a	252
7.63.3.2 b	252
7.64 Bds::FirEntry Class Reference	252
7.64.1 Detailed Description	252
7.64.2 Constructor & Destructor Documentation	253
7.64.2.1 FirEntry()	253
7.64.3 Member Data Documentation	253
7.64.3.1 coefficient	253
7.64.3.2 error	253
7.65 Bds::GcfChannel Struct Reference	253
7.65.1 Detailed Description	254
7.65.2 Member Data Documentation	254
7.65.2.1 channel	254



7.65.2.2 format	254
7.65.2.3 sampleRate	254
7.65.2.4 streamId	254
7.65.2.5 systemId	254
7.65.2.6 type	254
7.66 Bds::Group Class Reference	255
7.66.1 Detailed Description	255
7.66.2 Constructor & Destructor Documentation	255
7.66.2.1 Group()	255
7.66.3 Member Function Documentation	256
7.66.3.1 getMember()	256
7.66.3.2 getMembers()	256
7.66.3.3 getType()	256
7.66.3.4 setMember()	256
7.66.3.5 setMembers()	256
7.66.4 Member Data Documentation	257
7.66.4.1 description	257
7.66.4.2 group	257
7.66.4.3 id	257
7.67 Bds::ListRange Class Reference	257
7.67.1 Detailed Description	258
7.67.2 Constructor & Destructor Documentation	258
7.67.2.1 ListRange()	258
7.67.3 Member Function Documentation	258
7.67.3.1 getMember()	258
7.67.3.2 getMembers()	259
7.67.3.3 getType()	259
7.67.3.4 setMember()	259
7.67.3.5 setMembers()	259
7.67.4 Member Data Documentation	259
7.67.4.1 number	259
7.67.4.2 reverse	260
7.67.4.3 start	260
7.68 Bds::Location Class Reference	260
7.68.1 Detailed Description	261
7.68.2 Constructor & Destructor Documentation	261
7.68.2.1 Location()	261
7.68.3 Member Function Documentation	262
7.68.3.1 getMember()	262
7.68.3.2 getMembers()	262
7.68.3.3 getType()	262
7.68.3.4 setMember()	262



7.68.3.5 setMembers()	262
7.68.4 Member Data Documentation	263
7.68.4.1 arrayOffsetEast	263
7.68.4.2 arrayOffsetNorth	263
7.68.4.3 datum	263
7.68.4.4 elevation	263
7.68.4.5 endTime	263
7.68.4.6 id	264
7.68.4.7 latitude	264
7.68.4.8 longitude	264
7.68.4.9 network	264
7.68.4.10 startTime	264
7.68.4.11 station	264
7.69 Bds::Log Class Reference	265
7.69.1 Detailed Description	265
7.69.2 Constructor & Destructor Documentation	266
7.69.2.1 Log()	266
7.69.3 Member Function Documentation	266
7.69.3.1 getMember()	266
7.69.3.2 getMembers()	266
7.69.3.3 getType()	266
7.69.3.4 setMember()	267
7.69.3.5 setMembers()	267
7.69.4 Member Data Documentation	267
7.69.4.1 description	267
7.69.4.2 id	267
7.69.4.3 priority	267
7.69.4.4 subSystem	268
7.69.4.5 time	268
7.69.4.6 title	268
7.69.4.7 type	268
7.70 Bds::LogSelect Class Reference	268
7.70.1 Detailed Description	269
7.70.2 Constructor & Destructor Documentation	269
7.70.2.1 LogSelect()	269
7.70.3 Member Data Documentation	269
7.70.3.1 priority	269
7.70.3.2 startTime	269
7.70.3.3 subSystem	270
7.70.3.4 type	270
7.71 Bds::Network Class Reference	270
7.71.1 Detailed Description	271



7.71.2 Constructor & Destructor Documentation	271
7.71.2.1 Network()	271
7.71.3 Member Function Documentation	271
7.71.3.1 getMember()	271
7.71.3.2 getMembers()	271
7.71.3.3 getType()	271
7.71.3.4 setMember()	272
7.71.3.5 setMembers()	272
7.71.4 Member Data Documentation	272
7.71.4.1 description	272
7.71.4.2 id	272
7.71.4.3 network	272
7.71.4.4 stations	273
7.72 Bds::Note Class Reference	273
7.72.1 Detailed Description	274
7.72.2 Constructor & Destructor Documentation	274
7.72.2.1 Note()	274
7.72.3 Member Function Documentation	275
7.72.3.1 getMember()	275
7.72.3.2 getMembers()	275
7.72.3.3 getType()	275
7.72.3.4 setMember()	275
7.72.3.5 setMembers()	276
7.72.4 Member Data Documentation	276
7.72.4.1 channel	276
7.72.4.2 dataFileId	276
7.72.4.3 description	276
7.72.4.4 docFormat	276
7.72.4.5 docUrl	276
7.72.4.6 endTime	277
7.72.4.7 errorNumber	277
7.72.4.8 id	277
7.72.4.9 importFilename	277
7.72.4.10 network	277
7.72.4.11 source	277
7.72.4.12 startTime	278
7.72.4.13 station	278
7.72.4.14 timeAdded	278
7.72.4.15 title	278
7.72.4.16 type	278
7.72.4.17 user	278
7.73 Bds::Point Class Reference	279



7.73.1 Detailed Description	279
7.73.2 Constructor & Destructor Documentation	279
7.73.2.1 Point()	279
7.73.3 Member Data Documentation	279
7.73.3.1 x	279
7.73.3.2 y	280
7.74 Bds::PoleZero Class Reference	280
7.74.1 Detailed Description	280
7.74.2 Constructor & Destructor Documentation	280
7.74.2.1 PoleZero()	280
7.74.3 Member Data Documentation	281
7.74.3.1 poles	281
7.74.3.2 zeros	281
7.75 Bds::Response Class Reference	281
7.75.1 Detailed Description	282
7.75.2 Constructor & Destructor Documentation	283
7.75.2.1 Response()	283
7.75.3 Member Data Documentation	283
7.75.3.1 channel	283
7.75.3.2 decimation	283
7.75.3.3 description	284
7.75.3.4 endTime	284
7.75.3.5 faps	284
7.75.3.6 fir	284
7.75.3.7 gain	284
7.75.3.8 gainFrequency	284
7.75.3.9 id	285
7.75.3.10 measured	285
7.75.3.11 name	285
7.75.3.12 network	285
7.75.3.13 poleZeros	285
7.75.3.14 sampleRate	285
7.75.3.15 source	286
7.75.3.16 stage	286
7.75.3.17 stageType	286
7.75.3.18 startTime	286
7.75.3.19 station	286
7.75.3.20 symmetry	286
7.75.3.21 type	287
7.76 Bds::ResponseObj Class Reference	287
7.76.1 Detailed Description	287
7.76.2 Constructor & Destructor Documentation	287



7.76.2.1 ResponseObj()	287
7.76.2.2 ~ResponseObj()	288
7.76.3 Member Function Documentation	288
7.76.3.1 getString()	288
7.76.3.2 setString()	288
7.77 Bds::Selection Class Reference	288
7.77.1 Detailed Description	289
7.77.2 Constructor & Destructor Documentation	289
7.77.2.1 Selection()	289
7.77.3 Member Data Documentation	290
7.77.3.1 calibrationName	290
7.77.3.2 channelId	290
7.77.3.3 channels	290
7.77.3.4 completeSegments	290
7.77.3.5 digitiserId	290
7.77.3.6 endTime	291
7.77.3.7 id	291
7.77.3.8 range	291
7.77.3.9 sensorId	291
7.77.3.10 sensorOldId	291
7.77.3.11 startTime	291
7.78 Bds::SelectionChannel Class Reference	292
7.78.1 Detailed Description	292
7.78.2 Constructor & Destructor Documentation	292
7.78.2.1 SelectionChannel()	292
7.78.3 Member Data Documentation	292
7.78.3.1 channel	292
7.78.3.2 network	293
7.78.3.3 source	293
7.78.3.4 station	293
7.79 Bds::SelectionInfo Class Reference	293
7.79.1 Detailed Description	294
7.79.2 Constructor & Destructor Documentation	294
7.79.2.1 SelectionInfo()	294
7.79.3 Member Data Documentation	294
7.79.3.1 arrays	294
7.79.3.2 arraysAndStations	294
7.79.3.3 channels	294
7.79.3.4 endTime	295
7.79.3.5 networks	295
7.79.3.6 numDataChannels	295
7.79.3.7 sources	295



7.79.3.8	startTime	295
7.79.3.9	stations	295
7.80	Bds::Sensor Class Reference	296
7.80.1	Detailed Description	297
7.80.2	Constructor & Destructor Documentation	297
7.80.2.1	Sensor()	297
7.80.3	Member Function Documentation	297
7.80.3.1	getMember()	297
7.80.3.2	getMembers()	297
7.80.3.3	getType()	298
7.80.3.4	setMember()	298
7.80.3.5	setMembers()	298
7.80.4	Member Data Documentation	298
7.80.4.1	endTime	298
7.80.4.2	gain	298
7.80.4.3	gainUnits	298
7.80.4.4	id	299
7.80.4.5	name	299
7.80.4.6	numberChannels	299
7.80.4.7	oldId	299
7.80.4.8	serialNumber	299
7.80.4.9	shared	299
7.80.4.10	startTime	300
7.80.4.11	type	300
7.81	Bds::Source Class Reference	300
7.81.1	Detailed Description	301
7.81.2	Constructor & Destructor Documentation	301
7.81.2.1	Source()	301
7.81.3	Member Function Documentation	301
7.81.3.1	getMember()	301
7.81.3.2	getMembers()	302
7.81.3.3	getType()	302
7.81.3.4	setMember()	302
7.81.3.5	setMembers()	302
7.81.4	Member Data Documentation	302
7.81.4.1	alias	302
7.81.4.2	description	303
7.81.4.3	id	303
7.81.4.4	source	303
7.81.4.5	sourceMeta	303
7.82	Bds::SourcePriority Class Reference	303
7.82.1	Detailed Description	304



7.82.2 Constructor & Destructor Documentation	304
7.82.2.1 SourcePriority()	304
7.82.3 Member Function Documentation	304
7.82.3.1 getMember()	305
7.82.3.2 getMembers()	305
7.82.3.3 getType()	305
7.82.3.4 setMember()	305
7.82.3.5 setMembers()	305
7.82.4 Member Data Documentation	305
7.82.4.1 endTime	306
7.82.4.2 id	306
7.82.4.3 priority	306
7.82.4.4 source	306
7.82.4.5 startTime	306
7.83 Bds::Station Class Reference	306
7.83.1 Detailed Description	307
7.83.2 Constructor & Destructor Documentation	307
7.83.2.1 Station()	307
7.83.3 Member Data Documentation	307
7.83.3.1 alias	308
7.83.3.2 channels	308
7.83.3.3 description	308
7.83.3.4 id	308
7.83.3.5 name	308
7.83.3.6 type	308
7.84 Bds::TimePeriod Class Reference	309
7.84.1 Detailed Description	309
7.84.2 Constructor & Destructor Documentation	309
7.84.2.1 TimePeriod()	309
7.84.3 Member Function Documentation	310
7.84.3.1 getMember()	310
7.84.3.2 getMembers()	310
7.84.3.3 getType()	310
7.84.3.4 setMember()	310
7.84.3.5 setMembers()	310
7.84.4 Member Data Documentation	311
7.84.4.1 endTime	311
7.84.4.2 startTime	311
7.85 Bds::User Class Reference	311
7.85.1 Detailed Description	312
7.85.2 Constructor & Destructor Documentation	312
7.85.2.1 User()	312



7.85.3 Member Function Documentation	312
7.85.3.1 getMember()	313
7.85.3.2 getMembers()	313
7.85.3.3 getType()	313
7.85.3.4 setMember()	313
7.85.3.5 setMembers()	313
7.85.4 Member Data Documentation	313
7.85.4.1 address	314
7.85.4.2 email	314
7.85.4.3 enabled	314
7.85.4.4 groups	314
7.85.4.5 id	314
7.85.4.6 name	314
7.85.4.7 password	315
7.85.4.8 telephone	315
7.85.4.9 user	315
<b>8 File Documentation</b>	<b>317</b>
8.1 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp File Reference	317
8.2 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.d File Reference	317
8.3 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.h File Reference	317
8.4 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp File Reference	318
8.5 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.d File Reference	318
8.6 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h File Reference	318
8.7 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp File Reference	318
8.8 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.d File Reference	319
8.9 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h File Reference	319
8.10 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp File Reference	319
8.10.1 Macro Definition Documentation	320
8.10.1.1 DEBUG_VELATRACK	320
8.11 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.d File Reference	320
8.12 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h File Reference	320
8.13 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp File Reference	320
8.14 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.d File Reference	321
8.15 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h File Reference	321
8.16 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp File Reference	321
8.17 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.d File Reference	321
8.18 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h File Reference	321
8.19 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp File Reference	322
8.19.1 Macro Definition Documentation	322
8.19.1.1 ALLOW_TIMESTAMP_JITTER	322
8.19.1.2 dl2printf	323



8.19.1.3 dl3printf . . . . .	323
8.19.1.4 dlprintf . . . . .	323
8.19.1.5 LDEBUG . . . . .	323
8.19.1.6 LDEBUG2 . . . . .	323
8.19.1.7 LDEBUG3 . . . . .	323
8.19.1.8 TIMESTAMP_JITTER . . . . .	323
8.20 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.d File Reference . . . . .	324
8.21 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h File Reference . . . . .	324
8.22 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp File Reference . . . . .	324
8.22.1 Function Documentation . . . . .	325
8.22.1.1 clip() . . . . .	325
8.23 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.d File Reference . . . . .	325
8.24 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h File Reference . . . . .	325
8.25 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.cpp File Reference . . . . .	325
8.25.1 Macro Definition Documentation . . . . .	326
8.25.1.1 ALLOW_TIMESTAMP_JITTER . . . . .	326
8.25.1.2 dprintf . . . . .	326
8.25.1.3 htonll . . . . .	327
8.25.1.4 INCLUDE_CHANNEL_AUTH . . . . .	327
8.25.1.5 LDEBUG . . . . .	327
8.25.1.6 MULTIPLE_SEGMENT . . . . .	327
8.25.1.7 ntohll . . . . .	327
8.25.1.8 SEGMENT_GAP . . . . .	327
8.25.1.9 TIMESTAMP_JITTER . . . . .	327
8.25.2 Variable Documentation . . . . .	327
8.25.2.1 ErrorFormatNoDataFormat . . . . .	328
8.26 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.d File Reference . . . . .	328
8.27 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h File Reference . . . . .	328
8.28 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp File Reference . . . . .	328
8.29 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.d File Reference . . . . .	329
8.30 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h File Reference . . . . .	329
8.31 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp File Reference . . . . .	329
8.31.1 Macro Definition Documentation . . . . .	329
8.31.1.1 DEBUG . . . . .	329
8.31.1.2 TEST_REORDER . . . . .	330
8.32 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.d File Reference . . . . .	330
8.33 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h File Reference . . . . .	330
8.34 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp File Reference . . . . .	330
8.35 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.d File Reference . . . . .	331
8.36 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h File Reference . . . . .	331
8.37 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp File Reference . . . . .	331
8.38 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.d File Reference . . . . .	331



8.39 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h File Reference . . . . .	331
8.40 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.cpp File Reference . . . . .	332
8.41 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.d File Reference . . . . .	332
8.42 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.h File Reference . . . . .	332
8.43 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp File Reference . . . . .	333
8.43.1 Macro Definition Documentation . . . . .	333
8.43.1.1 dprintf . . . . .	333
8.43.1.2 LDEBUG . . . . .	333
8.44 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.d File Reference . . . . .	333
8.45 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h File Reference . . . . .	333
8.46 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp File Reference . . . . .	334
8.47 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.d File Reference . . . . .	334
8.48 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.h File Reference . . . . .	334
8.49 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp File Reference . . . . .	334
8.50 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.d File Reference . . . . .	335
8.51 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h File Reference . . . . .	335
8.52 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp File Reference . . . . .	335
8.53 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.d File Reference . . . . .	336
8.54 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h File Reference . . . . .	336
8.55 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp File Reference . . . . .	336
8.55.1 Function Documentation . . . . .	336
8.55.1.1 parseStringFixedFields() . . . . .	336
8.56 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.d File Reference . . . . .	337
8.57 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h File Reference . . . . .	337
8.58 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp File Reference . . . . .	337
8.59 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.d File Reference . . . . .	338
8.60 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h File Reference . . . . .	338
8.61 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp File Reference . . . . .	338
8.61.1 Macro Definition Documentation . . . . .	339
8.61.1.1 DEBUG . . . . .	339
8.61.1.2 DEBUG_BLOCKETTE . . . . .	339
8.61.1.3 DEBUG_BLOCKS . . . . .	339
8.61.1.4 dlprintf . . . . .	339
8.61.1.5 FILL_BLOCKS . . . . .	339
8.61.1.6 ROUND_TIMESTAMPS_TO_10US . . . . .	340
8.62 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.d File Reference . . . . .	340
8.63 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h File Reference . . . . .	340
8.64 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp File Reference . . . . .	340
8.65 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.d File Reference . . . . .	340
8.66 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.h File Reference . . . . .	340
8.67 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.cpp File Reference . . . . .	341
8.68 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.d File Reference . . . . .	341



8.69 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.idl File Reference . . . . .	341
8.70 /src/blacknest/bds/bds/bdsDataLib/canada_compress.d File Reference . . . . .	341
8.71 /src/blacknest/bds/bds/bdsDataLib/canada_compress.h File Reference . . . . .	341
8.71.1 Macro Definition Documentation . . . . .	342
8.71.1.1 CANCOMP_CORRUPT . . . . .	342
8.71.1.2 CANCOMP_ERR . . . . .	342
8.71.1.3 CANCOMP_EXCEED . . . . .	342
8.71.1.4 CANCOMP_NOT_20 . . . . .	342
8.71.1.5 CANCOMP_SUCCESS . . . . .	342
8.71.2 Function Documentation . . . . .	342
8.71.2.1 canada_compress() . . . . .	343
8.71.2.2 canada_uncompress() . . . . .	343
8.72 BdsC.cc File Reference . . . . .	343
8.73 BdsC.d File Reference . . . . .	343
8.74 BdsC.h File Reference . . . . .	343
8.75 BdsD.cc File Reference . . . . .	344
8.76 BdsD.d File Reference . . . . .	344
8.77 BdsD.h File Reference . . . . .	344
8.77.1 Detailed Description . . . . .	347
8.78 BdsLib.cpp File Reference . . . . .	347
8.79 BdsLib.d File Reference . . . . .	349
8.80 BdsLib.dox File Reference . . . . .	349
8.81 BdsLib.h File Reference . . . . .	349
8.81.1 Detailed Description . . . . .	351
8.82 BdsS.cc File Reference . . . . .	351
8.83 BdsS.d File Reference . . . . .	351
8.84 BdsT.cc File Reference . . . . .	351
8.85 /src/blacknest/bds/bds/doc/bdsApiOverview.dox File Reference . . . . .	351



# Chapter 1

## Main Page

### Author

Dr Terry Barnaby

### Version

2.2.0

### Date

2020-06-23

## 1.1 Introduction

This document provides detailed reference information for the BEAM BdsApi software API of the Blacknest Data System (BDS). The API provides the ability to store and access seismic sensor data and metadata as well as administer the BDS system. The API is an object orientated API implemented in 'C++' with a number of object classes. It also has bindings for other languages which include Python and PHP.

The API operates over a network type interface using an RPC type mechanism implemented by BEAM's BOAP RPC system. The BdsApi API makes use of the BEAM 'C++' class library for lower level and system independent functionality. 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 beam-lib documentation.

The BDS Python API is built on top of the standard BDS 'C++' API using the SWIG API generator. Thus all of the standard BDS C++ API documentation applies however there are some differences due to the language facility and syntax differences. The core difference is when returning data from functions. With C++ you can return data by passing references or pointers to objects. In Python this is not generally possible and so objects are returned at the left hand side of functions instead.

This is the reference documentation for the BdsApi. An overall API description and programming manual is provided separately in: [BdsDevelopment.pdf](#)



## 1.2 Overview

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 network IP Address/Socket numbers. The BOAP name server runs on the main BDS Server host. More information on the BOAP system can be found in the beam-lib documentation.

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

1. **Bds::DataAccess** BDS Data API: This provides read only access to the data and meta data. It is used by the AutoDRM email and Web systems as well as for user and general program access to the data.
2. **Bds::DataAddAccess** BDS DataAdd API: This provides 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 provides full read/write access to the data and meta data as well as administrative configuration information.

These access API's are released in that the DataAddAccess API is a subset of the AdminAccess API and the DataAccess API is a subset of the DataAddAccess API. These API access objects should be consulted to view the functionality provided by the BDS system API's.

## 1.3 C++ 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", 0, 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", 0, 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;
}

/*****
 * BdsMetaDatal.cpp  BDS API example code for a Meta Data Client
 *      T.Barnaby,  BEAM Ltd,  2009-07-01
 *****/

*
* This is a very basic example of using the BdsApi from a meta 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 bdsTest1(DataAccess& bds){
    BError          err;
    Selection        selection;
    BIter            i;
    BUInt            n;
    BList<Station>    stations;

    // 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 stations available
if(err = bds.stationGetList(selection, stations)){
    return err.set(1, BString("Error: Getting stations: ") + err.getString());
}
// This displays some of the information available
for(stations.start(i), n = 0; !stations.isEnd(i); stations.next(i), n++){
    Station& c = stations[i];

    cout << n << ": Station: " << c.name << " Type: " << c.type << "\n";
    cout << "      " << "Description: " << c.description
        << " Number of stations " << c.channels.number() << "\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 = bdsTest1(bds);

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

    return 0;
}

```

## 1.4 Python 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:

```

#!/usr/bin/python
import sys
import getopt
from bdslib import *
# Function to read some data
def bdsTest(bds):
    err = BError();
    selection = Selection();

    # Set up selection
    selection.startTime.setString("2008-01-01T00:00:00.000000");
    selection.endTime.setString("2008-01-01T00:01:00.000000");
    print("Selection: StartTime:", selection.startTime.getString(), "EndTime:",
        selection.endTime.getString());

    selection.channels.append(SelectionChannel("TT", "TSB01", "", ""));
    # bdsDumpSelection(selection);
    # Get list of all data available for the selection
    (err, dataInfo) = bds.dataSearch(selection);
    if(err):
        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 all of the data.
    s = dataInfo.channels.size();
    if(s == 0):
        return err.set(1, "No data found");

    # bdsDumpDataInfo(dataInfo);

```



```

# Open the data file for reading in IMS format
(err, dataHandle) = bds.dataOpen(dataInfo, "r", "IMS", 0);
if(err):
    return err;

# Read the formatted data
while(1):
    # print "Loop";
    (err, data) = bds.dataFormattedRead(dataHandle, 1024);
    if(err):
        return err;
    if(data.number() == 0):
        break;
    s = "".join(chr(x) for x in data);
    print(s);
    return err;
def main():
    hostName = "localhost";
    bds = DataAccess();
    # Connect to the DataAccess service
    err = bds.connectService("//" + hostName + "/bdsDataAccess");
    if(err):
        print("Error: " + str(err) + "\n");
        return 1;
    # Connect to service
    err = bds.connect("test", "beam00");
    if(err):
        print("Error: " + str(err) + "\n");
        return 1;

    (err, version, name) = bds.getVersion();
    if(err):
        print("Error: " + str(err) + "\n");
        return 1;
    print("Version:" , version, "Name:", name);
    err = bdsTest(bds);
    if(err):
        print("Error:", err.getErrorNo(), err.getString());
        return 1;

    return 0;
if __name__ == "__main__":
    main();
#!/usr/bin/python
import sys
import getopt
from bdslib import *
# Function to read some data
def bdsTest(bds):
    err = BError();
    selection = Selection();
    dataInfo = DataInfo();

    # Set up selection
    selection.startTime.setString("2008-01-01T00:00:00.000000");
    selection.endTime.setString("2008-01-01T00:01:00.000000");
    selection.channels.append(SelectionChannel("TT", "TSB01", "", ""));
    # bdsDumpSelection(selection);
    # Get list of all data available for the selection
    (err, dataInfo) = bds.dataSearch(selection);
    if(err):
        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.
    s = dataInfo.channels.size();
    if(s == 0):
        return err.set(1, "No data found");

    # bdsDumpDataInfo(dataInfo);

    (err, dataHandle) = bds.dataOpen(dataInfo, "r", "API", 0);
    if(err):
        return err;

    blockNumber = 0;
    while(1):
        # print "Loop";
        (err, data) = bds.dataGetBlock(dataHandle, 0, 1, blockNumber);
        if(err):
            return err;
        # print("DataChannels:", data.channelData.size());

        print("Data0:", data.channelData[0][0]);
        blockNumber += 1;
    return err;
def main():

```



```

hostName = "localhost";
bds = DataAccess();
# Connect to the DataAccess service
err = bds.connectService("//" + hostName + "/bdsDataAccess");
if(err):
    print("Error: " + str(err) + "\n");
    return 1;
# Connect to service
err = bds.connect("test", "beam00");
if(err):
    print("Error: " + str(err) + "\n");
    return 1;

(err, version, name) = bds.getVersion();
if(err):
    print("Error: " + str(err) + "\n");
    return 1;
print("Version:" , version, "Name:", name);
err = bdsTest(bds);
if(err):
    print("Error:", err.getErrorNo(), err.getString());
    return 1;

return 0;
if __name__ == "__main__":
    main();
#!/usr/bin/python
import sys
import getopt
from bdslib import *
# Function to read display info on Station
def bdsTest(bds):
    err = BError();
    selection = Selection();

    # Set up selection
    selection.startTime.setString("2008-01-01T00:00:00.000000");
    selection.endTime.setString("2008-01-01T00:01:00.000000");
    selection.channels.append(SelectionChannel("TT", "TSA", "", ""));
# bdsDumpSelection(selection);
# Get list of stations available
(err, stations) = bds.stationGetList(selection);
if(err):
    return err.set(1, "Error: Getting stations: " + err.getString());
# This displays some of the information available
for s in stations:
    print("Station: " + s.name + " Type: " + s.type);
    print("      " + "Description: " + s.description + " Number of station/channels " +
        str(s.channels.number()));
return err;
def main():
    hostName = "localhost";

    # Create DataAccess object to connect to BDS Server
    bds = DataAccess();
    # Connect to the DataAccess service
    err = bds.connectService("//" + hostName + "/bdsDataAccess");
    if(err):
        print("Error: " + str(err) + "\n");
        return 1;
    # Connect to service
    err = bds.connect("test", "beam00");
    if(err):
        print("Error: " + str(err) + "\n");
        return 1;

    (err, version, name) = bds.getVersion();
    if(err):
        print("Error: " + str(err) + "\n");
        return 1;
    print("Version:" , version, "Name:", name);
    err = bdsTest(bds);
    if(err):
        print("Error:", err.getErrorNo(), err.getString());
        return 1;

    return 0;
if __name__ == "__main__":
    main();

```







## Chapter 2

# Namespace Index

### 2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

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







## Chapter 3

# Hierarchical Index

### 3.1 Class Hierarchy

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

Bds::ArrayChannel . . . . .	69
BBuffer[external]	
BBufferStore[external]	
Bds::BdsDataPacket . . . . .	75
Bds::BdsDataBlock . . . . .	71
Bds::BdsDataBlockHeader . . . . .	71
Bds::BdsDataBlockPos . . . . .	73
Bds::BdsDataPacketHeader . . . . .	77
Bds::BdsDataSegment . . . . .	79
Bds::BdsDataStreamlet . . . . .	81
Bds::BdsSeedType . . . . .	82
Bds::DataFileCssData . . . . .	200
BObj[external]	
Bds::AccessGroup . . . . .	41
Bds::Calibration . . . . .	85
Bds::Change . . . . .	99
Bds::ChangeGroup . . . . .	102
Bds::Channel . . . . .	106
Bds::ChannelInstrument . . . . .	114
Bds::DataChannel . . . . .	157
Bds::DataFileInfo . . . . .	210
Bds::Digitiser . . . . .	245
Bds::Group . . . . .	255
Bds::ListRange . . . . .	257
Bds::Location . . . . .	260
Bds::Log . . . . .	265
Bds::Network . . . . .	270
Bds::Note . . . . .	273
Bds::Sensor . . . . .	296
Bds::Source . . . . .	300
Bds::SourcePriority . . . . .	303
Bds::TimePeriod . . . . .	309
Bds::User . . . . .	311
BSocket[external]	
BoapClientObject[external]	



Bds::AdminAccess	44
Bds::DataAccess	122
Bds::DataAddAccess	135
BoapClientObject[external]	
Bds::CdChannel_1v0	91
Bds::CdDataChannel	92
Bds::CdDataFormatFrame_1v0	94
Bds::CdFlag	95
Bds::CdPacketData	96
Bds::ChannelInfo	110
Bds::ChannelInfos	113
Bds::ChannelName	118
Bds::CleanOptions	120
Bds::CompressSteim1	121
Bds::DataAvail	149
Bds::DataAvailChan	151
Bds::DataBlock	153
Bds::DataBlockPos	155
Bds::DataCollate	163
Bds::DataError	165
Bds::DataFile	170
Bds::DataFileAd22	180
Bds::DataFileAscii	182
Bds::DataFileBdrs	186
Bds::DataFileBds	188
Bds::DataFileBknas	193
Bds::DataFileCd	195
Bds::DataFileCss	198
Bds::DataFileGcf	205
Bds::DataFileIms	207
Bds::DataFileLac	215
Bds::DataFileLog	217
Bds::DataFileResponse	222
Bds::DataFileSac	224
Bds::DataFileSeed	226
Bds::DataFileTapeDigitiser	230
Bds::DataFileWra	232
Bds::DataFileWraAgso	235
Bds::DataFileOptions	221
Bds::DataFormat	237
Bds::DataFormats	239
Bds::DataHandle	241
Bds::DataInfo	242
Bds::Fap	249
Bds::Fir	251
Bds::FirEntry	252
Bds::GcfChannel	253
Bds::LogSelect	268
Bds::Point	279
Bds::PoleZero	280
Bds::Response	281
Bds::ResponseObj	287
Bds::Selection	288
Bds::SelectionChannel	292
Bds::SelectionInfo	293
Bds::Station	306



## Chapter 4

# Class Index

### 4.1 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 . . . . .	41
<a href="#">Bds::AdminAccess</a>	
This is the <a href="#">AdminAccess</a> Access API interface . . . . .	44
<a href="#">Bds::ArrayChannel</a>	
This class defines an arrays channel . . . . .	69
<a href="#">Bds::BdsDataBlock</a>	
BdsDataFileBds: internal fixed size BDS Data Block . . . . .	71
<a href="#">Bds::BdsDataBlockHeader</a>	
BdsDataFileBds: internal fixed size BDS Data Block header . . . . .	71
<a href="#">Bds::BdsDataBlockPos</a>	
BdsDataFileBds: internal file storage data block position . . . . .	73
<a href="#">Bds::BdsDataPacket</a>	
BdsDataFileBds: internal file storage packet . . . . .	75
<a href="#">Bds::BdsDataPacketHeader</a>	
BdsDataFileBds internal file storage packet header . . . . .	77
<a href="#">Bds::BdsDataSegment</a>	
BdsDataFileBds: internal file storage data segment . . . . .	79
<a href="#">Bds::BdsDataStreamlet</a>	
BdsDataFileBds: internal file storage data streamlet . . . . .	81
<a href="#">Bds::BdsSeedType</a>	
BdsDataFileSeed internal parent for all SEED types . . . . .	82
<a href="#">Bds::Calibration</a>	
This class defines a calibration setting . . . . .	85
<a href="#">Bds::CdChannel_1v0</a>	
BdsDataFile: Internal CD1.0 channel information . . . . .	91
<a href="#">Bds::CdDataChannel</a>	
BdsDataFile: Internal CD channel information . . . . .	92
<a href="#">Bds::CdDataFormatFrame_1v0</a>	
BdsDataFile: Internal CD1.0 frame information . . . . .	94
<a href="#">Bds::CdFlag</a>	
BdsDataFile: Internal CD flag . . . . .	95
<a href="#">Bds::CdPacketData</a>	
BdsDataFile: Internal CD data packet . . . . .	96
<a href="#">Bds::Change</a>	
This holds information on a medatdata or sensor data change . . . . .	99



<a href="#">Bds::ChangeGroup</a>	
This holds information on a set of Changes	102
<a href="#">Bds::Channel</a>	
This class defines a seismic data <a href="#">Channel</a>	106
<a href="#">Bds::ChannelInfo</a>	
This class provides information on a channel	110
<a href="#">Bds::ChannelInfos</a>	
This class provides metadata information on a set of channels	113
<a href="#">Bds::ChannelInstrument</a>	
This class defines a <a href="#">Channel</a> 's instrument	114
<a href="#">Bds::ChannelName</a>	
This class defines a full channel name	118
<a href="#">Bds::CleanOptions</a>	
This defines the set of clean options used in the clean() function	120
<a href="#">Bds::CompressSteim1</a>	
Steim1 un-compress class	121
<a href="#">Bds::DataAccess</a>	
This is the Data Access API interface	122
<a href="#">Bds::DataAddAccess</a>	
This is the DataAdd Access API interface	135
<a href="#">Bds::DataAvail</a>	
This class provides availability information on a particular period of data	149
<a href="#">Bds::DataAvailChan</a>	
This class defines availability information on a set of data	151
<a href="#">Bds::DataBlock</a>	
This class provides the actual seismic data values contained within a single data block	153
<a href="#">Bds::DataBlockPos</a>	
This defines the position of a data block in a file. It is used by the BDS data converters to order blocks by time	155
<a href="#">Bds::DataChannel</a>	
This class defines information on a single channels set of data stored in a file	157
<a href="#">Bds::DataCollate</a>	
Not sure if this is used or what it does	163
<a href="#">Bds::DataError</a>	
This stores a data error. It includes an error number and a string as well as information on what seismic channel it is for	165
<a href="#">Bds::DataFile</a>	
This class defines the interface for generic data file access that all of the BDS data converters share	170
<a href="#">Bds::DataFileAd22</a>	
Data file converter for AD22 format files	180
<a href="#">Bds::DataFileAscii</a>	
Data file converter for ASCII format files	182
<a href="#">Bds::DataFileBdrs</a>	
Data file converter for BDRS format files	186
<a href="#">Bds::DataFileBds</a>	
This class implements the BDS Data File/Stream access system	188
<a href="#">Bds::DataFileBknas</a>	
Data file converter for BKNAS format files	193
<a href="#">Bds::DataFileCd</a>	
Data file converter for CD1.0 and CD1.1 file formats	195
<a href="#">Bds::DataFileCss</a>	
Data file converter for CSS format files	198
<a href="#">Bds::DataFileCssData</a>	
<a href="#">DataFileCss</a> internal CSS data type	200
<a href="#">Bds::DataFileGcf</a>	
Data file converter for GCF format files	205



<a href="#">Bds::DataFileIms</a>	
Data file convertor for IMS format files . . . . .	207
<a href="#">Bds::DataFileInfo</a>	
This class defines information on a sensor data file . . . . .	210
<a href="#">Bds::DataFileLac</a>	
Data file convertor for LAC format files . . . . .	215
<a href="#">Bds::DataFileLog</a>	
Data file convertor for LOG format files . . . . .	217
<a href="#">Bds::DataFileOptions</a>	
This defines a list of BDS data convtor options . . . . .	221
<a href="#">Bds::DataFileResponse</a>	
This class defines the interface for generic response data file access . . . . .	222
<a href="#">Bds::DataFileSac</a>	
Data file convertor for SAC format files . . . . .	224
<a href="#">Bds::DataFileSeed</a>	
Data file convertor for SEED file formats . . . . .	226
<a href="#">Bds::DataFileTapeDigitiser</a>	
This class implements the TapeDigitiser's file output conversion and storing system . . . . .	230
<a href="#">Bds::DataFileWra</a>	
Data file convertor for WRA format files . . . . .	232
<a href="#">Bds::DataFileWraAgso</a>	
Data file convertor for WRA AGSO format files . . . . .	235
<a href="#">Bds::DataFormat</a>	
This holds information on a seismic data format . . . . .	237
<a href="#">Bds::DataFormats</a>	
This class defines the interface for generic data file access . . . . .	239
<a href="#">Bds::DataHandle</a>	
This defines a handle to a sensor data stream/file when opened for read or write . . . . .	241
<a href="#">Bds::DataInfo</a>	
This class defines information on a set of data . . . . .	242
<a href="#">Bds::Digitiser</a>	
This class defines a seismic <a href="#">Digitiser</a> . . . . .	245
<a href="#">Bds::Fap</a>	
This class defines an entry in an Amplitude/Phase <a href="#">Response</a> table . . . . .	249
<a href="#">Bds::Fir</a>	
This class defines an FIR response table . . . . .	251
<a href="#">Bds::FirEntry</a>	
This class defines an entry in a FIR coefficient table . . . . .	252
<a href="#">Bds::GcfChannel</a>	
<a href="#">DataFileGcf</a> internal GCF channel information . . . . .	253
<a href="#">Bds::Group</a>	
This holds information on a user security group . . . . .	255
<a href="#">Bds::ListRange</a>	
This class defines an integer based range . . . . .	257
<a href="#">Bds::Location</a>	
This class defines the physical location of a <a href="#">Station</a> . . . . .	260
<a href="#">Bds::Log</a>	
This holds information on a <a href="#">Log</a> entry . . . . .	265
<a href="#">Bds::LogSelect</a>	
This defines the selection cirteria when requesting a set of log entries . . . . .	268
<a href="#">Bds::Network</a>	
This class defines a seismic <a href="#">Network</a> organisation . . . . .	270
<a href="#">Bds::Note</a>	
This holds information on a <a href="#">Note</a> for general information . . . . .	273
<a href="#">Bds::Point</a>	
This class defines an X,Y location . . . . .	279
<a href="#">Bds::PoleZero</a>	
This class defines a Pole/Zero <a href="#">Response</a> . . . . .	280



<a href="#">Bds::Response</a>	This class defines a seismic <a href="#">Response</a> characteristic . . . . .	281
<a href="#">Bds::ResponseObj</a>	<a href="#">Response</a> object adding string conversion . . . . .	287
<a href="#">Bds::Selection</a>	This class defines a generic metadata or seismic data selection . . . . .	288
<a href="#">Bds::SelectionChannel</a>	This class defines a channel for selection . . . . .	292
<a href="#">Bds::SelectionInfo</a>	This class defines the set of metadata or seismic data selected when <code>getSelectionInfo()</code> is use . . . . .	293
<a href="#">Bds::Sensor</a>	This class defines a seismic <a href="#">Sensor</a> . . . . .	296
<a href="#">Bds::Source</a>	This class defines a seismic data <a href="#">Source</a> . . . . .	300
<a href="#">Bds::SourcePriority</a>	This class defines a <a href="#">Source</a> Priority entry . . . . .	303
<a href="#">Bds::Station</a>	This class defines a seismic station . . . . .	306
<a href="#">Bds::TimePeriod</a>	This class defines a <a href="#">TimePeriod</a> . . . . .	309
<a href="#">Bds::User</a>	This holds information on a user . . . . .	311



## Chapter 5

# File Index

### 5.1 File List

Here is a list of all files with brief descriptions:

/src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp	317
/src/blacknest/bds/bds/bdsDataLib/BdsCompress.d	317
/src/blacknest/bds/bds/bdsDataLib/BdsCompress.h	317
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp	318
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.d	318
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h	318
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp	318
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.d	319
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h	319
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp	319
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.d	320
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h	320
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp	320
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.d	321
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h	321
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp	321
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.d	321
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h	321
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp	322
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.d	324
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h	324
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp	324
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.d	325
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h	325
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.cpp	325
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.d	328
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h	328
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp	328
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.d	329
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h	329
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp	329
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.d	330
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h	330
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp	330
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.d	331



/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h	331
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp	331
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.d	331
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h	331
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.cpp	332
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.d	332
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.h	332
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp	333
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.d	333
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h	333
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp	334
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.d	334
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.h	334
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp	334
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.d	335
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h	335
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp	335
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.d	336
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h	336
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp	336
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.d	337
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h	337
/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp	337
/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.d	338
/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h	338
/src/blacknest/bds/bds/bdsDataLib/canada_compress.d	341
/src/blacknest/bds/bds/bdsDataLib/canada_compress.h	341
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp	338
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.d	340
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h	340
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp	340
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.d	340
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.h	340
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.cpp	341
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.d	341
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.idl	341
BdsC.cc	343
BdsC.d	343
BdsC.h	343
BdsD.cc	344
BdsD.d	344
BdsD.h	
BOAP data class definitions for: Bds	344
BdsLib.cpp	347
BdsLib.d	349
BdsLib.h	
General BdsLib API functions	349
BdsS.cc	351
BdsS.d	351
BdsT.cc	351



## Chapter 6

# Namespace Documentation

### 6.1 Bds Namespace Reference

#### Classes

- class [AccessGroup](#)  
*This holds information on data access groups.*
- class [AdminAccess](#)  
*This is the [AdminAccess](#) Access API interface.*
- class [ArrayChannel](#)  
*This class defines an arrays channel.*
- struct [BdsDataBlock](#)  
*BdsDataFileBds: internal fixed size BDS Data Block.*
- struct [BdsDataBlockHeader](#)  
*BdsDataFileBds: internal fixed size BDS Data Block header.*
- class [BdsDataBlockPos](#)  
*BdsDataFileBds: internal file storage data block position.*
- class [BdsDataPacket](#)  
*BdsDataFileBds: internal file storage packet.*
- struct [BdsDataPacketHeader](#)  
*BdsDataFileBds internal file storage packet header.*
- class [BdsDataSegment](#)  
*BdsDataFileBds: internal file storage data segment.*
- class [BdsDataStreamlet](#)  
*BdsDataFileBds: internal file storage data streamlet.*
- class [BdsSeedType](#)  
*BdsDataFileSeed internal parent for all SEED types.*
- class [Calibration](#)  
*This class defines a calibration setting.*
- struct [CdChannel\\_1v0](#)  
*BdsDataFile: Internal CD1.0 channel information.*
- class [CdDataChannel](#)  
*BdsDataFile: Internal CD channel information.*
- struct [CdDataFormatFrame\\_1v0](#)  
*BdsDataFile: Internal CD1.0 frame information.*
- class [CdFlag](#)



- BdsDataFile: Internal CD flag.*

  - class [CdPacketData](#)

*BdsDataFile: Internal CD data packet.*
  - class [Change](#)

*This holds information on a medatdata or sensor data change.*
  - class [ChangeGroup](#)

*This holds information on a set of Changes.*
  - class [Channel](#)

*This class defines a seismic data [Channel](#).*
  - class [ChannelInfo](#)

*This class provides information on a channel.*
  - class [ChannelInfos](#)

*This class provides metadata information on a set of channels.*
  - class [ChannelInstrument](#)

*This class defines a [Channel](#)'s instrument.*
  - class [ChannelName](#)

*This class defines a full channel name.*
  - class [CleanOptions](#)

*This defines the set of clean options used in the `clean()` function.*
  - class [CompressSteim1](#)

*Steim1 un-compress class.*
  - class [DataAccess](#)

*This is the Data Access API interface.*
  - class [DataAddAccess](#)

*This is the DataAdd Access API interface.*
  - class [DataAvail](#)

*This class provides availability information on a particular period of data.*
  - class [DataAvailChan](#)

*This class defines availablity information on a set of data.*
  - class [DataBlock](#)

*This class provides the acual seismic data values contained within a single data block.*
  - class [DataBlockPos](#)

*This defines the position of a data block in a file. It is used by the BDS data convertors to order blocks by time.*
  - class [DataChannel](#)

*This class defines information on a single channels set of data stored in a file.*
  - class [DataCollate](#)

*Not sure if this is used or what it does.*
  - class [DataError](#)

*This stores a data error. It includes and error number and a string as well as information on what seismic channel it is for.*
  - class [DataFile](#)

*This class defines the interface for generic data file access that all of the BDS data conterors share.*
  - class [DataFileAd22](#)

*Data file convertor for AD22 format files.*
  - class [DataFileAscii](#)

*Data file convertor for ASCII format files.*
  - class [DataFileBdrs](#)

*Data file convertor for BDRS format files.*
  - class [DataFileBds](#)

*This class implements the BDS Data File/Stream access system.*
  - class [DataFileBknas](#)



- Data file convertor for BKNAS format files.*
- class [DataFileCd](#)
  - Data file convertor for CD1.0 and CD1.1 file formats.*
- class [DataFileCss](#)
  - Data file convertor for CSS format files.*
- class [DataFileCssData](#)
  - [DataFileCss](#) internal CSS data type.*
- class [DataFileGcf](#)
  - Data file convertor for GCF format files.*
- class [DataFileImms](#)
  - Data file convertor for IMS format files.*
- class [DataFileInfo](#)
  - This class defines information on a sensor data file.*
- class [DataFileLac](#)
  - Data file convertor for LAC format files.*
- class [DataFileLog](#)
  - Data file convertor for LOG format files.*
- class [DataFileOptions](#)
  - This defines a list of BDS data convtor options.*
- class [DataFileResponse](#)
  - This class defines the interface for generic response data file access.*
- class [DataFileSac](#)
  - Data file convertor for SAC format files.*
- class [DataFileSeed](#)
  - Data file convertor for SEED file formats.*
- class [DataFileTapeDigitiser](#)
  - This class implements the TapeDigitiser's file output conversion and storing system.*
- class [DataFileWra](#)
  - Data file convertor for WRA format files.*
- class [DataFileWraAgso](#)
  - Data file convertor for WRA AGSO format files.*
- class [DataFormat](#)
  - This holds information on a seismic data format.*
- class [DataFormats](#)
  - This class defines the interface for generic data file access.*
- class [DataHandle](#)
  - This defines a handle to a sensor data stream/file when opened for read or write.*
- class [DataInfo](#)
  - This class defines information on a set of data.*
- class [Digitiser](#)
  - This class defines a seismic [Digitiser](#).*
- class [Fap](#)
  - This class defines an entry in an Amplitude/Phase [Response](#) table.*
- class [Fir](#)
  - This class defines an FIR response table.*
- class [FirEntry](#)
  - This class defines an entry in a FIR coefficient table.*
- struct [GcfChannel](#)
  - [DataFileGcf](#) internal GCF channel information.*
- class [Group](#)
  - This holds information on a user security group.*



- class [ListRange](#)  
*This class defines an integer based range.*
- class [Location](#)  
*This class defines the physical location of a [Station](#).*
- class [Log](#)  
*This holds information on a [Log](#) entry.*
- class [LogSelect](#)  
*This defines the selection criteria when requesting a set of log entries.*
- class [Network](#)  
*This class defines a seismic [Network](#) organisation.*
- class [Note](#)  
*This holds information on a [Note](#) for general information.*
- class [Point](#)  
*This class defines an X,Y location.*
- class [PoleZero](#)  
*This class defines a Pole/Zero [Response](#).*
- class [Response](#)  
*This class defines a seismic [Response](#) characteristic.*
- class [ResponseObj](#)  
*[Response](#) object adding string conversion.*
- class [Selection](#)  
*This class defines a generic metadata or seismic data selection.*
- class [SelectionChannel](#)  
*This class defines a channel for selection.*
- class [SelectionInfo](#)  
*This class defines the set of metadata or seismic data selected when `getSelectionInfo()` is use.*
- class [Sensor](#)  
*This class defines a seismic [Sensor](#).*
- class [Source](#)  
*This class defines a seismic data [Source](#).*
- class [SourcePriority](#)  
*This class defines a [Source](#) Priority entry.*
- class [Station](#)  
*This class defines a seismic station.*
- class [TimePeriod](#)  
*This class defines a [TimePeriod](#).*
- class [User](#)  
*This holds information on a user.*

## Enumerations

- enum [Errors](#) {  
[ErrorNoMetaData](#) = 64, [ErrorDataQuality](#) = 65, [ErrorSlaveMode](#) = 66, [ErrorTimeStamp](#) = 67,  
[ErrorValidate](#) = 80, [ErrorValidateMissingBlocks](#) = 81, [ErrorValidateTimeBackwards](#) = 82, [ErrorValidateFilenameTime](#)  
= 83,  
[ErrorValidateMetaData](#) = 84, [ErrorValidateFix](#) = 85, [ErrorValidateDuplicate](#) = 86, [ErrorValidateReorder](#) = 87,  
[ErrorValidateBdsFudge](#) = 88 }  
*The System Error number list in addition to standard system error numbers.*
- enum [Priority](#) { [PriorityLow](#), [PriorityNormal](#), [PriorityHigh](#) }  
*Priority levels.*
- enum [Mode](#) { [ModeMaster](#), [ModeSlave](#) }



- BdsServer mode.*
- enum [DataFlags](#) {  
[DataFlagNone](#) = 0x00, [DataFlagClipDataToTime](#) = 0x01, [DataFlagClipDataToChannels](#) = 0x02,  
[DataFlagMergeSegments](#) = 0x04,  
[DataFlagNoMetadata](#) = 0x08 }
- Flags when opening data files.*
- enum [SelectionGroup](#) { [SelectionGroupData](#), [SelectionGroupMetaData](#), [SelectionGroupDataWithCount](#) }
- The Selection group when making selections.*
- enum [SampleFormat](#) {  
[SampleFormatUnknown](#), [SampleFormatInt16](#), [SampleFormatInt32](#), [SampleFormatFloat32](#),  
[SampleFormatFloat64](#), [SampleFormatInt24](#) }
- The actual format of a data sample.*
- enum [AvailType](#) { [AvailNone](#), [AvailPartial](#), [AvailFull](#) }
- A flag defining the data availability state.*
- enum [BdsDataType](#) { [BdsDataTypeBlock](#) = 0x42534442, [BdsDataTypeInfo](#) = 0x30534442, [BdsDataTypeData](#) = 0x31534442, [BdsDataTypeInfoExtra](#) = 0x32534442 }
- BdsDataFileBds: internal file block type field.*
- enum [FileHeaderType](#) { [FileHeaderType\\_Standard](#) = 1, [FileHeaderType\\_TapeDigitiser](#) = 10 }
- enum [FileSampleType](#) {  
[FileSampleType\\_Unknown](#), [FileSampleType\\_Float32](#), [FileSampleType\\_Float64](#), [FileSampleType\\_Int16](#),  
[FileSampleType\\_Int32](#) }

## Functions

- void [bdsDumpPoleZeros](#) ([PoleZero](#) poleZeros)  
*Debug print out a PoleZeros object.*
- void [bdsChannelGetTypeAux](#) ( [BString](#) name, [BString](#) &type, [BString](#) &aux)  
*Get the channel type and aux fields from a generic channel name.*
- [BString](#) [bdsChannelGetName](#) ( [BString](#) type, [BString](#) aux)  
*Create a full channel name from a channels type and aux fields.*
- [BError](#) [bdsDataInfoSetTimeRange](#) ([DataInfo](#) &dataInfo)  
*Restricts the time range of all of the [DataInfo](#)'s channels to match the [DataInfo](#)'s startTime/endTime fields.*
- [BError](#) [bdsDataInfoFromInfo](#) ( [BDictString](#) info, [DataInfo](#) &dataInfo, [Bool](#) append)  
*Convert info to [DataInfo](#).*
- [BError](#) [bdsInfoFromDataInfo](#) (const [DataInfo](#) &dataInfo, [BDictString](#) &info)  
*Converts a [DataInfo](#) object into a [BDictString](#) list of named strings.*
- [BError](#) [bdsDataInfoMergeFlatten](#) ([DataInfo](#) &dataInfo, const [DataInfo](#) &dataInfoAdd)  
*Merges a [DataInfo](#) into another flattening the segments to 1 for use in dataOpen() calls.*
- [BString](#) [bdsStationAlias](#) ([Station](#) station)  
*Returns the station alias if set else its name.*
- void [bdsDumpSelection](#) ([Selection](#) sel)  
*Debug print out a [Selection](#) object.*
- void [bdsDumpDataInfo](#) ([DataInfo](#) dataInfo, int includeInfo=0)  
*Debug print out a [DataInfo](#) object.*
- void [bdsDumpChannelInfos](#) (const [ChannelInfos](#) &channelInfos)  
*Debug print out a [ChannelInfos](#) object.*
- void [bdsDumpData](#) (const [DataBlock](#) &dataBlock, int nSamples=0)  
*Debug print out a [DataBlock](#) object.*
- [BString](#) [bdsDataChannelInfo](#) (const [DataChannel](#) &dataChannel)  
*Returns a string representation of a [DataChannel](#) object.*
- [BString](#) [bdsDataChannelRef](#) (const [DataChannel](#) &dataChannel)



- Returns the string reference name of a [DataChannel](#) object.*

  - **BString** [bdsDataChannelRef](#) (const [ChannelInfo](#) &channelInfo)

*Returns the string reference name of a [ChannelInfo](#) object.*

  - **BError** [bdsDataChannelOverallResponse](#) (const [ChannelInfo](#) &channelInfo, [Response](#) &response)

*Returns the overal response from the list of responses in a [ChannelInfo](#).*

  - **BString** [bdsSelectionChannelInfo](#) (const [Selection](#) &selection, **BUInt** channel)

*Returns a string describing the name and time period of a selection channel.*

  - double [bdsPoleZeroGain](#) (const [PoleZero](#) &poleZero, double frequency)

*Calculates the overal gain of the given [PoleZero](#) transfer function.*

  - void [bdsPoleZeroGainPhase](#) (const [PoleZero](#) &poleZero, double frequency, double &gain, double &phase)

*Calculates the overal gain and phase of the given [PoleZero](#) transfer function.*

  - void [bdsPoleZeroToFap](#) (const [PoleZero](#) &poleZero, **BUInt** nPoints, double calibrationFrequency, double sampleFrequency, **BArray**< [Fap](#) > &fap)

*Convert [PoleZero](#) to FAP.*

  - static **BString** [fileNameTime](#) ( **BTimeStamp** t)
  - **BString** [bdsFileNameExpand](#) ( **BString** fileName, [ChannelInfo](#) &channelInfo)

*Default filename from a [ChannelInfo](#).*

  - **BString** [bdsFileNameExpand](#) ( **BString** fileName, [ChannelInfos](#) &channelInfos)

*Default filename from a list of [ChannelInfo](#)'s.*

  - **BError** [bdsUnCompressCm8](#) ( **BUInt8** \*buffer, **BUInt** n, **BArray**< **BInt32** > & data)

*Uncompress CM8 formatted data.*

  - **BError** [bdsUnCompressSteim1](#) ( **BUInt8** \*buffer, **BUInt** n, **BArray**< **BInt32** > & data)

*Uncompress STEIM1 formatted data.*

  - static **BString** [nullString](#) ( **BString** s)
  - **BUInt32** [crc](#) ( **BUInt32** crc, void \* data, int numBytes)
  - static void [crclnit](#) ()
  - static uint64\_t [crc64](#) (const void \*buffer, const uint32\_t len)
  - **BString** [getHexString](#) (char \* data, int len)
  - int [duplicateDump](#) ([DataBlock](#) &data1, [DataBlock](#) &data2, int channel)
  - static **BError** [fixedString](#) (double v, int fieldWidth, int numDecimal, **BString** &str)
  - void [dataCalculateDifference](#) ( **BInt32** &prevValue, **BArray**< **BInt32** > & data)
  - void [dataCalculateUnDifference](#) ( **BInt32** &prevValue, **BArray**< **BInt32** > & data)
  - **BInt32** [dataChecksum](#) ( **BInt32** checksum, **BArray**< **BInt32** > & data)
  - **BError** [dataCompressCm6](#) (int &prevValue1, int &prevValue2, **BArray**< **BInt32** > & data, **BString** &d)
  - **BError** [dataDeCompressCm6](#) (int &prevValue1, int &prevValue2, **BString** &d, **BArray**< **BInt32** > & data)
  - static void [dataConvert](#) (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BInt32** > &dataOut)
  - static **BString** [stringFormat](#) ( **BTimeStamp** t)
  - static **BString** [removeCR](#) ( **BString** str)
  - **BString** [fixedWidthValue](#) (double v, int width)

*This returns a double as a fixed width string truncating the data.*

  - static void [bdsDataFileSeedLogWarning](#) (char \*str)
  - static void [bdsDataFileSeedLogError](#) (char \*str)
  - static hptime\_t [seedTime](#) ( **BTimeStamp** t)
  - static **BString** [seedTimeString](#) ( **BTimeStamp** t)
  - static void [dataConvert](#) (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BInt32** > &dataOut)
  - static void [dataConvert](#) (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BFloat32** > &dataOut)
  - static void [record\\_handler](#) (char \*record, int reclen, void \*info)



## Variables

- const **BUInt32** `apiVersion` = 0
- const int `NetworkNameLen` = 3  
*Maximum `Network` name length.*
- const int `StationNameLen` = 5  
*Maximum `Station` name length.*
- const int `ChannelTypeLen` = 3  
*Maximum `Channel` type name length.*
- const int `ChannelAuxLen` = 2  
*Maximum `Channel` Aux length.*
- const int `SourceLen` = 16  
*Maximum `Source` length.*
- const **BString** `BdsDataFileVersion` = "1.2.0"
- static uint64\_t `crcVec` [256]
- static int `crclInitDone`
- static char `cm6Table` [64]
- static **BUInt8** `cm6TableRev` [128]
- const double `Scale` = 16777216.0
- `DataFormats` `dataFormats`

### 6.1.1 Enumeration Type Documentation

#### 6.1.1.1 AvailType

enum `Bds::AvailType`

A flag defining the data availability state.

##### Enumerator

<code>AvailNone</code>	There is no data available There is full data available
<code>AvailPartial</code>	There is partial data available
<code>AvailFull</code>	There is no data available

#### 6.1.1.2 BdsDataType

enum `Bds::BdsDataType`

`BdsDataFileBds`: internal file block type field.

##### Enumerator

<code>BdsDataTypeBlock</code>	
-------------------------------	--



**Enumerator**

BdsDataTypeInfo	
BdsDataTypeData	
BdsDataTypeInfoExtra	

**6.1.1.3 DataFlags**

```
enum Bds::DataFlags
```

Flags when opening data files.

**Enumerator**

DataFlagNone	No data flags
DataFlagClipDataToTime	Clip the data to the time period requested so that data begins and ends with the sample at the requested time. Normally the BDS will return data beginning at the startTime of the data block in which the user startTime occurred and the endTime of the block that the user supplied endTime occurs so that complete original data blocks are returned.
DataFlagClipDataToChannels	When requesting data from a number of channels the start and end times per channel may be different due to missing blocks or other reasons. This option asks the BDS to truncate the data so that all channels start and end with the sample timed sample.
DataFlagMergeSegments	Data will normally be segmented at file boundaries. This option merges these segments assuming the start/end times match.
DataFlagNoMetadata	Don't include Metadata in export data files.

**6.1.1.4 Errors**

```
enum Bds::Errors
```

The System Error number list in addition to standard system error numbers.

**Enumerator**

ErrorNoMetaData	No Metadata is available
ErrorDataQuality	Data quality error
ErrorSlaveMode	BdsServer is in slave mode
ErrorTimeStamp	Timestamp invalid
ErrorValidate	A validation error occurred
ErrorValidateMissingBlocks	Validation found missing blocks
ErrorValidateTimeBackwards	Validation found the time went backwards between blocks
ErrorValidateFilenameTime	Validation of the file name failed
ErrorValidateMetaData	There was no Metadata available



## Enumerator

ErrorValidateFix	Validation has fixed some issues
ErrorValidateDuplicate	Validation has found duplicate blocks
ErrorValidateReorder	Validation has re-orderd blocks
ErrorValidateBdsFudge	Special BDS SensorData/Metadata changes have been applied

## 6.1.1.5 FileHeaderType

```
enum Bds::FileHeaderType
```

## Enumerator

FileHeaderType_Standard	
FileHeaderType_TapeDigitiser	

## 6.1.1.6 FileSampleType

```
enum Bds::FileSampleType
```

## Enumerator

FileSampleType_Unknown	
FileSampleType_Float32	
FileSampleType_Float64	
FileSampleType_Int16	
FileSampleType_Int32	

## 6.1.1.7 Mode

```
enum Bds::Mode
```

BdsServer mode.

## Enumerator

ModeMaster	BdsServer is a master
ModeSlave	BdsServer is a slave



#### 6.1.1.8 Priority

enum [Bds::Priority](#)

Priority levels.

##### Enumerator

PriorityLow	The lowest priority level
PriorityNormal	The normal priority level
PriorityHigh	The highest priority level

#### 6.1.1.9 SampleFormat

enum [Bds::SampleFormat](#)

The actual format of a data sample.

##### Enumerator

SampleFormatUnknown	Unknown sample format
SampleFormatInt16	16 bit signed integer format
SampleFormatInt32	32 bit signed integer format
SampleFormatFloat32	IEEE 32 bit floating point format
SampleFormatFloat64	IEEE 64 bit floating point format
SampleFormatInt24	24 bit signed integer format

#### 6.1.1.10 SelectionGroup

enum [Bds::SelectionGroup](#)

The [Selection](#) group when making selections.

##### Enumerator

SelectionGroupData	Select items from <a href="#">Sensor</a> data
SelectionGroupMetaData	Select items from Metadata
SelectionGroupDataWithCount	Select items from <a href="#">Sensor</a> data and return the number of items found

### 6.1.2 Function Documentation



### 6.1.2.1 bdsChannelGetName()

```
BString Bds::bdsChannelGetName (
    BString type,
    BString aux )
```

Create a full channel name from a channels type and aux fields.

### 6.1.2.2 bdsChannelGetTypeAux()

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

Get the channel type and aux fields from a generic channel name.

#### Parameters

in	<i>name</i>	The channels full name
out	<i>type</i>	Returns the type component of the channel's name
out	<i>aux</i>	Returns the aux component of the channel's name

### 6.1.2.3 bdsDataChannelInfo()

```
BString Bds::bdsDataChannelInfo (
    const DataChannel & dataChannel )
```

Returns a string representation of a [DataChannel](#) object.

### 6.1.2.4 bdsDataChannelOverallResponse()

```
BError Bds::bdsDataChannelOverallResponse (
    const ChannelInfo & channelInfo,
    Response & response )
```

Returns the overal response from the list of responses in a [ChannelInfo](#).



6.1.2.5 `bdsDataChannelRef()` [1/2]

```
BString Bds::bdsDataChannelRef (
    const DataChannel & dataChannel )
```

Returns the string reference name of a [DataChannel](#) object.

6.1.2.6 `bdsDataChannelRef()` [2/2]

```
BString Bds::bdsDataChannelRef (
    const ChannelInfo & channelInfo )
```

Returns the string reference name of a [ChannelInfo](#) object.

6.1.2.7 `bdsDataFileSeedLogError()`

```
static void Bds::bdsDataFileSeedLogError (
    char * str ) [static]
```

6.1.2.8 `bdsDataFileSeedLogWarning()`

```
static void Bds::bdsDataFileSeedLogWarning (
    char * str ) [static]
```

6.1.2.9 `bdsDataInfoFromInfo()`

```
BError Bds::bdsDataInfoFromInfo (
    BDictString info,
    DataInfo & dataInfo,
    Bool append )
```

Convert info to [DataInfo](#).

Sets up a [DataInfo](#) object from a **BDictString** list of named strings.

6.1.2.10 `bdsDataInfoMergeFlatten()`

```
BError Bds::bdsDataInfoMergeFlatten (
    DataInfo & dataInfo,
    const DataInfo & dataInfoAdd )
```

Merges a [DataInfo](#) into another flattening the segments to 1 for use in `dataOpen()` calls.



#### 6.1.2.11 bdsDataInfoSetTimeRange()

```
BError Bds::bdsDataInfoSetTimeRange (
    DataInfo & dataInfo )
```

Restricts the time range of all of the DataInfo's channels to match the DataInfo's startTime/endTime fields.

#### 6.1.2.12 bdsDumpChannelInfos()

```
void Bds::bdsDumpChannelInfos (
    const ChannelInfos & channelInfos )
```

Debug print out a ChannelInfos object.

#### 6.1.2.13 bdsDumpData()

```
void Bds::bdsDumpData (
    const DataBlock & dataBlock,
    int nSamples )
```

Debug print out a DataBlock object.

#### 6.1.2.14 bdsDumpDataInfo()

```
void Bds::bdsDumpDataInfo (
    DataInfo dataInfo,
    int includeInfo )
```

Debug print out a DataInfo object.

#### 6.1.2.15 bdsDumpPoleZeros()

```
void Bds::bdsDumpPoleZeros (
    PoleZero poleZeros )
```

Debug print out a PoleZeros object.



#### 6.1.2.16 bdsDumpSelection()

```
void Bds::bdsDumpSelection (
    Selection sel )
```

Debug print out a [Selection](#) object.

#### 6.1.2.17 bdsFileNameExpand() [1/2]

```
BString Bds::bdsFileNameExpand (
    BString fileName,
    ChannelInfo & channelInfo )
```

Default filename from a [ChannelInfo](#).

#### 6.1.2.18 bdsFileNameExpand() [2/2]

```
BString Bds::bdsFileNameExpand (
    BString fileName,
    ChannelInfos & channelInfos )
```

Default filename from a list of [ChannelInfo](#)'s.

#### 6.1.2.19 bdsInfoFromDataInfo()

```
BError Bds::bdsInfoFromDataInfo (
    const DataInfo & dataInfo,
    BDictString & info )
```

Converts a [DataInfo](#) object into a BDictString list of named strings.

#### 6.1.2.20 bdsPoleZeroGain()

```
double Bds::bdsPoleZeroGain (
    const PoleZero & poleZero,
    double frequency )
```

Calculates the overall gain of the given [PoleZero](#) transfer function.



#### 6.1.2.21 bdsPoleZeroGainPhase()

```
void Bds::bdsPoleZeroGainPhase (
    const PoleZero & poleZero,
    double frequency,
    double & gain,
    double & phase )
```

Calculates the overall gain and phase of the given PoleZero transfer function.

#### 6.1.2.22 bdsPoleZeroToFap()

```
void Bds::bdsPoleZeroToFap (
    const PoleZero & poleZero,
    BUInt nPoints,
    double calibrationFrequency,
    double sampleFrequency,
    BArray< Fap > & fap )
```

Convert PoleZero to FAP.

#### 6.1.2.23 bdsSelectionChannelInfo()

```
BString Bds::bdsSelectionChannelInfo (
    const Selection & selection,
    BUInt channel )
```

Returns a string describing the name and time period of a selection channel.

#### 6.1.2.24 bdsStationAlias()

```
BString Bds::bdsStationAlias (
    Station station )
```

Returns the station alias if set else its name.

#### 6.1.2.25 bdsUnCompressCm8()

```
BError Bds::bdsUnCompressCm8 (
    BUInt8 * buffer,
    BUInt n,
    BArray< BInt32 > & data )
```

Uncompress CM8 formatted data.



#### 6.1.2.26 bdsUnCompressSteim1()

```
BError Bds::bdsUnCompressSteim1 (
    BUInt8 * buffer,
    BUInt n,
    BArray< BInt32 > & data )
```

Uncompress STEIM1 formatted data.

#### 6.1.2.27 crc()

```
BUInt32 Bds::crc (
    BUInt32 crc,
    void * data,
    int numBytes )
```

#### 6.1.2.28 crc64()

```
static uint64_t Bds::crc64 (
    const void * buffer,
    const uint32_t len ) [static]
```

#### 6.1.2.29 crcInit()

```
static void Bds::crcInit ( ) [static]
```

#### 6.1.2.30 dataCalculateDifference()

```
void Bds::dataCalculateDifference (
    BInt32 & prevValue,
    BArray< BInt32 > & data )
```

#### 6.1.2.31 dataCalculateUnDifference()

```
void Bds::dataCalculateUnDifference (
    BInt32 & prevValue,
    BArray< BInt32 > & data )
```



## 6.1.2.32 dataChecksum()

```
BInt32 Bds::dataChecksum (
    BInt32 checksum,
    BArray< BInt32 > & data )
```

## 6.1.2.33 dataCompressCm6()

```
BError Bds::dataCompressCm6 (
    int & prevValue1,
    int & prevValue2,
    BArray< BInt32 > & data,
    BString & d )
```

## 6.1.2.34 dataConvert() [1/3]

```
static void Bds::dataConvert (
    const BArray< BFloat64 > & dataIn,
    BArray< BInt32 > & dataOut ) [static]
```

## 6.1.2.35 dataConvert() [2/3]

```
static void Bds::dataConvert (
    const BArray< BFloat64 > & dataIn,
    BArray< BFloat32 > & dataOut ) [static]
```

## 6.1.2.36 dataConvert() [3/3]

```
static void Bds::dataConvert (
    const BArray< BFloat64 > & dataIn,
    BArray< BInt32 > & dataOut ) [static]
```

## 6.1.2.37 dataDeCompressCm6()

```
BError Bds::dataDeCompressCm6 (
    int & prevValue1,
    int & prevValue2,
    BString & d,
    BArray< BInt32 > & data )
```



#### 6.1.2.38 duplicateDump()

```
int Bds::duplicateDump (
    DataBlock & data1,
    DataBlock & data2,
    int channel )
```

#### 6.1.2.39 fileNameTime()

```
static BString Bds::fileNameTime (
    BTimeStamp t ) [static]
```

#### 6.1.2.40 fixedString()

```
static BError Bds::fixedString (
    double v,
    int fieldWidth,
    int numDecimal,
    BString & str ) [static]
```

#### 6.1.2.41 fixedWidthValue()

```
BString Bds::fixedWidthValue (
    double v,
    int width )
```

This returns a double as a fixed width string truncating the data.

#### 6.1.2.42 getHexString()

```
BString Bds::getHexString (
    char * data,
    int len )
```

#### 6.1.2.43 nullString()

```
static BString Bds::nullString (
    BString s ) [static]
```



#### 6.1.2.44 record\_handler()

```
static void Bds::record_handler (
    char * record,
    int reclen,
    void * info ) [static]
```

#### 6.1.2.45 removeCR()

```
static BString Bds::removeCR (
    BString str ) [static]
```

#### 6.1.2.46 seedTime()

```
static hptime_t Bds::seedTime (
    BTimeStamp t ) [static]
```

#### 6.1.2.47 seedTimeString()

```
static BString Bds::seedTimeString (
    BTimeStamp t ) [static]
```

#### 6.1.2.48 stringFormat()

```
static BString Bds::stringFormat (
    BTimeStamp t ) [static]
```

### 6.1.3 Variable Documentation

#### 6.1.3.1 apiVersion

```
const BUInt32 Bds::apiVersion = 0
```



### 6.1.3.2 BdsDataFileVersion

```
const BString Bds::BdsDataFileVersion = "1.2.0"
```

### 6.1.3.3 ChannelAuxLen

```
const int Bds::ChannelAuxLen = 2
```

Maximum [Channel](#) Aux length.

### 6.1.3.4 ChannelTypeLen

```
const int Bds::ChannelTypeLen = 3
```

Maximum [Channel](#) type name length.

### 6.1.3.5 cm6Table

```
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'
}
```

### 6.1.3.6 cm6TableRev

```
BUInt8 Bds::cm6TableRev[128] [static]
```

Initial value:

```
= {
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0x00, 0xff, 0x01, 0xff, 0xff,
    0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x09,
    0x0a, 0x0b, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0x0c, 0x0d, 0x0e, 0x0f, 0x10, 0x11, 0x12,
    0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1a,
    0x1b, 0x1c, 0x1d, 0x1e, 0x1f, 0x20, 0x21, 0x22,
    0x23, 0x24, 0x25, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0x26, 0x27, 0x28, 0x29, 0x2a, 0x2b, 0x2c,
    0x2d, 0x2e, 0x2f, 0x30, 0x31, 0x32, 0x33, 0x34,
    0x35, 0x36, 0x37, 0x38, 0x39, 0x3a, 0x3b, 0x3c,
    0x3d, 0x3e, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff,
}
```



#### 6.1.3.7 crcInitDone

```
int Bds::crcInitDone [static]
```

#### 6.1.3.8 crcVec

```
uint64_t Bds::crcVec[256] [static]
```

#### 6.1.3.9 dataFormats

```
DataFormats Bds::dataFormats
```

#### 6.1.3.10 NetworkNameLen

```
const int Bds::NetworkNameLen = 3
```

Maximum [Network](#) name length.

#### 6.1.3.11 Scale

```
const double Bds::Scale = 16777216.0
```

#### 6.1.3.12 SourceLen

```
const int Bds::SourceLen = 16
```

Maximum [Source](#) length.

#### 6.1.3.13 StationNameLen

```
const int Bds::StationNameLen = 5
```

Maximum [Station](#) name length.







## Chapter 7

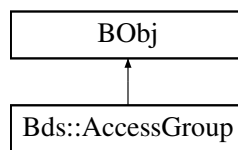
# Class Documentation

### 7.1 Bds::AccessGroup Class Reference

This holds information on data access groups.

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

Inheritance diagram for Bds::AccessGroup:



#### Public Member Functions

- **AccessGroup** ( **BUInt32** id=0, **BString** group= **BString**(), **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), **BString** network= **BString**(), **BString** station= **BString**())
- **BString** getType ()
- **BError** setMembers ( **BDictString** &members)
- **BError** setMember ( **BString** name, **BString** value)
- **BError** getMembers ( **BDictString** &members)
- **BError** getMember ( **BString** name, **BString** &value)

#### 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.*



### 7.1.1 Detailed Description

This holds information on data access groups.

A particular network:station may contain sensitive data. This database linked object links a period of data from a particular network:station to a security group.

### 7.1.2 Constructor & Destructor Documentation

#### 7.1.2.1 AccessGroup()

```
Bds::AccessGroup::AccessGroup (
    BUInt32 id = 0,
    BString group = BString(),
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString network = BString(),
    BString station = BString() )
```

### 7.1.3 Member Function Documentation

#### 7.1.3.1 getMember()

```
BError Bds::AccessGroup::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.1.3.2 getMembers()

```
BError Bds::AccessGroup::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

#### 7.1.3.3 getType()

```
BString Bds::AccessGroup::getType ( )
```



#### 7.1.3.4 setMember()

```
BError Bds::AccessGroup::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.1.3.5 setMembers()

```
BError Bds::AccessGroup::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.1.4 Member Data Documentation

#### 7.1.4.1 endTime

```
BTimeStamp Bds::AccessGroup::endTime
```

The End Time.

#### 7.1.4.2 group

```
BString Bds::AccessGroup::group
```

The [Group](#) name.

#### 7.1.4.3 id

```
BUInt32 Bds::AccessGroup::id
```

The unique id.



#### 7.1.4.4 network

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

The [Network](#) Name.

#### 7.1.4.5 startTime

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

The Start Time.

#### 7.1.4.6 station

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

The [Station](#) name.

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

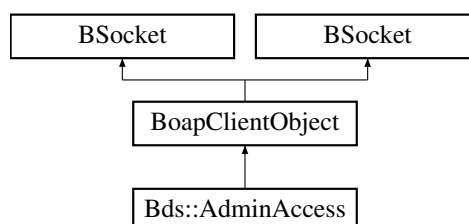
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.2 Bds::AdminAccess Class Reference

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

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

Inheritance diagram for Bds::AdminAccess:





## Public Member Functions

- [AdminAccess](#) ( **BString** name="" )
- **BError** [connect](#) ( **BString** user, **BString** password )  
*Provides user/password information.*
- **BError** [validateUser](#) ( **BString** user, **BString** email )  
*Checks the user given name or email.*
- **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, **BString** &name )  
*Gets the software version and server name.*
- **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** [userGetFromId](#) ( **BUInt32** id, [User](#) &user )  
*Get user info given user ID.*
- **BError** [userGet](#) ( [User](#) &user )  
*Get user info.*
- **BError** [userSet](#) ( [User](#) user )  
*Set user info.*
- **BError** [userGetGroups](#) ( **BList**< **BString** > &groups )  
*Get list of groups the user belongs to.*
- **BError** [groupGetList](#) ( **BList**< [Group](#) > &groups )  
*Get list of Groups.*
- **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, [Selection](#) selectionIn, [SelectionInfo](#) &selectionInfo )  
*Get information on possible selections. Use in GUI programs to list options available.*
- **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 )  
*Add or update a [Network](#) entry.*
- **BError** [networkDelete](#) ( **BUInt32** id )  
*Delete a [Network](#) entry.*
- **BError** [stationGetList](#) ( [Selection](#) sel, **BList**< [Station](#) > &stations )  
*Get list of Stations.*



- **BError** [stationUpdate](#) ( **BInt32** append, [Station](#) station, **BUInt32** &id)  
*Add or update a [Station](#) entry.*
- **BError** [stationDelete](#) ( **BUInt32** id)  
*Delete a [Station](#) entry.*
- **BError** [locationGetList](#) ( [Selection](#) sel, **BList**< [Location](#) > &locations)  
*Get list of [Station](#) Locations.*
- **BError** [locationUpdate](#) ( **BInt32** append, [Location](#) location, **BUInt32** &id)  
*Add or update a [Station Location](#) entry.*
- **BError** [locationDelete](#) ( **BUInt32** id)  
*Delete a [Station Location](#) entry.*
- **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)  
*Add or update a [Channel](#) entry.*
- **BError** [channelDelete](#) ( **BUInt32** id)  
*Delete a [Channel](#) entry.*
- **BError** [sourceGetList](#) ( **BList**< [Source](#) > &sources)  
*Get list of Sources.*
- **BError** [sourceUpdate](#) ( **BInt32** append, [Source](#) source, **BUInt32** &id)  
*Add or update a [Source](#) entry.*
- **BError** [sourceDelete](#) ( **BUInt32** id)  
*Delete a [Source](#) entry.*
- **BError** [sourcePriorityGetList](#) ( **BList**< [SourcePriority](#) > &sourcePriorities)  
*Get list of SourcePriorities.*
- **BError** [sourcePriorityUpdate](#) ( **BInt32** append, [SourcePriority](#) sourcePriority, **BUInt32** &id)
- **BError** [sourcePriorityDelete](#) ( **BUInt32** id)
- **BError** [channellInstrumentGetList](#) ( [Selection](#) sel, **BList**< [ChannellInstrument](#) > &channellInstruments)  
*Get list of [Channel](#) Instruments.*
- **BError** [channellInstrumentUpdate](#) ( **BInt32** append, [ChannellInstrument](#) channellInstrument, **BUInt32** &id)  
*Add or update a Instrument entry.*
- **BError** [channellInstrumentDelete](#) ( **BUInt32** id)  
*Delete an Instrument entry.*
- **BError** [digitiserGetList](#) ( [Selection](#) sel, **BList**< [Digitiser](#) > &digitisers)  
*Get list of Digitisers.*
- **BError** [digitiserGet](#) ( **BUInt32** id, [Digitiser](#) &digitiser)  
*Get a [Digitiser](#) object given its ID.*
- **BError** [digitiserUpdate](#) ( **BInt32** append, [Digitiser](#) digitiser, **BUInt32** &id)  
*Add or update a [Digitiser](#) entry.*
- **BError** [digitiserDelete](#) ( **BUInt32** id)  
*Delete a [Digitiser](#) entry.*
- **BError** [sensorGetList](#) ( [Selection](#) sel, **BList**< [Sensor](#) > &sensors)  
*Get list of Sensors.*
- **BError** [sensorGet](#) ( **BUInt32** id, [Sensor](#) &sensor)  
*Get a [Sensor](#) object given its ID.*
- **BError** [sensorUpdate](#) ( **BInt32** append, [Sensor](#) sensor, **BUInt32** &id)  
*Add or update a [Sensor](#) entry.*
- **BError** [sensorDelete](#) ( **BUInt32** id)  
*Delete a [Sensor](#) entry.*
- **BError** [calibrationGetList](#) ( [Selection](#) sel, **BList**< [Calibration](#) > &calibrations)



- Get list of Calibrations.*

  - **BError** [calibrationUpdate](#) (**BInt32** append, [Calibration](#) calibration, **BUInt32** &id)

*Add or update a [Calibration](#) entry.*
- **BError** [calibrationDelete](#) (**BUInt32** id)

*Delete a [Calibration](#) entry.*
- **BError** [responseGetList](#) ([Selection](#) sel, **BList**< [Response](#) > &responses)

*Get list of Responses.*
- **BError** [responseUpdate](#) (**BInt32** append, [Response](#) response, **BUInt32** &id)

*Add or update a [Response](#) entry.*
- **BError** [responseDelete](#) (**BUInt32** id)

*Delete a [Response](#) entry.*
- **BError** [dataFileGetList](#) ([Selection](#) sel, **BList**< [DataFileInfo](#) > &dataFile)

*Get list of DataFiles.*
- **BError** [dataFileUpdate](#) (**BInt32** append, [DataFileInfo](#) dataFile, **BUInt32** &id)

*Add or update a [DataFile](#) entry.*
- **BError** [dataFileDelete](#) (**BUInt32** id)

*Delete a [DataFile](#) entry.*
- **BError** [dataChannelGetList](#) ([Selection](#) sel, **BList**< [DataChannel](#) > &dataChannel)

*Get list of DataChannels.*
- **BError** [dataChannelUpdate](#) (**BInt32** append, [DataChannel](#) dataChannel, **BUInt32** &id)

*Add or update a [DataChannel](#) entry.*
- **BError** [dataChannelDelete](#) (**BUInt32** id)

*Delete a [DataChannel](#) entry.*
- **BError** [dataAvailability](#) ([Selection](#) selection, **BUInt32** num, **BArray**< [DataAvailChan](#) > &dataAvailChans)

*Return availability for data matching the given selection parameters.*
- **BError** [dataSearch](#) ([Selection](#) selection, [DataInfo](#) &dataInfo)

*Search for data matching the given selection parameters.*
- **BError** [dataGetChannelInfo](#) ([DataInfo](#) dataInfo, [ChannelInfos](#) &channelInfos)

*Return the channel MetaData in structured form.*
- **BError** [dataOpen](#) ([DataInfo](#) dataInfo, **BString** mode, **BString** format, **BUInt32** flags, [DataHandle](#) &dataHandle)

*Open a data file.*
- **BError** [dataGetInfo](#) ([DataHandle](#) dataHandle, **BUInt32** infoExtra, [DataInfo](#) &dataInfo)

*Get information on the data file.*
- **BError** [dataGetNotes](#) ([DataHandle](#) dataHandle, **BList**< [Note](#) > &notes)

*Get notes on the data file.*
- **BError** [dataGetWarnings](#) ([DataHandle](#) dataHandle, **BList**< **BString** > &warnings)

*Get information on the data file.*
- **BError** [dataGetBlock](#) ([DataHandle](#) dataHandle, **BUInt32** channel, **BUInt32** segment, **BUInt32** blockNumber, [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 file.*
- **BError** [dataPutBlock](#) ([DataHandle](#) dataHandle, [DataBlock](#) data)

*Send a block of data.*
- **BError** [dataClose](#) ([DataHandle](#) dataHandle, **BError** error, **BInt32** del)

*Close a file.*
- **BError** [dataFormattedRead](#) ([DataHandle](#) dataHandle, **BUInt32** number, **BArray**< **BUInt8** > & data)



- Read the raw data from the file.*

  - **BError** `dataFormattedGetLength` (`DataHandle` dataHandle, `BUInt64` & length)
- Read the raw data from the file.*

  - **BError** `changeGroupStart` (`ChangeGroup` changeGroup)

*Start a new `ChangeGroup` when making a set of changes to the BDS's database.*
- **BError** `changeGroupEnd` ()

*End a `ChangeGroup`.*
- **BError** `changeGroupGetList` (`ListRange` range, `BList`< `ChangeGroup` > &changeGroups)

*Return a list of `ChangeGroups`.*
- **BError** `changeGroupDelete` (`BTimeStamp` beforeDate, `BString` type, `BInt32` empty)

*Delete a `ChangeGroup`.*
- **BError** `changeGetListNumber` (`BUInt32` id, `BUInt32` & number)

*Get the number of changes in a `ChangeGroup`.*
- **BError** `changeGetList` (`BUInt32` id, `ListRange` range, `BList`< `Change` > &changes)

*Get a list of `Changes`.*
- **BError** `changeDelete` (`BTimeStamp` beforeDate, `BString` type)

*Delete a `Change`.*
- **BError** `noteGetList` (`Selection` sel, `BList`< `Note` > &notes)

*Get a list of `Notes`.*
- **BError** `noteUpdate` (`BInt32` append, `Note` note, `BUInt32` &id)

*Add or update a `Note`.*
- **BError** `noteDelete` (`BUInt32` id)

*Delete a `Note`.*
- **BError** `noteWriteDocument` (`BUInt32` id, `BString` format, `BArray`< `BUInt8` > data)

*Given a `Note` write a document associated with it.*
- **BError** `noteReadDocument` (`BUInt32` id, `BString` &format, `BArray`< `BUInt8` > & data)

*Read a document associated with a `Note`.*
- **BError** `logGetList` (`LogSelect` sel, `BList`< `Log` > &logs)

*Get list of log entries.*
- **BError** `logUpdate` (`BInt32` append, `Log` log, `BUInt32` &id)

*Add or Update a `Log` item.*
- **BError** `logDelete` (`BUInt32` id)

*Delete a `Log` item.*
- **BError** `logAppend` (`BString` type, `BUInt32` priority, `BString` subSystem, `BString` title, `BString` description)

*Append a log item.*
- **BError** `statisticsGet` (`BDict`< `BString` > &info)

*Get a list of system statistics.*
- **BError** `dataFormatGetList` (`BList`< `DataFormat` > &formats)

*Get list of supported data formats.*
- **BError** `transactionStart` ()

*Starts a set of transactions.*
- **BError** `transactionEnd` (`BInt32` abort)

*Ends a set of transactions.*
- **BError** `modeSet` (`Mode` mode, `Mode` &previousMode)

*Changes the system mode from Master to slave.*
- **BError** `modeSnapshotPause` (`BInt32` on)

*Enables/disables backup synchronisation pause.*
- **BError** `clean` (`CleanOptions` cleanOptions)

*Cleans the system logs and Changes information.*
- **BError** `databaseBackup` (`BString` &ref)

*Backup the database.*



- **BError** [databaseRestore](#) ( **BString** ref, **BString** type)  
*Restore the database.*
- **BError** [sqlQuery](#) ( **BString** query, **BList**< **BDict**< **BString** > > &result)  
*A low level SQL access function.*

## Additional Inherited Members

### 7.2.1 Detailed Description

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

This object provides the set of API functions that make RPC network calls to a BdsServer. It provides the full unresiticted data access API allowing all data read and write operations to be performed.

### 7.2.2 Constructor & Destructor Documentation

#### 7.2.2.1 AdminAccess()

```
Bds::AdminAccess::AdminAccess (
    BString name = "" )
```

### 7.2.3 Member Function Documentation

#### 7.2.3.1 accessGroupDelete()

```
BError Bds::AdminAccess::accessGroupDelete (
    BUInt32 id )
```

Delete an [AccessGroup](#) entry.

#### 7.2.3.2 accessGroupGetList()

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

Get list of AccessGroups.



### 7.2.3.3 accessGroupUpdate()

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

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

### 7.2.3.4 calibrationDelete()

```
BError Bds::AdminAccess::calibrationDelete (
    BUInt32 id )
```

Delete a [Calibration](#) entry.

### 7.2.3.5 calibrationGetList()

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

Get list of Calibrations.

### 7.2.3.6 calibrationUpdate()

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

Add or update a [Calibration](#) entry.

### 7.2.3.7 changeDelete()

```
BError Bds::AdminAccess::changeDelete (
    BTimeStamp beforeDate,
    BString type )
```

Delete a [Change](#).



#### 7.2.3.8 changeGetList()

```
BError Bds::AdminAccess::changeGetList (
    BUInt32 id,
    ListRange range,
    BList< Change > & changes )
```

Get a list of Changes.

#### 7.2.3.9 changeGetListNumber()

```
BError Bds::AdminAccess::changeGetListNumber (
    BUInt32 id,
    BUInt32 & number )
```

Get the number of changes in a [ChangeGroup](#).

#### 7.2.3.10 changeGroupDelete()

```
BError Bds::AdminAccess::changeGroupDelete (
    BTimeStamp beforeDate,
    BString type,
    BInt32 empty )
```

Delete a [ChangeGroup](#).

#### 7.2.3.11 changeGroupEnd()

```
BError Bds::AdminAccess::changeGroupEnd ( )
```

End a [ChangeGroup](#).

#### 7.2.3.12 changeGroupGetList()

```
BError Bds::AdminAccess::changeGroupGetList (
    ListRange range,
    BList< ChangeGroup > & changeGroups )
```

Return a list of ChangeGroups.



### 7.2.3.13 changeGroupStart()

```
BError Bds::AdminAccess::changeGroupStart (
    ChangeGroup changeGroup )
```

Start a new [ChangeGroup](#) when making a set of changes to the BDS's database.

### 7.2.3.14 channelDelete()

```
BError Bds::AdminAccess::channelDelete (
    BUInt32 id )
```

Delete a [Channel](#) entry.

### 7.2.3.15 channelGet()

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

Get a channel.

### 7.2.3.16 channelGetList()

```
BError Bds::AdminAccess::channelGetList (
    Selection sel,
    BList< Channel > & channels )
```

Get list of Channels.

### 7.2.3.17 channelInstrumentDelete()

```
BError Bds::AdminAccess::channelInstrumentDelete (
    BUInt32 id )
```

Delete an Instrument entry.



### 7.2.3.18 channelInstrumentGetList()

```
BError Bds::AdminAccess::channelInstrumentGetList (
    Selection sel,
    BList< ChannelInstrument > & channelInstruments )
```

Get list of [Channel](#) Instruments.

### 7.2.3.19 channelInstrumentUpdate()

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

Add or update a Instrument entry.

### 7.2.3.20 channelUpdate()

```
BError Bds::AdminAccess::channelUpdate (
    BInt32 append,
    Channel channel,
    BUInt32 & id )
```

Add or update a [Channel](#) entry.

### 7.2.3.21 clean()

```
BError Bds::AdminAccess::clean (
    CleanOptions cleanOptions )
```

Cleans the system logs and Changes information.

### 7.2.3.22 connect()

```
BError Bds::AdminAccess::connect (
    BString user,
    BString password )
```

Provides user/password information.



#### 7.2.3.23 dataAvailability()

```
BError Bds::AdminAccess::dataAvailability (
    Selection selection,
    BUInt32 num,
    BArray< DataAvailChan > & dataAvailChans )
```

Return availability for data matching the given selection parameters.

#### 7.2.3.24 databaseBackup()

```
BError Bds::AdminAccess::databaseBackup (
    BString & ref )
```

Backup the database.

#### 7.2.3.25 databaseRestore()

```
BError Bds::AdminAccess::databaseRestore (
    BString ref,
    BString type )
```

Restore the database.

#### 7.2.3.26 dataChannelDelete()

```
BError Bds::AdminAccess::dataChannelDelete (
    BUInt32 id )
```

Delete a *DataChannel* entry.

#### 7.2.3.27 dataChannelGetList()

```
BError Bds::AdminAccess::dataChannelGetList (
    Selection sel,
    BList< DataChannel > & dataChannel )
```

Get list of DataChannels.



### 7.2.3.28 dataChannelUpdate()

```
BError Bds::AdminAccess::dataChannelUpdate (
    BInt32 append,
    DataChannel dataChannel,
    BUInt32 & id )
```

Add or update a [DataChannel](#) entry.

### 7.2.3.29 dataClose()

```
BError Bds::AdminAccess::dataClose (
    DataHandle dataHandle,
    BError error,
    BInt32 del )
```

Close a file.

### 7.2.3.30 dataFileDelete()

```
BError Bds::AdminAccess::dataFileDelete (
    BUInt32 id )
```

Delete a [DataFile](#) entry.

### 7.2.3.31 dataFileGetList()

```
BError Bds::AdminAccess::dataFileGetList (
    Selection sel,
    BList< DataFileInfo > & dataFile )
```

Get list of DataFiles.

### 7.2.3.32 dataFileUpdate()

```
BError Bds::AdminAccess::dataFileUpdate (
    BInt32 append,
    DataFileInfo dataFile,
    BUInt32 & id )
```

Add or update a [DataFile](#) entry.



### 7.2.3.33 dataFormatGetList()

```
BError Bds::AdminAccess::dataFormatGetList (
    BList< DataFormat > & formats )
```

Get list of supported data formats.

### 7.2.3.34 dataFormattedGetLength()

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

Read the raw data from the file.

### 7.2.3.35 dataFormattedRead()

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

Read the raw data from the file.

### 7.2.3.36 dataGetBlock()

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

Return a block of data.

### 7.2.3.37 dataGetChannelInfo()

```
BError Bds::AdminAccess::dataGetChannelInfo (
    DataInfo dataInfo,
    ChannelInfos & channelInfos )
```

Return the channel MetaData in structured form.



## 7.2.3.38 dataGetInfo()

```
BError Bds::AdminAccess::dataGetInfo (
    DataHandle dataHandle,
    BUInt32 infoExtra,
    DataInfo & dataInfo )
```

Get information on the data file.

## 7.2.3.39 dataGetNotes()

```
BError Bds::AdminAccess::dataGetNotes (
    DataHandle dataHandle,
    BList< Note > & notes )
```

Get notes on the data file.

## 7.2.3.40 dataGetWarnings()

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

Get information on the data file.

## 7.2.3.41 dataOpen()

```
BError Bds::AdminAccess::dataOpen (
    DataInfo dataInfo,
    BString mode,
    BString format,
    BUInt32 flags,
    DataHandle & dataHandle )
```

Open a data file.

## Parameters

in	<i>dataInfo</i>	The sensor data to open selection
in	<i>mode</i>	The open format. The mode should be set to "w" for writing data, "a" when appending data and "r" for reading data.
in	<i>format</i>	What format to open the data stream as. This can be API for raw BDS API access or one of the supported formats that the BDS is able to convert to.
in	<i>flags</i>	A bitset of flags as defined by <a href="#">Bds::DataFlags</a> .
out	<i>dataHandle</i>	The handle for the open data set



#### 7.2.3.42 dataPutBlock()

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

Send a block of data.

#### 7.2.3.43 dataSearch()

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

Search for data matching the given selection parameters.

#### 7.2.3.44 dataSeekBlock()

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

Searches for a data block matching the time given.

#### 7.2.3.45 dataSetInfo()

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

Set the info when writing to a file.

#### 7.2.3.46 digitiserDelete()

```
BError Bds::AdminAccess::digitiserDelete (
    BUInt32 id )
```

Delete a [Digitiser](#) entry.



## 7.2.3.47 digitiserGet()

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

Get a [Digitiser](#) object given its ID.

## 7.2.3.48 digitiserGetList()

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

Get list of Digitisers.

## 7.2.3.49 digitiserUpdate()

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

Add or update a [Digitiser](#) entry.

## 7.2.3.50 getSelectionInfo()

```
BError Bds::AdminAccess::getSelectionInfo (
    SelectionGroup group,
    Selection selectionIn,
    SelectionInfo & selectionInfo )
```

Get information on possible selections. Use in GUI programs to list options available.

## 7.2.3.51 getSelections()

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

Get selection list.



#### 7.2.3.52 getVersion()

```
BError Bds::AdminAccess::getVersion (
    BString & version,
    BString & name )
```

Gets the software version and server name.

#### 7.2.3.53 groupDelete()

```
BError Bds::AdminAccess::groupDelete (
    BUInt32 id )
```

Delete a group entry.

#### 7.2.3.54 groupGetList()

```
BError Bds::AdminAccess::groupGetList (
    BList< Group > & groups )
```

Get list of Groups.

#### 7.2.3.55 groupUpdate()

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

Update or append a group entry.

#### 7.2.3.56 locationDelete()

```
BError Bds::AdminAccess::locationDelete (
    BUInt32 id )
```

Delete a [Station Location](#) entry.



## 7.2.3.57 locationGetList()

```
BError Bds::AdminAccess::locationGetList (
    Selection sel,
    BList< Location > & locations )
```

Get list of [Station](#) Locations.

## 7.2.3.58 locationUpdate()

```
BError Bds::AdminAccess::locationUpdate (
    BInt32 append,
    Location location,
    BUInt32 & id )
```

Add or update a [Station Location](#) entry.

## 7.2.3.59 logAppend()

```
BError Bds::AdminAccess::logAppend (
    BString type,
    BUInt32 priority,
    BString subSystem,
    BString title,
    BString description )
```

Append a log item.

## 7.2.3.60 logDelete()

```
BError Bds::AdminAccess::logDelete (
    BUInt32 id )
```

Delete a [Log](#) item.

## 7.2.3.61 logGetList()

```
BError Bds::AdminAccess::logGetList (
    LogSelect sel,
    BList< Log > & logs )
```

Get list of log entries.



### 7.2.3.62 logUpdate()

```
BError Bds::AdminAccess::logUpdate (
    BInt32 append,
    Log log,
    BUInt32 & id )
```

Add or Update a [Log](#) item.

### 7.2.3.63 modeSet()

```
BError Bds::AdminAccess::modeSet (
    Mode mode,
    Mode & previousMode )
```

Changes the system mode from Master to slave.

### 7.2.3.64 modeSnapshotPause()

```
BError Bds::AdminAccess::modeSnapshotPause (
    BInt32 on )
```

Enables/disables backup synchronisation pause.

### 7.2.3.65 networkDelete()

```
BError Bds::AdminAccess::networkDelete (
    BUInt32 id )
```

Delete a [Network](#) entry.

### 7.2.3.66 networkGetList()

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

Get list of Networks.



## 7.2.3.67 networkUpdate()

```
BError Bds::AdminAccess::networkUpdate (
    BInt32 append,
    Network network,
    BUInt32 & id )
```

Add or update a [Network](#) entry.

## 7.2.3.68 noteDelete()

```
BError Bds::AdminAccess::noteDelete (
    BUInt32 id )
```

Delete a [Note](#).

## 7.2.3.69 noteGetList()

```
BError Bds::AdminAccess::noteGetList (
    Selection sel,
    BList< Note > & notes )
```

Get a list of Notes.

## 7.2.3.70 noteReadDocument()

```
BError Bds::AdminAccess::noteReadDocument (
    BUInt32 id,
    BString & format,
    BArray< BUInt8 > & data )
```

Read a document associated with a [Note](#).

## 7.2.3.71 noteUpdate()

```
BError Bds::AdminAccess::noteUpdate (
    BInt32 append,
    Note note,
    BUInt32 & id )
```

Add or update a [Note](#).



#### 7.2.3.72 noteWriteDocument()

```
BError Bds::AdminAccess::noteWriteDocument (
    BUInt32 id,
    BString format,
    BArray< BUInt8 > data )
```

Given a [Note](#) write a document associated with it.

#### 7.2.3.73 responseDelete()

```
BError Bds::AdminAccess::responseDelete (
    BUInt32 id )
```

Delete a [Response](#) entry.

#### 7.2.3.74 responseGetList()

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

Get list of Responses.

#### 7.2.3.75 responseUpdate()

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

Add or update a [Response](#) entry.

#### 7.2.3.76 sensorDelete()

```
BError Bds::AdminAccess::sensorDelete (
    BUInt32 id )
```

Delete a [Sensor](#) entry.



## 7.2.3.77 sensorGet()

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

Get a [Sensor](#) object given its ID.

## 7.2.3.78 sensorGetList()

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

Get list of Sensors.

## 7.2.3.79 sensorUpdate()

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

Add or update a [Sensor](#) entry.

## 7.2.3.80 setUser()

```
BError Bds::AdminAccess::setUser (
    BString user,
    BString email )
```

Sets user to given name or email.

## 7.2.3.81 setUserReal()

```
BError Bds::AdminAccess::setUserReal ( )
```

Sets user back to real user.



#### 7.2.3.82 sourceDelete()

```
BError Bds::AdminAccess::sourceDelete (
    BUInt32 id )
```

Delete a [Source](#) entry.

#### 7.2.3.83 sourceGetList()

```
BError Bds::AdminAccess::sourceGetList (
    BList< Source > & sources )
```

Get list of Sources.

#### 7.2.3.84 sourcePriorityDelete()

```
BError Bds::AdminAccess::sourcePriorityDelete (
    BUInt32 id )
```

#### 7.2.3.85 sourcePriorityGetList()

```
BError Bds::AdminAccess::sourcePriorityGetList (
    BList< SourcePriority > & sourcePrioritys )
```

Get list of SourcePriorities.

#### 7.2.3.86 sourcePriorityUpdate()

```
BError Bds::AdminAccess::sourcePriorityUpdate (
    BInt32 append,
    SourcePriority sourcePriority,
    BUInt32 & id )
```

#### 7.2.3.87 sourceUpdate()

```
BError Bds::AdminAccess::sourceUpdate (
    BInt32 append,
    Source source,
    BUInt32 & id )
```

Add or update a [Source](#) entry.



7.2.3.88 `sqlQuery()`

```
BError Bds::AdminAccess::sqlQuery (
    BString query,
    BList< BDict< BString > > & result )
```

A low level SQL access function.

7.2.3.89 `stationDelete()`

```
BError Bds::AdminAccess::stationDelete (
    BUInt32 id )
```

Delete a [Station](#) entry.

7.2.3.90 `stationGetList()`

```
BError Bds::AdminAccess::stationGetList (
    Selection sel,
    BList< Station > & stations )
```

Get list of Stations.

7.2.3.91 `stationUpdate()`

```
BError Bds::AdminAccess::stationUpdate (
    BInt32 append,
    Station station,
    BUInt32 & id )
```

Add or update a [Station](#) entry.

7.2.3.92 `statisticsGet()`

```
BError Bds::AdminAccess::statisticsGet (
    BDict< BString > & info )
```

Get a list of system statistics.



### 7.2.3.93 transactionEnd()

```
BError Bds::AdminAccess::transactionEnd (
    BInt32 abort )
```

Ends a set of transactions.

### 7.2.3.94 transactionStart()

```
BError Bds::AdminAccess::transactionStart ( )
```

Starts a set of transactions.

### 7.2.3.95 userDelete()

```
BError Bds::AdminAccess::userDelete (
    BUInt32 id )
```

Delete a user entry.

### 7.2.3.96 userGet()

```
BError Bds::AdminAccess::userGet (
    User & user )
```

Get user info.

### 7.2.3.97 userGetFromId()

```
BError Bds::AdminAccess::userGetFromId (
    BUInt32 id,
    User & user )
```

Get user info given user ID.

### 7.2.3.98 userGetGroups()

```
BError Bds::AdminAccess::userGetGroups (
    BList< BString > & groups )
```

Get list of groups the user belongs to.



## 7.2.3.99 userGetList()

```
BError Bds::AdminAccess::userGetList (
    BList< User > & users )
```

Get list of Users.

## 7.2.3.100 userSet()

```
BError Bds::AdminAccess::userSet (
    User user )
```

Set user info.

## 7.2.3.101 updateUser()

```
BError Bds::AdminAccess::updateUser (
    BInt32 append,
    User user,
    BUInt32 & id )
```

Update or append a user entry.

## 7.2.3.102 validateUser()

```
BError Bds::AdminAccess::validateUser (
    BString user,
    BString email )
```

Checks the user given name or email.

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

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

## 7.3 Bds::ArrayChannel Class Reference

This class defines an arrays channel.

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



## Public Member Functions

- [ArrayChannel](#) ( [BString](#) station= [BString](#)(), [BString](#) channel= [BString](#)() )

## Public Attributes

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

### 7.3.1 Detailed Description

This class defines an arrays channel.

A [Station](#) could be an individual station or an array of stations. If it is an array of stations the [Station](#) object contains the list of Stations and Channels that make up the array.

### 7.3.2 Constructor & Destructor Documentation

#### 7.3.2.1 ArrayChannel()

```
Bds::ArrayChannel::ArrayChannel (
    BString station = BString(),
    BString channel = BString() )
```

### 7.3.3 Member Data Documentation

#### 7.3.3.1 channel

```
BString Bds::ArrayChannel::channel
```

The Channels name.

#### 7.3.3.2 station

```
BString Bds::ArrayChannel::station
```

The Stations name.

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

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



## 7.4 Bds::BdsDataBlock Struct Reference

BdsDataFileBds: internal fixed size BDS Data Block.

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

### Public Attributes

- [BdsDataBlockHeader](#) header
- char [data](#) [4]

*The packet data.*

### 7.4.1 Detailed Description

BdsDataFileBds: internal fixed size BDS Data Block.

### 7.4.2 Member Data Documentation

#### 7.4.2.1 data

```
char Bds::BdsDataBlock::data[4]
```

The packet data.

#### 7.4.2.2 header

[BdsDataBlockHeader](#) Bds::BdsDataBlock::header

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

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

## 7.5 Bds::BdsDataBlockHeader Struct Reference

BdsDataFileBds: internal fixed size BDS Data Block header.

```
#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.*

### 7.5.1 Detailed Description

BdsDataFileBds: internal fixed size BDS Data Block header.

### 7.5.2 Member Data Documentation

#### 7.5.2.1 length

**BUInt32** Bds::BdsDataBlockHeader::length

Packet length.

#### 7.5.2.2 packetOffset

**BUInt32** Bds::BdsDataBlockHeader::packetOffset

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

#### 7.5.2.3 type

**BUInt32** Bds::BdsDataBlockHeader::type

Blocks type, contains magic number for synchronisation.

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

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



## 7.6 Bds::BdsDataBlockPos Class Reference

BdsDataFileBds: internal file storage data block position.

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

### Public Member Functions

- [BdsDataBlockPos](#) ( [BTimeStamp](#) *startTime*=0, [BTimeStamp](#) *endTime*=0, [BUInt32](#) *channel*=0, [BUInt32](#) *numChannels*=0, [BUInt32](#) *segment*=0, [BUInt64](#) *position*=0, [BUInt64](#) *numSamples*=0)
- [int operator<](#) (const [BdsDataBlockPos](#) &b) const

### Public Attributes

- [BTimeStamp](#) *startTime*
- [BTimeStamp](#) *endTime*
- [BUInt32](#) *channel*
- [BUInt32](#) *numChannels*
- [BUInt32](#) *segment*
- [BUInt64](#) *position*
- [BUInt64](#) *numSamples*

### 7.6.1 Detailed Description

BdsDataFileBds: internal file storage data block position.

### 7.6.2 Constructor & Destructor Documentation

#### 7.6.2.1 BdsDataBlockPos()

```
Bds::BdsDataBlockPos::BdsDataBlockPos (
    BTimeStamp startTime = 0,
    BTimeStamp endTime = 0,
    BUInt32 channel = 0,
    BUInt32 numChannels = 0,
    BUInt32 segment = 0,
    BUInt64 position = 0,
    BUInt64 numSamples = 0 ) [inline]
```

### 7.6.3 Member Function Documentation



### 7.6.3.1 operator<()

```
int Bds::BdsDataBlockPos::operator< (
    const BdsDataBlockPos & b ) const [inline]
```

## 7.6.4 Member Data Documentation

### 7.6.4.1 channel

**BUInt32** Bds::BdsDataBlockPos::channel

### 7.6.4.2 endTime

**BTimeStamp** Bds::BdsDataBlockPos::endTime

### 7.6.4.3 numChannels

**BUInt32** Bds::BdsDataBlockPos::numChannels

### 7.6.4.4 numSamples

**BUInt64** Bds::BdsDataBlockPos::numSamples

### 7.6.4.5 position

**BUInt64** Bds::BdsDataBlockPos::position

### 7.6.4.6 segment

**BUInt32** Bds::BdsDataBlockPos::segment



## 7.6.4.7 startTime

**BTimeStamp** Bds::BdsDataBlockPos::startTime

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

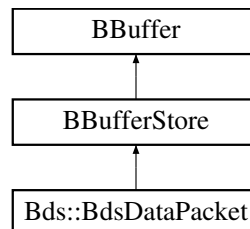
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h

## 7.7 Bds::BdsDataPacket Class Reference

BdsDataFileBds: internal file storage packet.

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

Inheritance diagram for Bds::BdsDataPacket:



### Public Member Functions

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

### Additional Inherited Members

#### 7.7.1 Detailed Description

BdsDataFileBds: internal file storage packet.

#### 7.7.2 Constructor & Destructor Documentation



### 7.7.2.1 BdsDataPacket()

```
Bds::BdsDataPacket::BdsDataPacket ( )
```

### 7.7.2.2 ~BdsDataPacket()

```
Bds::BdsDataPacket::~~BdsDataPacket ( )
```

## 7.7.3 Member Function Documentation

### 7.7.3.1 clear()

```
void Bds::BdsDataPacket::clear ( )
```

### 7.7.3.2 dump()

```
void Bds::BdsDataPacket::dump ( )
```

### 7.7.3.3 getHeader()

```
BEError Bds::BdsDataPacket::getHeader (
    BdsDataPacketHeader & header )
```

### 7.7.3.4 reset()

```
void Bds::BdsDataPacket::reset ( )
```

### 7.7.3.5 setChecksumAndLength()

```
void Bds::BdsDataPacket::setChecksumAndLength ( )
```



## 7.7.3.6 setHeader()

```
BError Bds::BdsDataPacket::setHeader (
    const BdsDataPacketHeader & header )
```

## 7.7.3.7 validateChecksum()

```
BError Bds::BdsDataPacket::validateChecksum ( )
```

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](#)

## 7.8 Bds::BdsDataPacketHeader Struct Reference

BdsDataFileBds internal file storage packet header.

```
#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.*

### 7.8.1 Detailed Description

BdsDataFileBds internal file storage packet header.

### 7.8.2 Member Data Documentation



#### 7.8.2.1 checksum

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

Checksum of packet.

#### 7.8.2.2 endTime

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

The time of the last sample + 1.

#### 7.8.2.3 length

**BUInt32** Bds::BdsDataPacketHeader::length

Length in bytes of packet.

#### 7.8.2.4 sequence

**BUInt32** Bds::BdsDataPacketHeader::sequence

The streamlet packet sequence number.

#### 7.8.2.5 startTime

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

The time of the first sample.

#### 7.8.2.6 streamlet

**BUInt32** Bds::BdsDataPacketHeader::streamlet

The streamlet id.



### 7.8.2.7 type

**BUInt32** Bds::BdsDataPacketHeader::type

Packets type.

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

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

## 7.9 Bds::BdsDataSegment Class Reference

BdsDataFileBds: internal file storage data segment.

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

### Public Member Functions

- [BdsDataSegment](#) ()
- [operator<](#) (const [BdsDataSegment](#) &b) const

### Public Attributes

- **BTimeStamp** [startTime](#)
- **BTimeStamp** [endTime](#)
- **BUInt32** [numBlocks](#)
- **BUInt32** [numSamples](#)
- double [sampleRate](#)
- **BArray**< [BdsDataBlockPos](#) > [blocks](#)

### 7.9.1 Detailed Description

BdsDataFileBds: internal file storage data segment.

### 7.9.2 Constructor & Destructor Documentation

#### 7.9.2.1 BdsDataSegment()

```
Bds::BdsDataSegment::BdsDataSegment ( ) [inline]
```

### 7.9.3 Member Function Documentation



### 7.9.3.1 operator<()

```
int Bds::BdsDataSegment::operator< (
    const BdsDataSegment & b ) const [inline]
```

## 7.9.4 Member Data Documentation

### 7.9.4.1 blocks

```
BArray<BdsDataBlockPos> Bds::BdsDataSegment::blocks
```

### 7.9.4.2 endTime

```
BTimeStamp Bds::BdsDataSegment::endTime
```

### 7.9.4.3 numBlocks

```
BUInt32 Bds::BdsDataSegment::numBlocks
```

### 7.9.4.4 numSamples

```
BUInt32 Bds::BdsDataSegment::numSamples
```

### 7.9.4.5 sampleRate

```
double Bds::BdsDataSegment::sampleRate
```

### 7.9.4.6 startTime

```
BTimeStamp Bds::BdsDataSegment::startTime
```

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

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



## 7.10 Bds::BdsDataStreamlet Class Reference

BdsDataFileBds: internal file storage data streamlet.

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

### Public Member Functions

- [BdsDataStreamlet](#) ()

### Public Attributes

- **BUInt32** [packetNumber](#)
- **BUInt64** [position](#)
- **BUInt32** [channel](#)
- **BUInt32** [numChannels](#)
- **BArray**< [BdsDataBlockPos](#) > [blocks](#)
- **BArray**< [BdsDataSegment](#) > [segments](#)

### 7.10.1 Detailed Description

BdsDataFileBds: internal file storage data streamlet.

### 7.10.2 Constructor & Destructor Documentation

#### 7.10.2.1 BdsDataStreamlet()

```
Bds::BdsDataStreamlet::BdsDataStreamlet ( ) [inline]
```

### 7.10.3 Member Data Documentation

#### 7.10.3.1 blocks

```
BArray<BdsDataBlockPos> Bds::BdsDataStreamlet::blocks
```



### 7.10.3.2 channel

**BUInt32** Bds::BdsDataStreamlet::channel

### 7.10.3.3 numChannels

**BUInt32** Bds::BdsDataStreamlet::numChannels

### 7.10.3.4 packetNumber

**BUInt32** Bds::BdsDataStreamlet::packetNumber

### 7.10.3.5 position

**BUInt64** Bds::BdsDataStreamlet::position

### 7.10.3.6 segments

**BArray**<[BdsDataSegment](#)> Bds::BdsDataStreamlet::segments

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

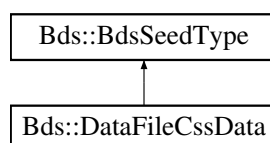
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h](#)

## 7.11 Bds::BdsSeedType Class Reference

BdsDataFileSeed internal parent for all SEED types.

```
#include <BdsSeedType.h>
```

Inheritance diagram for Bds::BdsSeedType:





## Public Member Functions

- [BdsSeedType](#) ()
- **BError** [getInt](#) (char \*\* **data**, int size, int &v)
- **BError** [getUInt](#) (char \*\* **data**, int size, unsigned int &v)
- **BError** [getDouble](#) (char \*\* **data**, int size, double &v)
- **BError** [getString](#) (char \*\* **data**, int size, **BString** &v)
- **BError** [getStringVariable](#) (char \*\* **data**, int size, **BString** &v)
- **BError** [appendInt](#) ( **BString** &s, int v, int size)
- **BError** [appendDouble](#) ( **BString** &s, double v, int size, int precision)
- **BError** [appendExp](#) ( **BString** &s, double v, int size, int precision, int sign)
- **BError** [appendString](#) ( **BString** &s, **BString** v, int size)
- **BError** [appendStringVariable](#) ( **BString** &s, **BString** v, int size)

### 7.11.1 Detailed Description

BdsDataFileSeed internal parent for all SEED types.

### 7.11.2 Constructor & Destructor Documentation

#### 7.11.2.1 BdsSeedType()

```
Bds::BdsSeedType::BdsSeedType ( )
```

### 7.11.3 Member Function Documentation

#### 7.11.3.1 appendDouble()

```
BError Bds::BdsSeedType::appendDouble (
    BString & s,
    double v,
    int size,
    int precision )
```

#### 7.11.3.2 appendExp()

```
BError Bds::BdsSeedType::appendExp (
    BString & s,
    double v,
    int size,
    int precision,
    int sign )
```



#### 7.11.3.3 appendInt()

```
BError Bds::BdsSeedType::appendInt (
    BString & s,
    int v,
    int size )
```

#### 7.11.3.4 appendString()

```
BError Bds::BdsSeedType::appendString (
    BString & s,
    BString v,
    int size )
```

#### 7.11.3.5 appendStringVariable()

```
BError Bds::BdsSeedType::appendStringVariable (
    BString & s,
    BString v,
    int size )
```

#### 7.11.3.6 getDouble()

```
BError Bds::BdsSeedType::getDouble (
    char ** data,
    int size,
    double & v )
```

#### 7.11.3.7 getInt()

```
BError Bds::BdsSeedType::getInt (
    char ** data,
    int size,
    int & v )
```

#### 7.11.3.8 getString()

```
BError Bds::BdsSeedType::getString (
    char ** data,
    int size,
    BString & v )
```



## 7.11.3.9 getStringVariable()

```
BError Bds::BdsSeedType::getStringVariable (
    char ** data,
    int size,
    BString & v )
```

## 7.11.3.10 getUInt()

```
BError Bds::BdsSeedType::getUInt (
    char ** data,
    int size,
    unsigned int & v )
```

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

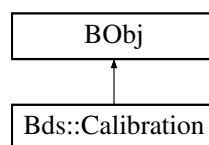
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.h
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp

## 7.12 Bds::Calibration Class Reference

This class defines a calibration setting.

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

Inheritance diagram for Bds::Calibration:



### Public Member Functions

- [Calibration](#) ( BUInt32 id=0, BTimeStamp startTime= BTimeStamp(), BTimeStamp endTime= BTimeStamp(), BString network= BString(), BString station= BString(), BString channel= BString(), BString source= BString(), BString name= BString(), BFloat64 samplingFrequency=0, BFloat64 calibrationFrequency=0, BFloat64 calibrationFactor=0, BString calibrationUnits= BString(), BFloat64 depth=0, BFloat64 horizontalAngle=0, BFloat64 verticalAngle=0)
- [BString](#) [getType](#) ()
- [BError](#) [setMembers](#) ( BDictString &members)
- [BError](#) [setMember](#) ( BString name, BString value)
- [BError](#) [getMembers](#) ( BDictString &members)
- [BError](#) [getMember](#) ( BString name, BString &value)



## 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.*
- **BString** [source](#)  
*The source.*
- **BString** [name](#)  
*The Calibrations name, "Main", "Measured".*
- **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 data to the units. This is a measured value at the calibration frequency.*
- **BString** [calibrationUnits](#)  
*The measurment units.*
- **BFloat64** [depth](#)  
*The depth of the sensor below ground level in meters.*
- **BFloat64** [horizontalAngle](#)  
*The Sensors channel placement horizontal angle in degrees clockwise from north.*
- **BFloat64** [verticalAngle](#)  
*The Sensors channel placement vertical angle in degrees degrees with zero = vertically up.*

### 7.12.1 Detailed Description

This class defines a calibration setting.

Each channel has a `samplingFrequency` and a `calibrationFactor` (scaling factor) associated with it at a particular `calibrationFrequency`. There may be additional calibration information such as the depth of the sensor and its positional angles.

### 7.12.2 Constructor & Destructor Documentation



## 7.12.2.1 Calibration()

```

Bds::Calibration::Calibration (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString network = BString(),
    BString station = BString(),
    BString channel = BString(),
    BString source = BString(),
    BString name = BString(),
    BFloat64 samplingFrequency = 0,
    BFloat64 calibrationFrequency = 0,
    BFloat64 calibrationFactor = 0,
    BString calibrationUnits = BString(),
    BFloat64 depth = 0,
    BFloat64 horizontalAngle = 0,
    BFloat64 verticalAngle = 0 )

```

## 7.12.3 Member Function Documentation

## 7.12.3.1 getMember()

```

BError Bds::Calibration::getMember (
    BString name,
    BString & value ) [virtual]

```

Reimplemented from **BObj**.

## 7.12.3.2 getMembers()

```

BError Bds::Calibration::getMembers (
    BDictString & members ) [virtual]

```

Reimplemented from **BObj**.

## 7.12.3.3 getType()

```

BString Bds::Calibration::getType ( )

```



#### 7.12.3.4 setMember()

```
BError Bds::Calibration::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.12.3.5 setMembers()

```
BError Bds::Calibration::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.12.4 Member Data Documentation

#### 7.12.4.1 calibrationFactor

```
BFloat64 Bds::Calibration::calibrationFactor
```

The scaling value to apply to the data to normalise data to the units. This is a measured value at the calibration frequency.

#### 7.12.4.2 calibrationFrequency

```
BFloat64 Bds::Calibration::calibrationFrequency
```

The frequency that the CalibrationFactor value is valid for.

#### 7.12.4.3 calibrationUnits

```
BString Bds::Calibration::calibrationUnits
```

The measurment units.



#### 7.12.4.4 channel

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

The channel.

#### 7.12.4.5 depth

**BFloat64** Bds::Calibration::depth

The depth of the sensor below ground level in meters.

#### 7.12.4.6 endTime

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

The End Time.

#### 7.12.4.7 horizontalAngle

**BFloat64** Bds::Calibration::horizontalAngle

The Sensors channel placement horizontal angle in degrees clockwise from north.

#### 7.12.4.8 id

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

The ID.

#### 7.12.4.9 name

**BString** Bds::Calibration::name

The Calibrations name, "Main", "Measured".



#### 7.12.4.10 network

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

The Network/Organisation Name.

#### 7.12.4.11 samplingFrequency

**BFloat64** Bds::Calibration::samplingFrequency

The sample rate used.

#### 7.12.4.12 source

**BString** Bds::Calibration::source

The source.

#### 7.12.4.13 startTime

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

The Start Time.

#### 7.12.4.14 station

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

The station.

#### 7.12.4.15 verticalAngle

**BFloat64** Bds::Calibration::verticalAngle

The Sensors channel 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](#)



## 7.13 Bds::CdChannel\_1v0 Struct Reference

BdsDataFile: Internal CD1.0 channel information.

```
#include <BdsDataFileCd.h>
```

### Public Attributes

- **BUInt8** [auth](#)
- **BUInt8** [compress](#)
- **BUInt8** [spare0](#)
- **BUInt8** [spare1](#)
- **BFloat32** [calibrationFactor](#)
- **BFloat32** [calibrationPeriod](#)
- char [name](#) [16]
- **BUInt32** [channel](#)

### 7.13.1 Detailed Description

BdsDataFile: Internal CD1.0 channel information.

### 7.13.2 Member Data Documentation

#### 7.13.2.1 [auth](#)

**BUInt8** Bds::CdChannel\_1v0::auth

#### 7.13.2.2 [calibrationFactor](#)

**BFloat32** Bds::CdChannel\_1v0::calibrationFactor

#### 7.13.2.3 [calibrationPeriod](#)

**BFloat32** Bds::CdChannel\_1v0::calibrationPeriod



#### 7.13.2.4 channel

```
BUInt32 Bds::CdChannel_1v0::channel
```

#### 7.13.2.5 compress

```
BUInt8 Bds::CdChannel_1v0::compress
```

#### 7.13.2.6 name

```
char Bds::CdChannel_1v0::name[16]
```

#### 7.13.2.7 spare0

```
BUInt8 Bds::CdChannel_1v0::spare0
```

#### 7.13.2.8 spare1

```
BUInt8 Bds::CdChannel_1v0::spare1
```

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

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

## 7.14 Bds::CdDataChannel Class Reference

BdsDataFile: Internal CD channel information.

```
#include <BdsDataFileCd.h>
```

### Public Attributes

- **BString** [station](#)
- **BString** [channel](#)
- char [mode](#) [24]
- char [status](#) [32]
- **BTimeStamp** [startTime](#)
- **BUInt32** [period](#)
- **BUInt32** [numSamples](#)
- **BUInt32** [dataSize](#)
- **BUInt8** \* [data](#)



### 7.14.1 Detailed Description

BdsDataFile: Internal CD channel information.

### 7.14.2 Member Data Documentation

#### 7.14.2.1 channel

**BString** Bds::CdDataChannel::channel

#### 7.14.2.2 data

**BUInt8\*** Bds::CdDataChannel::data

#### 7.14.2.3 dataSize

**BUInt32** Bds::CdDataChannel::dataSize

#### 7.14.2.4 mode

char Bds::CdDataChannel::mode[24]

#### 7.14.2.5 numSamples

**BUInt32** Bds::CdDataChannel::numSamples

#### 7.14.2.6 period

**BUInt32** Bds::CdDataChannel::period



#### 7.14.2.7 startTime

**BTimeStamp** Bds::CdDataChannel::startTime

#### 7.14.2.8 station

**BString** Bds::CdDataChannel::station

#### 7.14.2.9 status

char Bds::CdDataChannel::status[32]

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

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

### 7.15 Bds::CdDataFormatFrame\_1v0 Struct Reference

BdsDataFile: Internal CD1.0 frame information.

```
#include <BdsDataFileCd.h>
```

#### Public Attributes

- **BUInt32** [frameType](#)
- **BUInt32** [frameLength](#)
- **BUInt32** [maxFrameLength](#)
- **BUInt32** [numChannels](#)
- **BUInt32** [period](#)
- [CdChannel\\_1v0](#) [channels](#) [100]

#### 7.15.1 Detailed Description

BdsDataFile: Internal CD1.0 frame information.

#### 7.15.2 Member Data Documentation



### 7.15.2.1 channels

`CdChannel_1v0` Bds::CdDataFormatFrame\_1v0::channels[100]

### 7.15.2.2 frameLength

**BUInt32** Bds::CdDataFormatFrame\_1v0::frameLength

### 7.15.2.3 frameType

**BUInt32** Bds::CdDataFormatFrame\_1v0::frameType

### 7.15.2.4 maxFrameLength

**BUInt32** Bds::CdDataFormatFrame\_1v0::maxFrameLength

### 7.15.2.5 numChannels

**BUInt32** Bds::CdDataFormatFrame\_1v0::numChannels

### 7.15.2.6 period

**BUInt32** Bds::CdDataFormatFrame\_1v0::period

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

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

## 7.16 Bds::CdFlag Class Reference

BdsDataFile: Internal CD flag.

```
#include <BdsDataFileCd.h>
```



## Public Member Functions

- [CdFlag](#) ()

## Public Attributes

- int [dead](#)
- int [zeroed](#)

### 7.16.1 Detailed Description

BdsDataFile: Internal CD flag.

### 7.16.2 Constructor & Destructor Documentation

#### 7.16.2.1 CdFlag()

```
Bds::CdFlag::CdFlag ( ) [inline]
```

### 7.16.3 Member Data Documentation

#### 7.16.3.1 dead

```
int Bds::CdFlag::dead
```

#### 7.16.3.2 zeroed

```
int Bds::CdFlag::zeroed
```

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

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

## 7.17 Bds::CdPacketData Class Reference

BdsDataFile: Internal CD data packet.

```
#include <BdsDataFileCd.h>
```



## Public Attributes

- **BUInt32** [frameType](#)
- **BUInt32** [trailerOffset](#)
- char [creator](#) [8]
- char [destination](#) [8]
- **BUInt64** [sequenceNum](#)
- **BUInt32** [series](#)
- **BUInt32** [numChannels](#)
- **BUInt32** [period](#)
- **BTimeStamp** [startTime](#)
- **BArray**< [CdDataChannel](#) > [channels](#)
- **BUInt32** [authKey](#)
- **BUInt32** [authSize](#)
- char \* [auth](#)
- **BUInt64** [crc](#)

### 7.17.1 Detailed Description

BdsDataFile: Internal CD data packet.

### 7.17.2 Member Data Documentation

#### 7.17.2.1 [auth](#)

```
char* Bds::CdPacketData::auth
```

#### 7.17.2.2 [authKey](#)

```
BUInt32 Bds::CdPacketData::authKey
```

#### 7.17.2.3 [authSize](#)

```
BUInt32 Bds::CdPacketData::authSize
```

#### 7.17.2.4 [channels](#)

```
BArray<CdDataChannel> Bds::CdPacketData::channels
```



#### 7.17.2.5 crc

**BUInt64** Bds::CdPacketData::crc

#### 7.17.2.6 creator

char Bds::CdPacketData::creator[8]

#### 7.17.2.7 destination

char Bds::CdPacketData::destination[8]

#### 7.17.2.8 frameType

**BUInt32** Bds::CdPacketData::frameType

#### 7.17.2.9 numChannels

**BUInt32** Bds::CdPacketData::numChannels

#### 7.17.2.10 period

**BUInt32** Bds::CdPacketData::period

#### 7.17.2.11 sequenceNum

**BUInt64** Bds::CdPacketData::sequenceNum

#### 7.17.2.12 series

**BUInt32** Bds::CdPacketData::series



## 7.17.2.13 startTime

**BTimeStamp** Bds::CdPacketData::startTime

## 7.17.2.14 trailerOffset

**BUInt32** Bds::CdPacketData::trailerOffset

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

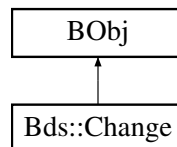
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h

## 7.18 Bds::Change Class Reference

This holds information on a medatdata or sensor data change.

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

Inheritance diagram for Bds::Change:



### Public Member Functions

- **Change** ( **BUInt32** id=0, **BUInt32** changeGroupId=0, **BTimeStamp** time= **BTimeStamp**(), **BString** type= **BString**(), **BString** table= **BString**(), **BUInt32** rowId=0)
- **BString** getType ()
- **BError** setMembers ( **BDictString** &members)
- **BError** setMember ( **BString** name, **BString** value)
- **BError** getMembers ( **BDictString** &members)
- **BError** getMember ( **BString** name, **BString** &value)

### 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.*



### 7.18.1 Detailed Description

This holds information on a metadata or sensor data change.

Whenever a change is made to the BDS metadata or data a [Change](#) object is added to the BDS Changes database. This describes which database table and object that was added or modified. [Change](#)'s are normally grouped together by a [ChangeGroup](#).

### 7.18.2 Constructor & Destructor Documentation

#### 7.18.2.1 Change()

```
Bds::Change::Change (
    BUInt32 id = 0,
    BUInt32 changeGroupId = 0,
    BTimeStamp time = BTimeStamp(),
    BString type = BString(),
    BString table = BString(),
    BUInt32 rowId = 0 )
```

### 7.18.3 Member Function Documentation

#### 7.18.3.1 getMember()

```
BError Bds::Change::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from [BObj](#).

#### 7.18.3.2 getMembers()

```
BError Bds::Change::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from [BObj](#).



### 7.18.3.3 getType()

```
BString Bds::Change::getType ( )
```

### 7.18.3.4 setMember()

```
BError Bds::Change::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

### 7.18.3.5 setMembers()

```
BError Bds::Change::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

## 7.18.4 Member Data Documentation

### 7.18.4.1 changeGroupId

```
BUInt32 Bds::Change::changeGroupId
```

The [Change](#) group ID.

### 7.18.4.2 id

```
BUInt32 Bds::Change::id
```

The unique id.

### 7.18.4.3 rowId

```
BUInt32 Bds::Change::rowId
```

The database row affected.



#### 7.18.4.4 table

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

The database table affected.

#### 7.18.4.5 time

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

The Time the change was made.

#### 7.18.4.6 type

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

The change type.

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

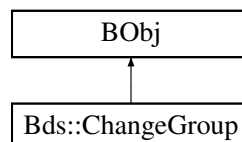
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.19 Bds::ChangeGroup Class Reference

This holds information on a set of Changes.

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

Inheritance diagram for Bds::ChangeGroup:



### Public Member Functions

- [ChangeGroup](#) ( **BUInt32** id=0, **BTimeStamp** time= **BTimeStamp**(), **BString** type= **BString**(), **BString** user= **BString**(), **BString** title= **BString**(), **BString** description= **BString**())
- **BString** [getType](#) ()
- **BError** [setMembers](#) ( **BDictString** &members)
- **BError** [setMember](#) ( **BString** name, **BString** value)
- **BError** [getMembers](#) ( **BDictString** &members)
- **BError** [getMember](#) ( **BString** name, **BString** &value)



## 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.*

### 7.19.1 Detailed Description

This holds information on a set of Changes.

A set of changes to the BDS database are grouped into a [ChangeGroup](#). This could be a set of changes whilst a user is logged in or by a program making a set of changes.

### 7.19.2 Constructor & Destructor Documentation

#### 7.19.2.1 ChangeGroup()

```
Bds::ChangeGroup::ChangeGroup (
    BUInt32 id = 0,
    BTimeStamp time = BTimeStamp(),
    BString type = BString(),
    BString user = BString(),
    BString title = BString(),
    BString description = BString() )
```

### 7.19.3 Member Function Documentation

#### 7.19.3.1 getMember()

```
BError Bds::ChangeGroup::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.



#### 7.19.3.2 getMembers()

```
BError Bds::ChangeGroup::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

#### 7.19.3.3 getType()

```
BString Bds::ChangeGroup::getType ( )
```

#### 7.19.3.4 setMember()

```
BError Bds::ChangeGroup::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.19.3.5 setMembers()

```
BError Bds::ChangeGroup::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.19.4 Member Data Documentation

#### 7.19.4.1 description

```
BString Bds::ChangeGroup::description
```

The Description of the change.



#### 7.19.4.2 id

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

The unique id.

#### 7.19.4.3 time

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

The Time the change was made.

#### 7.19.4.4 title

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

The Changes title.

#### 7.19.4.5 type

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

The type of change.

#### 7.19.4.6 user

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

The user who made the change.

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

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

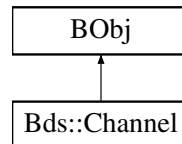


## 7.20 Bds::Channel Class Reference

This class defines a seismic data [Channel](#).

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

Inheritance diagram for Bds::Channel:



### Public Member Functions

- [Channel](#) ( **BUInt32** [id](#)=0, **BTimeStamp** [startTime](#)= **BTimeStamp**(), **BTimeStamp** [endTime](#)= **BTimeStamp**(), **BString** [network](#)= **BString**(), **BString** [station](#)= **BString**(), **BString** [channel](#)= **BString**(), **BString** [channelType](#)= **BString**(), **BString** [channelAux](#)= **BString**(), **BString** [dataType](#)= **BString**(), **BString** [description](#)= **BString**())
- **BString** [getType](#) ()
- **BError** [setMembers](#) ( **BDictString** &members)
- **BError** [setMember](#) ( **BString** name, **BString** value)
- **BError** [getMembers](#) ( **BDictString** &members)
- **BError** [getMember](#) ( **BString** name, **BString** &value)

### Public Attributes

- **BUInt32** [id](#)  
*Unique ID when stored in a database or for other uses.*
- **BTimeStamp** [startTime](#)  
*The Start Time.*
- **BTimeStamp** [endTime](#)  
*The End Time the channel was available.*
- **BString** [network](#)  
*The [Network](#) Name.*
- **BString** [station](#)  
*The Stations name.*
- **BString** [channel](#)  
*The channels name (often as <channelType>\_<>channelAux)*
- **BString** [channelType](#)  
*The channels type (component of station field)*
- **BString** [channelAux](#)  
*The channels auxiliary identifier (component of station field)*
- **BString** [dataType](#)  
*The Type of data (seismic, seismicUnknown, data, log, unknown, empty)*
- **BString** [description](#)  
*The channels description.*



### 7.20.1 Detailed Description

This class defines a seismic data [Channel](#).

This class defines a seismic data channel with network:station:channel definitions. The class also splits the channel's name field into channelType and channelAux (channel name is <channelType>\_<channelAux> to] allow easy database searches etc. As well as seismic data a channel can contain other data types.

### 7.20.2 Constructor & Destructor Documentation

#### 7.20.2.1 Channel()

```
Bds::Channel::Channel (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString network = BString(),
    BString station = BString(),
    BString channel = BString(),
    BString channelType = BString(),
    BString channelAux = BString(),
    BString dataType = BString(),
    BString description = BString() )
```

### 7.20.3 Member Function Documentation

#### 7.20.3.1 getMember()

```
BError Bds::Channel::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.20.3.2 getMembers()

```
BError Bds::Channel::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.



### 7.20.3.3 getType()

```
BString Bds::Channel::getType ( )
```

### 7.20.3.4 setMember()

```
BError Bds::Channel::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

### 7.20.3.5 setMembers()

```
BError Bds::Channel::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

## 7.20.4 Member Data Documentation

### 7.20.4.1 channel

```
BString Bds::Channel::channel
```

The channels name (often as <channelType>\_<>channelAux)

### 7.20.4.2 channelAux

```
BString Bds::Channel::channelAux
```

The channels auxiliary identifier (component of station field)

### 7.20.4.3 channelType

```
BString Bds::Channel::channelType
```

The channels type (component of station field)



#### 7.20.4.4 dataType

**BString** Bds::Channel::dataType

The Type of data (seismic, seismicUnknown, data, log, unknown, empty)

#### 7.20.4.5 description

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

The channels description.

#### 7.20.4.6 endTime

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

The End Time the channel was available.

#### 7.20.4.7 id

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

Unique ID when stored in a database or for other uses.

#### 7.20.4.8 network

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

The [Network](#) Name.

#### 7.20.4.9 startTime

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

The Start Time.



#### 7.20.4.10 station

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

The Stations name.

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

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

## 7.21 Bds::ChannelInfo Class Reference

This class provides information on a channel.

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

### Public Member Functions

- [ChannelInfo](#) ( [BTimeStamp](#) startTime= [BTimeStamp](#)(), [BTimeStamp](#) endTime= [BTimeStamp](#)(), [Station](#) station=[Station](#)(), [Location](#) location=[Location](#)(), [Channel](#) channel=[Channel](#)(), [BString](#) source= [BString](#)(), [BString](#) dataType= [BString](#)(), [Digitiser](#) digitiser=[Digitiser](#)(), [Sensor](#) sensor=[Sensor](#)(), [Calibration](#) calibration=[Calibration](#)(), [BList](#)< [Response](#) > responses= [BList](#)< [Response](#) >())

### Public Attributes

- [BTimeStamp](#) [startTime](#)  
*The Start Time.*
- [BTimeStamp](#) [endTime](#)  
*The End Time.*
- [Station](#) [station](#)  
*The [Station](#) info.*
- [Location](#) [location](#)  
*The [Station](#) location.*
- [Channel](#) [channel](#)  
*The [Channel](#) data.*
- [BString](#) [source](#)  
*The data source.*
- [BString](#) [dataType](#)  
*The [DataType](#) (seismic, seismicUnknown, data, log, unknown, empty)*
- [Digitiser](#) [digitiser](#)  
*The [Digitiser](#) in use.*
- [Sensor](#) [sensor](#)  
*The [Sensor](#) in use.*
- [Calibration](#) [calibration](#)  
*The [Calibration](#) info.*
- [BList](#)< [Response](#) > [responses](#)  
*The list of frequency responses.*



### 7.21.1 Detailed Description

This class provides information on a channel.

This returns the metadata available for a channel over a particular time period. There are likely to be multiple [ChannelInfo](#) objects over larger time periods, one for each change in metadata. The [ChannelInfos](#) object contains an array of these [ChannelInfo](#) objects.

### 7.21.2 Constructor & Destructor Documentation

#### 7.21.2.1 ChannelInfo()

```
Bds::ChannelInfo::ChannelInfo (
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    Station station = Station(),
    Location location = Location(),
    Channel channel = Channel(),
    BString source = BString(),
    BString dataType = BString(),
    Digitiser digitiser = Digitiser(),
    Sensor sensor = Sensor(),
    Calibration calibration = Calibration(),
    BList< Response > responses = BList<Response >() )
```

### 7.21.3 Member Data Documentation

#### 7.21.3.1 calibration

[Calibration](#) Bds::ChannelInfo::calibration

The [Calibration](#) info.

#### 7.21.3.2 channel

[Channel](#) Bds::ChannelInfo::channel

The [Channel](#) data.



### 7.21.3.3 dataType

**BString** Bds::ChannelInfo::dataType

The DataType (seismic, seismicUnknown, data, log, unknown, empty)

### 7.21.3.4 digitiser

[Digitiser](#) Bds::ChannelInfo::digitiser

The [Digitiser](#) in use.

### 7.21.3.5 endTime

**BTimeStamp** Bds::ChannelInfo::endTime

The End Time.

### 7.21.3.6 location

[Location](#) Bds::ChannelInfo::location

The [Station](#) location.

### 7.21.3.7 responses

**BList**<[Response](#) > Bds::ChannelInfo::responses

The list of frequency responses.

### 7.21.3.8 sensor

[Sensor](#) Bds::ChannelInfo::sensor

The [Sensor](#) in use.



## 7.21.3.9 source

**BString** Bds::ChannelInfo::source

The data source.

## 7.21.3.10 startTime

**BTimeStamp** Bds::ChannelInfo::startTime

The Start Time.

## 7.21.3.11 station

**Station** Bds::ChannelInfo::station

The [Station](#) info.

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

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

## 7.22 Bds::ChannelInfos Class Reference

This class provides metadata information on a set of channels.

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

### Public Member Functions

- [ChannelInfos](#) ( **BArray**< **BArray**< [ChannelInfo](#) > > channels= **BArray**< **BArray**< [ChannelInfo](#) > >())

### Public Attributes

- **BArray**< **BArray**< [ChannelInfo](#) > > channels

### 7.22.1 Detailed Description

This class provides metadata information on a set of channels.

This provides all of the metadata information for a set of channels over a given time period. For each channel there can be one or more [ChannelInfo](#) objects each defining a set of constant metadata over a particular time period. The BdsServer will create a [ChannelInfo](#) object on each change in metadata when the user asks for a set of metadata over some time period.



## 7.22.2 Constructor & Destructor Documentation

### 7.22.2.1 ChannelInfos()

```
Bds::ChannelInfos::ChannelInfos (
    BArray< BArray< ChannelInfo > > channels = BArray< BArray<ChannelInfo > > ()
)
```

## 7.22.3 Member Data Documentation

### 7.22.3.1 channels

```
BArray< BArray<ChannelInfo > > Bds::ChannelInfos::channels
```

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

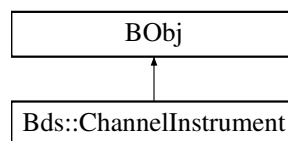
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.23 Bds::ChannelInstrument Class Reference

This class defines a [Channel](#)'s instrument.

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

Inheritance diagram for Bds::ChannelInstrument:



### Public Member Functions

- [ChannelInstrument](#) ( [BUInt32](#) id=0, [BTimeStamp](#) startTime= [BTimeStamp](#)(), [BTimeStamp](#) endTime= [BTimeStamp](#)(), [BUInt32](#) channelId=0, [BString](#) source= [BString](#)(), [BUInt32](#) digitiserId=0, [BUInt32](#) sensorId=0)
- [BString](#) [getType](#) ()
- [BError](#) [setMembers](#) ( [BDictString](#) &members)
- [BError](#) [setMember](#) ( [BString](#) name, [BString](#) value)
- [BError](#) [getMembers](#) ( [BDictString](#) &members)
- [BError](#) [getMember](#) ( [BString](#) name, [BString](#) &value)



## Public Attributes

- **BUInt32** [id](#)  
*Unique ID when stored in a database or for other uses.*
- **BTimeStamp** [startTime](#)  
*The Start Time.*
- **BTimeStamp** [endTime](#)  
*The End Time the channel was available.*
- **BUInt32** [channelId](#)  
*The channels Id.*
- **BString** [source](#)  
*The source.*
- **BUInt32** [digitiserId](#)  
*The [Digitiser](#) in use.*
- **BUInt32** [sensorId](#)  
*The sensor in use.*

### 7.23.1 Detailed Description

This class defines a [Channel](#)'s instrument.

It links a seismic data channel with a particular sensor and digitiser. [Note](#) that it is possible to share sensor's and digitisers between channels if wanted for generic sensor/digitiser definitions. However if particular serial numbers are needed the sensor/digitiser needs to be unique.

### 7.23.2 Constructor & Destructor Documentation

#### 7.23.2.1 ChannelInstrument()

```
Bds::ChannelInstrument::ChannelInstrument (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BUInt32 channelId = 0,
    BString source = BString(),
    BUInt32 digitiserId = 0,
    BUInt32 sensorId = 0 )
```

### 7.23.3 Member Function Documentation



#### 7.23.3.1 getMember()

```
BError Bds::ChannelInstrument::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.23.3.2 getMembers()

```
BError Bds::ChannelInstrument::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

#### 7.23.3.3 getType()

```
BString Bds::ChannelInstrument::getType ( )
```

#### 7.23.3.4 setMember()

```
BError Bds::ChannelInstrument::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.23.3.5 setMembers()

```
BError Bds::ChannelInstrument::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.23.4 Member Data Documentation



#### 7.23.4.1 channelId

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

The channels Id.

#### 7.23.4.2 digitiserId

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

The [Digitiser](#) in use.

#### 7.23.4.3 endTime

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

The End Time the channel was available.

#### 7.23.4.4 id

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

Unique ID when stored in a database or for other uses.

#### 7.23.4.5 sensorId

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

The sensor in use.

#### 7.23.4.6 source

**BString** Bds::ChannelInstrument::source

The source.



#### 7.23.4.7 `startTime`

```
BTimeStamp Bds::ChannelInstrument::startTime
```

The Start Time.

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

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

## 7.24 `Bds::ChannelName` Class Reference

This class defines a full channel name.

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

### Public Member Functions

- `ChannelName` ( `BString` `network`= `BString()`, `BString` `station`= `BString()`, `BString` `channel`= `BString()`, `BString` `source`= `BString()`)

### Public Attributes

- `BString` `network`  
*The Channels network.*
- `BString` `station`  
*The Channels station.*
- `BString` `channel`  
*The Channels name.*
- `BString` `source`  
*The Channels source.*

#### 7.24.1 Detailed Description

This class defines a full channel name.

A channel's data is fully defined by the Network:Station:Channel:[Source](#). This class stores all of these component names.

#### 7.24.2 Constructor & Destructor Documentation



### 7.24.2.1 ChannelName()

```
Bds::ChannelName::ChannelName (
    BString network = BString(),
    BString station = BString(),
    BString channel = BString(),
    BString source = BString() )
```

## 7.24.3 Member Data Documentation

### 7.24.3.1 channel

```
BString Bds::ChannelName::channel
```

The Channels name.

### 7.24.3.2 network

```
BString Bds::ChannelName::network
```

The Channels network.

### 7.24.3.3 source

```
BString Bds::ChannelName::source
```

The Channels source.

### 7.24.3.4 station

```
BString Bds::ChannelName::station
```

The Channels station.

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

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



## 7.25 Bds::CleanOptions Class Reference

This defines the set of clean options used in the clean() function.

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

### Public Member Functions

- [CleanOptions](#) ( **BInt32** logs=0, **BInt32** changes=0, **BInt32** deletedFiles=0)

### Public Attributes

- **BInt32** logs  
*Clean the Logs.*
- **BInt32** changes  
*Clean the changes.*
- **BInt32** deletedFiles  
*Clean deleted data files.*

### 7.25.1 Detailed Description

This defines the set of clean options used in the clean() function.

### 7.25.2 Constructor & Destructor Documentation

#### 7.25.2.1 CleanOptions()

```
Bds::CleanOptions::CleanOptions (
    BInt32 logs = 0,
    BInt32 changes = 0,
    BInt32 deletedFiles = 0 )
```

### 7.25.3 Member Data Documentation

#### 7.25.3.1 changes

**BInt32** Bds::CleanOptions::changes

Clean the changes.



## 7.25.3.2 deletedFiles

**BInt32** Bds::CleanOptions::deletedFiles

Clean deleted data files.

## 7.25.3.3 logs

**BInt32** Bds::CleanOptions::logs

Clean the Logs.

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

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

## 7.26 Bds::CompressSteim1 Class Reference

Steim1 un-compress class.

```
#include <BdsCompress.h>
```

### Public Member Functions

- [CompressSteim1](#) ()
- void [setByteOrder](#) (int swap)
- void [clear](#) ()
- **BError** [unCompress](#) (void \*buffer, **BUInt** numSamples, **BArray**< **BInt32** > & data)

### 7.26.1 Detailed Description

Steim1 un-compress class.

### 7.26.2 Constructor & Destructor Documentation

#### 7.26.2.1 CompressSteim1()

```
Bds::CompressSteim1::CompressSteim1 ( )
```



## 7.26.3 Member Function Documentation

### 7.26.3.1 clear()

```
void Bds::CompressSteim1::clear ( )
```

### 7.26.3.2 setByteOrder()

```
void Bds::CompressSteim1::setByteOrder (
    int swap )
```

### 7.26.3.3 unCompress()

```
BError Bds::CompressSteim1::unCompress (
    void * buffer,
    BUInt numSamples,
    BArray< BInt32 > & data )
```

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

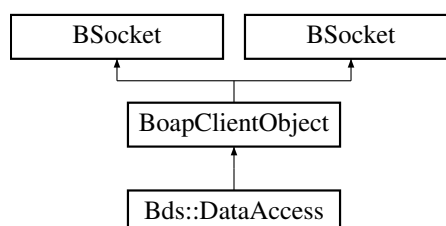
- [/src/blacknest/bds/bds/bdsDataLib/BdsCompress.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp](#)

## 7.27 Bds::DataAccess Class Reference

This is the Data Access API interface.

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

Inheritance diagram for Bds::DataAccess:





## Public Member Functions

- [DataAccess](#) ( **BString** name="" )
- **BError** [connect](#) ( **BString** user, **BString** password )  
*Provides user/password information for secure connection.*
- **BError** [validateUser](#) ( **BString** user, **BString** email )  
*Checks the user given name or email.*
- **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, **BString** &name )  
*Gets the software version and server name.*
- **BError** [userGetFromId](#) ( **BUInt32** id, [User](#) &user )  
*Get user info given user ID.*
- **BError** [userGet](#) ( [User](#) &user )  
*Get user info.*
- **BError** [userSet](#) ( [User](#) user )  
*Set user info.*
- **BError** [userGetGroups](#) ( **BList**< **BString** > &groups )  
*Get list of groups the user belongs to.*
- **BError** [groupGetList](#) ( **BList**< [Group](#) > &groups )  
*Get list of Groups.*
- **BError** [networkGetList](#) ( **BList**< [Network](#) > &networks )  
*Get list of Networks.*
- **BError** [stationGetList](#) ( [Selection](#) sel, **BList**< [Station](#) > &stations )  
*Get list of Stations.*
- **BError** [channelGetList](#) ( [Selection](#) sel, **BList**< [Channel](#) > &channels )  
*Get list of Channels.*
- **BError** [sourceGetList](#) ( **BList**< [Source](#) > &sources )  
*Get list of Sources.*
- **BError** [sourcePriorityGetList](#) ( **BList**< [SourcePriority](#) > &sourcePriorities )  
*Get list of SourcePriorities.*
- **BError** [dataFileGetList](#) ( [Selection](#) sel, **BList**< [DataFileInfo](#) > &dataFile )  
*Get list of DataFiles.*
- **BError** [dataChannelGetList](#) ( [Selection](#) sel, **BList**< [DataChannel](#) > &dataChannel )  
*Get list of DataChannels.*
- **BError** [channellInstrumentGetList](#) ( [Selection](#) sel, **BList**< [ChannellInstrument](#) > &channellInstruments )  
*Get list of Instruments.*
- **BError** [digitiserGetList](#) ( [Selection](#) sel, **BList**< [Digitiser](#) > &digitisers )  
*Get list of Digitisers.*
- **BError** [digitiserGet](#) ( **BUInt32** id, [Digitiser](#) &digitiser )  
*Get a [Digitiser](#) object given its ID.*
- **BError** [sensorGetList](#) ( [Selection](#) sel, **BList**< [Sensor](#) > &sensors )  
*Get list of Sensors.*
- **BError** [sensorGet](#) ( **BUInt32** id, [Sensor](#) &sensor )  
*Get a [Sensor](#) object given its ID.*
- **BError** [calibrationGetList](#) ( [Selection](#) sel, **BList**< [Calibration](#) > &calibrations )  
*Get list of Calibrations.*
- **BError** [responseGetList](#) ( [Selection](#) sel, **BList**< [Response](#) > &responses )  
*Get list of Responses.*



- **Error** `locationGetList` (`Selection` sel, `BList`< `Location` > &locations)  
*Get list of `Station` Locations.*
- **Error** `getSelectionInfo` (`SelectionGroup` group, `Selection` selectionIn, `SelectionInfo` &selectionInfo)  
*Get information on possible selections. Use in GUI programs to list options available.*
- **Error** `getSelections` (`SelectionGroup` group, `Selection` selectionIn, `Selection` &selectionOut)  
*Get selection list.*
- **Error** `dataAvailability` (`Selection` selection, `BUInt32` num, `BArray`< `DataAvailChan` > &dataAvailChans)  
*Return availability for data matching the given selection parameters.*
- **Error** `dataSearch` (`Selection` selection, `DataInfo` &dataInfo)  
*Search for data matching the given selection parameters.*
- **Error** `dataGetChannelInfo` (`DataInfo` dataInfo, `ChannelInfos` &channelInfos)  
*Return the channel MetaData in structured form.*
- **Error** `dataOpen` (`DataInfo` dataInfo, `BString` mode, `BString` format, `BUInt32` flags, `DataHandle` &data↵  
Handle)  
*Open a data file.*
- **Error** `dataGetInfo` (`DataHandle` dataHandle, `BUInt32` infoExtra, `DataInfo` &dataInfo)  
*Get information on the data file.*
- **Error** `dataGetNotes` (`DataHandle` dataHandle, `BList`< `Note` > &notes)  
*Get notes on the data file.*
- **Error** `dataGetWarnings` (`DataHandle` dataHandle, `BList`< `BString` > &warnings)  
*Get information on the data file.*
- **Error** `dataSeekBlock` (`DataHandle` dataHandle, `BUInt32` channel, `BUInt32` segment, `BTimeStamp` time, `BUInt32` &blockNumber)  
*Searches for a data block matching the time given.*
- **Error** `dataGetBlock` (`DataHandle` dataHandle, `BUInt32` channel, `BUInt32` segment, `BUInt32` block↵  
Number, `DataBlock` & data)  
*Return a block of data.*
- **Error** `dataClose` (`DataHandle` dataHandle, `BError` error, `BInt32` del)  
*Close a file.*
- **Error** `dataFormattedRead` (`DataHandle` dataHandle, `BUInt32` number, `BArray`< `BUInt8` > & data)  
*Read the raw data from the file.*
- **Error** `dataFormattedGetLength` (`DataHandle` dataHandle, `BUInt64` & length)  
*Read the raw data from the file.*
- **Error** `noteGetList` (`Selection` sel, `BList`< `Note` > &notes)  
*Return a list of Notes.*
- **Error** `noteUpdate` ( `BInt32` append, `Note` note, `BUInt32` &id)  
*Add or update a `Note`.*
- **Error** `noteWriteDocument` ( `BUInt32` id, `BString` format, `BArray`< `BUInt8` > data)  
*Given a `Note` write a document associated with it.*
- **Error** `noteReadDocument` ( `BUInt32` id, `BString` &format, `BArray`< `BUInt8` > & data)  
*Read a document associated with a `Note`.*
- **Error** `logUpdate` ( `BInt32` append, `Log` log, `BUInt32` &id)  
*Add or update a `Log` item.*
- **Error** `logAppend` ( `BString` type, `BUInt32` priority, `BString` subSystem, `BString` title, `BString` description)  
*Append a log item.*
- **Error** `modeSet` (`Mode` mode, `Mode` &previousMode)  
*Changes the system mode from Master to slave.*
- **Error** `modeSnapshotPause` ( `BInt32` on)  
*Enables/disables backup synchronisation pause.*
- **Error** `clean` (`CleanOptions` cleanOptions)  
*Cleans the system logs and Changes information.*



- **BError** `databaseBackup` ( **BString** &ref)  
*Backup the database.*
- **BError** `statisticsGet` ( **BDict**< **BString** > &info)  
*Get a list of system statistics.*
- **BError** `dataFormatGetList` ( **BList**< **DataFormat** > &formats)  
*Get list of data formats.*

## Additional Inherited Members

### 7.27.1 Detailed Description

This is the Data Access API interface.

This object provides the set of API functions that make RPC network calls to a BdsServer. It provides the basic and restricted user orientated read only, data access API.

### 7.27.2 Constructor & Destructor Documentation

#### 7.27.2.1 DataAccess()

```
Bds::DataAccess::DataAccess (
    BString name = "" )
```

### 7.27.3 Member Function Documentation

#### 7.27.3.1 calibrationGetList()

```
BError Bds::DataAccess::calibrationGetList (
    Selection sel,
    BList< Calibration > & calibrations )
```

Get list of Calibrations.

#### 7.27.3.2 channelGetList()

```
BError Bds::DataAccess::channelGetList (
    Selection sel,
    BList< Channel > & channels )
```

Get list of Channels.



### 7.27.3.3 channelInstrumentGetList()

```
BError Bds::DataAccess::channelInstrumentGetList (
    Selection sel,
    BList< ChannelInstrument > & channelInstruments )
```

Get list of Instruments.

### 7.27.3.4 clean()

```
BError Bds::DataAccess::clean (
    CleanOptions cleanOptions )
```

Cleans the system logs and Changes information.

### 7.27.3.5 connect()

```
BError Bds::DataAccess::connect (
    BString user,
    BString password )
```

Provides user/password information for secure connection.

### 7.27.3.6 dataAvailability()

```
BError Bds::DataAccess::dataAvailability (
    Selection selection,
    BUInt32 num,
    BArray< DataAvailChan > & dataAvailChans )
```

Return availability for data matching the given selection parameters.

### 7.27.3.7 databaseBackup()

```
BError Bds::DataAccess::databaseBackup (
    BString & ref )
```

Backup the database.



## 7.27.3.8 dataChannelGetList()

```
BError Bds::DataAccess::dataChannelGetList (
    Selection sel,
    BList< DataChannel > & dataChannel )
```

Get list of DataChannels.

## 7.27.3.9 dataClose()

```
BError Bds::DataAccess::dataClose (
    DataHandle dataHandle,
    BError error,
    BInt32 del )
```

Close a file.

## 7.27.3.10 dataFileGetList()

```
BError Bds::DataAccess::dataFileGetList (
    Selection sel,
    BList< DataFileInfo > & dataFile )
```

Get list of DataFiles.

## 7.27.3.11 dataFormatGetList()

```
BError Bds::DataAccess::dataFormatGetList (
    BList< DataFormat > & formats )
```

Get list of data formats.

## 7.27.3.12 dataFormattedGetLength()

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

Read the raw data from the file.



#### 7.27.3.13 dataFormattedRead()

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

Read the raw data from the file.

#### 7.27.3.14 dataGetBlock()

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

Return a block of data.

#### 7.27.3.15 dataGetChannelInfo()

```
BError Bds::DataAccess::dataGetChannelInfo (
    DataInfo dataInfo,
    ChannelInfos & channelInfos )
```

Return the channel MetaData in structured form.

#### 7.27.3.16 dataGetInfo()

```
BError Bds::DataAccess::dataGetInfo (
    DataHandle dataHandle,
    BUInt32 infoExtra,
    DataInfo & dataInfo )
```

Get information on the data file.

#### 7.27.3.17 dataGetNotes()

```
BError Bds::DataAccess::dataGetNotes (
    DataHandle dataHandle,
    BList< Note > & notes )
```

Get notes on the data file.



## 7.27.3.18 dataGetWarnings()

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

Get information on the data file.

## 7.27.3.19 dataOpen()

```
BError Bds::DataAccess::dataOpen (
    DataInfo dataInfo,
    BString mode,
    BString format,
    BUInt32 flags,
    DataHandle & dataHandle )
```

Open a data file.

## Parameters

in	<i>dataInfo</i>	The sensor data to open selection
in	<i>mode</i>	The open format. The mode should be set to "w" for writing data, "a" when appending data and "r" for reading data.
in	<i>format</i>	What format to open the data stream as. This can be API for raw BDS API access or one of the supported formats that the BDS is able to convert to.
in	<i>flags</i>	A bitset of flags as defined by <a href="#">Bds::DataFlags</a> .
out	<i>dataHandle</i>	The handle for the open data set

## 7.27.3.20 dataSearch()

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

Search for data matching the given selection parameters.

## 7.27.3.21 dataSeekBlock()

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

Searches for a data block matching the time given.



#### 7.27.3.22 digitiserGet()

```
BError Bds::DataAccess::digitiserGet (
    BUInt32 id,
    Digitiser & digitiser )
```

Get a [Digitiser](#) object given its ID.

#### 7.27.3.23 digitiserGetList()

```
BError Bds::DataAccess::digitiserGetList (
    Selection sel,
    BList< Digitiser > & digitisers )
```

Get list of Digitisers.

#### 7.27.3.24 getSelectionInfo()

```
BError Bds::DataAccess::getSelectionInfo (
    SelectionGroup group,
    Selection selectionIn,
    SelectionInfo & selectionInfo )
```

Get information on possible selections. Use in GUI programs to list options available.

#### 7.27.3.25 getSelections()

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

Get selection list.

#### 7.27.3.26 getVersion()

```
BError Bds::DataAccess::getVersion (
    BString & version,
    BString & name )
```

Gets the software version and server name.



## 7.27.3.27 groupGetList()

```
BError Bds::DataAccess::groupGetList (
    BList< Group > & groups )
```

Get list of Groups.

## 7.27.3.28 locationGetList()

```
BError Bds::DataAccess::locationGetList (
    Selection sel,
    BList< Location > & locations )
```

Get list of [Station](#) Locations.

## 7.27.3.29 logAppend()

```
BError Bds::DataAccess::logAppend (
    BString type,
    BUInt32 priority,
    BString subSystem,
    BString title,
    BString description )
```

Append a log item.

## 7.27.3.30 logUpdate()

```
BError Bds::DataAccess::logUpdate (
    BInt32 append,
    Log log,
    BUInt32 & id )
```

Add or update a [Log](#) item.

## 7.27.3.31 modeSet()

```
BError Bds::DataAccess::modeSet (
    Mode mode,
    Mode & previousMode )
```

Changes the system mode from Master to slave.



#### 7.27.3.32 modeSnapshotPause()

```
BError Bds::DataAccess::modeSnapshotPause (
    BInt32 on )
```

Enables/disables backup synchronisation pause.

#### 7.27.3.33 networkGetList()

```
BError Bds::DataAccess::networkGetList (
    BList< Network > & networks )
```

Get list of Networks.

#### 7.27.3.34 noteGetList()

```
BError Bds::DataAccess::noteGetList (
    Selection sel,
    BList< Note > & notes )
```

Return a list of Notes.

#### 7.27.3.35 noteReadDocument()

```
BError Bds::DataAccess::noteReadDocument (
    BUInt32 id,
    BString & format,
    BArray< BUInt8 > & data )
```

Read a document associated with a [Note](#).

#### 7.27.3.36 noteUpdate()

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

Add or update a [Note](#).



## 7.27.3.37 noteWriteDocument()

```
BError Bds::DataAccess::noteWriteDocument (
    BUInt32 id,
    BString format,
    BArray< BUInt8 > data )
```

Given a [Note](#) write a document associated with it.

## 7.27.3.38 responseGetList()

```
BError Bds::DataAccess::responseGetList (
    Selection sel,
    BList< Response > & responses )
```

Get list of Responses.

## 7.27.3.39 sensorGet()

```
BError Bds::DataAccess::sensorGet (
    BUInt32 id,
    Sensor & sensor )
```

Get a [Sensor](#) object given its ID.

## 7.27.3.40 sensorGetList()

```
BError Bds::DataAccess::sensorGetList (
    Selection sel,
    BList< Sensor > & sensors )
```

Get list of Sensors.

## 7.27.3.41 setUser()

```
BError Bds::DataAccess::setUser (
    BString user,
    BString email )
```

Sets user to given name or email.



#### 7.27.3.42 setUserReal()

```
BError Bds::DataAccess::setUserReal ( )
```

Sets user back to real user.

#### 7.27.3.43 sourceGetList()

```
BError Bds::DataAccess::sourceGetList (
    BList< Source > & sources )
```

Get list of Sources.

#### 7.27.3.44 sourcePriorityGetList()

```
BError Bds::DataAccess::sourcePriorityGetList (
    BList< SourcePriority > & sourcePrioritys )
```

Get list of SourcePriorities.

#### 7.27.3.45 stationGetList()

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

Get list of Stations.

#### 7.27.3.46 statisticsGet()

```
BError Bds::DataAccess::statisticsGet (
    BDict< BString > & info )
```

Get a list of system statistics.

#### 7.27.3.47 userGet()

```
BError Bds::DataAccess::userGet (
    User & user )
```

Get user info.



## 7.27.3.48 userGetFromId()

```
BError Bds::DataAccess::userGetFromId (
    BUInt32 id,
    User & user )
```

Get user info given user ID.

## 7.27.3.49 userGetGroups()

```
BError Bds::DataAccess::userGetGroups (
    BList< BString > & groups )
```

Get list of groups the user belongs to.

## 7.27.3.50 userSet()

```
BError Bds::DataAccess::userSet (
    User user )
```

Set user info.

## 7.27.3.51 validateUser()

```
BError Bds::DataAccess::validateUser (
    BString user,
    BString email )
```

Checks the user given name or email.

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

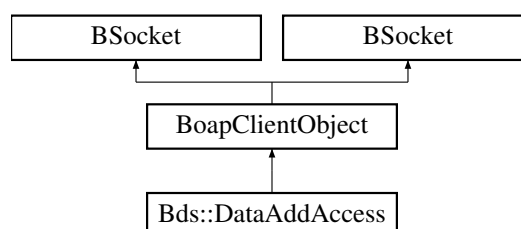
- [BdsC.h](#)
- [BdsC.cc](#)
- [BdsLib.dox](#)

## 7.28 Bds::DataAddAccess Class Reference

This is the DataAdd Access API interface.

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

Inheritance diagram for Bds::DataAddAccess:





## Public Member Functions

- [DataAddAccess](#) ( **BString** name="" )
- **BError** [connect](#) ( **BString** user, **BString** password )  
*Provides user/password information.*
- **BError** [validateUser](#) ( **BString** user, **BString** email )  
*Checks the user given name or email.*
- **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, **BString** &name )  
*Gets the software version and server name.*
- **BError** [userGetFromId](#) ( **BUInt32** id, [User](#) &user )  
*Get user info given user ID.*
- **BError** [userGet](#) ( [User](#) &user )  
*Get user info.*
- **BError** [userSet](#) ( [User](#) user )  
*Set user info.*
- **BError** [userGetGroups](#) ( **BList**< **BString** > &groups )  
*Get list of groups the user belongs to.*
- **BError** [groupGetList](#) ( **BList**< [Group](#) > &groups )  
*Get list of Groups.*
- **BError** [networkGetList](#) ( **BList**< [Network](#) > &networks )  
*Get list of Networks.*
- **BError** [stationGetList](#) ( [Selection](#) sel, **BList**< [Station](#) > &stations )  
*Get list of Stations.*
- **BError** [channelGetList](#) ( [Selection](#) sel, **BList**< [Channel](#) > &channels )  
*Get list of Channels.*
- **BError** [sourceGetList](#) ( **BList**< [Source](#) > &sources )  
*Get list of Sources.*
- **BError** [sourcePriorityGetList](#) ( **BList**< [SourcePriority](#) > &sourcePriorities )  
*Get list of SourcePriorities.*
- **BError** [dataFileGetList](#) ( [Selection](#) sel, **BList**< [DataFileInfo](#) > &dataFile )  
*Get list of DataFiles.*
- **BError** [dataChannelGetList](#) ( [Selection](#) sel, **BList**< [DataChannel](#) > &dataChannel )  
*Get list of DataChannels.*
- **BError** [channellInstrumentGetList](#) ( [Selection](#) sel, **BList**< [ChannellInstrument](#) > &channellInstruments )  
*Get list of Instruments.*
- **BError** [digitiserGetList](#) ( [Selection](#) sel, **BList**< [Digitiser](#) > &digitisers )  
*Get list of Channels.*
- **BError** [digitiserGet](#) ( **BUInt32** id, [Digitiser](#) &digitiser )  
*Get a [Digitiser](#) object given its ID.*
- **BError** [sensorGetList](#) ( [Selection](#) sel, **BList**< [Sensor](#) > &sensors )  
*Get list of Sensors.*
- **BError** [sensorGet](#) ( **BUInt32** id, [Sensor](#) &sensor )  
*Get a [Sensor](#) object given its ID.*
- **BError** [calibrationGetList](#) ( [Selection](#) sel, **BList**< [Calibration](#) > &calibrations )  
*Get list of Calibrations.*
- **BError** [responseGetList](#) ( [Selection](#) sel, **BList**< [Response](#) > &responses )  
*Get list of Responses.*



- **BError** `locationGetList` ([Selection](#) sel, **BList**< [Location](#) > &locations)  
*Get list of [Station](#) Locations.*
- **BError** `getSelectionInfo` ([SelectionGroup](#) group, [Selection](#) selectionIn, [SelectionInfo](#) &selectionInfo)  
*Get information on possible selections. Use in GUI programs to list options available.*
- **BError** `getSelections` ([SelectionGroup](#) group, [Selection](#) selectionIn, [Selection](#) &selectionOut)  
*Get selection list.*
- **BError** `dataAvailability` ([Selection](#) selection, **BUInt32** num, **BArray**< [DataAvailChan](#) > &dataAvailChans)  
*Return availability for data matching the given selection parameters.*
- **BError** `dataSearch` ([Selection](#) selection, [DataInfo](#) &dataInfo)  
*Search for data matching the given selection parameters.*
- **BError** `dataGetChannelInfo` ([DataInfo](#) dataInfo, [ChannelInfos](#) &channelInfos)  
*Return the channel MetaData in structured form.*
- **BError** `dataOpen` ([DataInfo](#) dataInfo, **BString** mode, **BString** format, **BUInt32** flags, [DataHandle](#) &data↵  
Handle)  
*Open a data file.*
- **BError** `dataGetInfo` ([DataHandle](#) dataHandle, **BUInt32** infoExtra, [DataInfo](#) &dataInfo)  
*Get information on the data file.*
- **BError** `dataGetNotes` ([DataHandle](#) dataHandle, **BList**< [Note](#) > &notes)  
*Get notes on the data file.*
- **BError** `dataGetWarnings` ([DataHandle](#) dataHandle, **BList**< **BString** > &warnings)  
*Get information on the data file.*
- **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** `dataSetInfo` ([DataHandle](#) dataHandle, [DataInfo](#) dataInfo)  
*Set the info when writing to a file.*
- **BError** `dataPutBlock` ([DataHandle](#) dataHandle, [DataBlock](#) data)  
*Send a block of data.*
- **BError** `dataClose` ([DataHandle](#) dataHandle, **BError** error, **BInt32** del)  
*Close a file.*
- **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** `noteGetList` ([Selection](#) sel, **BList**< [Note](#) > &notes)  
*Return a list of Notes.*
- **BError** `noteUpdate` ( **BInt32** append, [Note](#) note, **BUInt32** &id)  
*Add or update a [Note](#).*
- **BError** `noteWriteDocument` ( **BUInt32** id, **BString** format, **BArray**< **BUInt8** > data)  
*Given a [Note](#) write a document associated with it.*
- **BError** `noteReadDocument` ( **BUInt32** id, **BString** &format, **BArray**< **BUInt8** > & data)  
*Read a document associated with a [Note](#).*
- **BError** `logUpdate` ( **BInt32** append, [Log](#) log, **BUInt32** &id)  
*Append a log item ///< Add or update a [Log](#) item.*
- **BError** `logAppend` ( **BString** type, **BUInt32** priority, **BString** subSystem, **BString** title, **BString** description)  
*Append a log item.*
- **BError** `modeSet` ([Mode](#) mode, [Mode](#) &previousMode)  
*Changes the system mode from Master to slave.*



- **BError** `modeSnapshotPause` ( **BInt32** on)  
*Enables/disables backup synchronisation pause.*
- **BError** `clean` ( **CleanOptions** cleanOptions)  
*Cleans the system logs and Changes information.*
- **BError** `databaseBackup` ( **BString** &ref)  
*Backup the database.*
- **BError** `statisticsGet` ( **BDict**< **BString** > &info)  
*Get a list of system statistics.*
- **BError** `dataFormatGetList` ( **BList**< **DataFormat** > &formats)  
*Get list of data formats.*

## Additional Inherited Members

### 7.28.1 Detailed Description

This is the DataAdd Access API interface.

This object provides the set of API functions that make RPC network calls to a BdsServer. It provides the basic and restricted user orientated read only, data access API along with the ability to import data. It will normally be used by data import client programs.

### 7.28.2 Constructor & Destructor Documentation

#### 7.28.2.1 DataAddAccess()

```
Bds::DataAddAccess::DataAddAccess (
    BString name = "" )
```

### 7.28.3 Member Function Documentation

#### 7.28.3.1 calibrationGetList()

```
BError Bds::DataAddAccess::calibrationGetList (
    Selection sel,
    BList< Calibration > & calibrations )
```

Get list of Calibrations.



### 7.28.3.2 channelGetList()

```
BError Bds::DataAddAccess::channelGetList (
    Selection sel,
    BList< Channel > & channels )
```

Get list of Channels.

### 7.28.3.3 channelInstrumentGetList()

```
BError Bds::DataAddAccess::channelInstrumentGetList (
    Selection sel,
    BList< ChannelInstrument > & channelInstruments )
```

Get list of Instruments.

### 7.28.3.4 clean()

```
BError Bds::DataAddAccess::clean (
    CleanOptions cleanOptions )
```

Cleans the system logs and Changes information.

### 7.28.3.5 connect()

```
BError Bds::DataAddAccess::connect (
    BString user,
    BString password )
```

Provides user/password information.

### 7.28.3.6 dataAvailability()

```
BError Bds::DataAddAccess::dataAvailability (
    Selection selection,
    BUInt32 num,
    BArray< DataAvailChan > & dataAvailChans )
```

Return availability for data matching the given selection parameters.



#### 7.28.3.7 databaseBackup()

```
BError Bds::DataAddAccess::databaseBackup (
    BString & ref )
```

Backup the database.

#### 7.28.3.8 dataChannelGetList()

```
BError Bds::DataAddAccess::dataChannelGetList (
    Selection sel,
    BList< DataChannel > & dataChannel )
```

Get list of DataChannels.

#### 7.28.3.9 dataClose()

```
BError Bds::DataAddAccess::dataClose (
    DataHandle dataHandle,
    BError error,
    BInt32 del )
```

Close a file.

#### 7.28.3.10 dataFileGetList()

```
BError Bds::DataAddAccess::dataFileGetList (
    Selection sel,
    BList< DataFileInfo > & dataFile )
```

Get list of DataFiles.

#### 7.28.3.11 dataFormatGetList()

```
BError Bds::DataAddAccess::dataFormatGetList (
    BList< DataFormat > & formats )
```

Get list of data formats.



#### 7.28.3.12 dataFormattedGetLength()

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

The total length in bytes of the formatted data.

#### 7.28.3.13 dataFormattedRead()

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

Read the raw data from the stream.

#### 7.28.3.14 dataGetBlock()

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

Return a block of data.

#### 7.28.3.15 dataGetChannelInfo()

```
BError Bds::DataAddAccess::dataGetChannelInfo (
    DataInfo dataInfo,
    ChannelInfos & channelInfos )
```

Return the channel MetaData in structured form.

#### 7.28.3.16 dataGetInfo()

```
BError Bds::DataAddAccess::dataGetInfo (
    DataHandle dataHandle,
    BUInt32 infoExtra,
    DataInfo & dataInfo )
```

Get information on the data file.



## 7.28.3.17 dataGetNotes()

```
BError Bds::DataAddAccess::dataGetNotes (
    DataHandle dataHandle,
    BList< Note > & notes )
```

Get notes on the data file.

## 7.28.3.18 dataGetWarnings()

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

Get information on the data file.

## 7.28.3.19 dataOpen()

```
BError Bds::DataAddAccess::dataOpen (
    DataInfo dataInfo,
    BString mode,
    BString format,
    BUInt32 flags,
    DataHandle & dataHandle )
```

Open a data file.

## Parameters

in	<i>dataInfo</i>	The sensor data to open selection
in	<i>mode</i>	The open format. The mode should be set to "w" for writing data, "a" when appending data and "r" for reading data.
in	<i>format</i>	What format to open the data stream as. This can be API for raw BDS API access or one of the supported formats that the BDS is able to convert to.
in	<i>flags</i>	A bitset of flags as defined by <a href="#">Bds::DataFlags</a> .
out	<i>dataHandle</i>	The handle for the open data set

## 7.28.3.20 dataPutBlock()

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

Send a block of data.



## 7.28.3.21 dataSearch()

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

Search for data matching the given selection parameters.

## 7.28.3.22 dataSeekBlock()

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

Searches for a data block matching the time given.

## 7.28.3.23 dataSetInfo()

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

Set the info when writing to a file.

## 7.28.3.24 digitiserGet()

```
BError Bds::DataAddAccess::digitiserGet (
    BUInt32 id,
    Digitiser & digitiser )
```

Get a [Digitiser](#) object given its ID.

## 7.28.3.25 digitiserGetList()

```
BError Bds::DataAddAccess::digitiserGetList (
    Selection sel,
    BList< Digitiser > & digitisers )
```

Get list of Channels.



#### 7.28.3.26 getSelectionInfo()

```
BError Bds::DataAddAccess::getSelectionInfo (
    SelectionGroup group,
    Selection selectionIn,
    SelectionInfo & selectionInfo )
```

Get information on possible selections. Use in GUI programs to list options available.

#### 7.28.3.27 getSelections()

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

Get selection list.

#### 7.28.3.28 getVersion()

```
BError Bds::DataAddAccess::getVersion (
    BString & version,
    BString & name )
```

Gets the software version and server name.

#### 7.28.3.29 groupGetList()

```
BError Bds::DataAddAccess::groupGetList (
    BList< Group > & groups )
```

Get list of Groups.

#### 7.28.3.30 locationGetList()

```
BError Bds::DataAddAccess::locationGetList (
    Selection sel,
    BList< Location > & locations )
```

Get list of [Station](#) Locations.



## 7.28.3.31 logAppend()

```
BError Bds::DataAddAccess::logAppend (
    BString type,
    BUInt32 priority,
    BString subSystem,
    BString title,
    BString description )
```

Append a log item.

## 7.28.3.32 logUpdate()

```
BError Bds::DataAddAccess::logUpdate (
    BInt32 append,
    Log log,
    BUInt32 & id )
```

Append a log item ///< Add or update a [Log](#) item.

## 7.28.3.33 modeSet()

```
BError Bds::DataAddAccess::modeSet (
    Mode mode,
    Mode & previousMode )
```

Changes the system mode from Master to slave.

## 7.28.3.34 modeSnapshotPause()

```
BError Bds::DataAddAccess::modeSnapshotPause (
    BInt32 on )
```

Enables/disables backup synchronisation pause.

## 7.28.3.35 networkGetList()

```
BError Bds::DataAddAccess::networkGetList (
    BList< Network > & networks )
```

Get list of Networks.



#### 7.28.3.36 noteGetList()

```
BError Bds::DataAddAccess::noteGetList (
    Selection sel,
    BList< Note > & notes )
```

Return a list of Notes.

#### 7.28.3.37 noteReadDocument()

```
BError Bds::DataAddAccess::noteReadDocument (
    BUInt32 id,
    BString & format,
    BArray< BUInt8 > & data )
```

Read a document associated with a [Note](#).

#### 7.28.3.38 noteUpdate()

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

Add or update a [Note](#).

#### 7.28.3.39 noteWriteDocument()

```
BError Bds::DataAddAccess::noteWriteDocument (
    BUInt32 id,
    BString format,
    BArray< BUInt8 > data )
```

Given a [Note](#) write a document associated with it.

#### 7.28.3.40 responseGetList()

```
BError Bds::DataAddAccess::responseGetList (
    Selection sel,
    BList< Response > & responses )
```

Get list of Responses.



## 7.28.3.41 sensorGet()

```
BError Bds::DataAddAccess::sensorGet (
    BUInt32 id,
    Sensor & sensor )
```

Get a [Sensor](#) object given its ID.

## 7.28.3.42 sensorGetList()

```
BError Bds::DataAddAccess::sensorGetList (
    Selection sel,
    BList< Sensor > & sensors )
```

Get list of Sensors.

## 7.28.3.43 setUser()

```
BError Bds::DataAddAccess::setUser (
    BString user,
    BString email )
```

Sets user to given name or email.

## 7.28.3.44 setUserReal()

```
BError Bds::DataAddAccess::setUserReal ( )
```

Sets user back to real user.

## 7.28.3.45 sourceGetList()

```
BError Bds::DataAddAccess::sourceGetList (
    BList< Source > & sources )
```

Get list of Sources.



#### 7.28.3.46 sourcePriorityGetList()

```
BError Bds::DataAddAccess::sourcePriorityGetList (
    BList< SourcePriority > & sourcePrioritys )
```

Get list of SourcePriorities.

#### 7.28.3.47 stationGetList()

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

Get list of Stations.

#### 7.28.3.48 statisticsGet()

```
BError Bds::DataAddAccess::statisticsGet (
    BDict< BString > & info )
```

Get a list of system statistics.

#### 7.28.3.49 userGet()

```
BError Bds::DataAddAccess::userGet (
    User & user )
```

Get user info.

#### 7.28.3.50 userGetFromId()

```
BError Bds::DataAddAccess::userGetFromId (
    BUInt32 id,
    User & user )
```

Get user info given user ID.



## 7.28.3.51 userGetGroups()

```
BError Bds::DataAddAccess::userGetGroups (
    BList< BString > & groups )
```

Get list of groups the user belongs to.

## 7.28.3.52 userSet()

```
BError Bds::DataAddAccess::userSet (
    User user )
```

Set user info.

## 7.28.3.53 validateUser()

```
BError Bds::DataAddAccess::validateUser (
    BString user,
    BString email )
```

Checks the user given name or email.

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

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

## 7.29 Bds::DataAvail Class Reference

This class provides availability information on a particular period of data.

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

### Public Member Functions

- [DataAvail](#) ( **BTimeStamp** [startTime](#)= **BTimeStamp**(), **BTimeStamp** [endTime](#)= **BTimeStamp**(), [AvailType](#) [availType](#)=[AvailType](#)() )

### Public Attributes

- **BTimeStamp** [startTime](#)  
*The Start Time.*
- **BTimeStamp** [endTime](#)  
*The End Time.*
- [AvailType](#) [availType](#)  
*The availability type. Can be: AvailNone, AvailPartial or AvailFull.*



### 7.29.1 Detailed Description

This class provides availability information on a particular period of data.

### 7.29.2 Constructor & Destructor Documentation

#### 7.29.2.1 DataAvail()

```
Bds::DataAvail::DataAvail (
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    AvailType availType = AvailType() )
```

### 7.29.3 Member Data Documentation

#### 7.29.3.1 availType

```
AvailType Bds::DataAvail::availType
```

The availability type. Can be: AvailNone, AvailPartial or AvailFull.

#### 7.29.3.2 endTime

```
BTimeStamp Bds::DataAvail::endTime
```

The End Time.

#### 7.29.3.3 startTime

```
BTimeStamp Bds::DataAvail::startTime
```

The Start Time.

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

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



## 7.30 Bds::DataAvailChan Class Reference

This class defines availability information on a set of data.

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

### Public Member Functions

- **DataAvailChan** ( **BTimeStamp** *startTime*= **BTimeStamp**(), **BTimeStamp** *endTime*= **BTimeStamp**(), **BString** *network*= **BString**(), **BString** *station*= **BString**(), **BString** *channel*= **BString**(), **BString** *source*= **BString**(), **BArray**< **DataAvail** > *segments*= **BArray**< **DataAvail** >())

### Public Attributes

- **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.*
- **BArray**< **DataAvail** > *segments*  
*Segment info.*

#### 7.30.1 Detailed Description

This class defines availability information on a set of data.

Availability information is provided for a particular channel network:station:channel:source. For a particular time period this will be broken down into time period segments. The time granularity of the segments is dependent of the type of data availability search. Normally the granularity is that defined by the [DataChannel](#) information in the database. The actual seismic data files may not contains some data over the given periods due to missing blocks etc. An in-depth adta availability search could provide more fuller data availability information but with much more data processing. This has yet to be implemented.

#### 7.30.2 Constructor & Destructor Documentation



### 7.30.2.1 DataAvailChan()

```
Bds::DataAvailChan::DataAvailChan (
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString network = BString(),
    BString station = BString(),
    BString channel = BString(),
    BString source = BString(),
    BArray< DataAvail > segments = BArray<DataAvail >() )
```

## 7.30.3 Member Data Documentation

### 7.30.3.1 channel

```
BString Bds::DataAvailChan::channel
```

The Channels name.

### 7.30.3.2 endTime

```
BTimeStamp Bds::DataAvailChan::endTime
```

The End Time.

### 7.30.3.3 network

```
BString Bds::DataAvailChan::network
```

The [Network](#) Name.

### 7.30.3.4 segments

```
BArray<DataAvail > Bds::DataAvailChan::segments
```

Segment info.



## 7.30.3.5 source

**BString** Bds::DataAvailChan::source

The Data [Source](#).

## 7.30.3.6 startTime

**BTimeStamp** Bds::DataAvailChan::startTime

The Start Time.

## 7.30.3.7 station

**BString** Bds::DataAvailChan::station

The [Station](#) name.

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

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

## 7.31 Bds::DataBlock Class Reference

This class provides the actual seismic data values contained within a single data block.

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

### Public Member Functions

- [DataBlock](#) ( **BTimeStamp** [startTime](#)= **BTimeStamp**(), **BTimeStamp** [endTime](#)= **BTimeStamp**(), **BUInt32** [channelNumber](#)=0, **BUInt32** [segmentNumber](#)=0, **BArray**< **BArray**< **BFloat64** > > [channelData](#)= **BArray**< **BArray**< **BFloat64** > >(), **BDict**< **BString** > [info](#)= **BDict**< **BString** >())

### Public Attributes

- **BTimeStamp** [startTime](#)  
*The Start Time.*
- **BTimeStamp** [endTime](#)  
*The End Time the channel was available.*
- **BUInt32** [channelNumber](#)  
*The first channel number. (1, 2, 3 ...)*
- **BUInt32** [segmentNumber](#)  
*The segment number. (1, 2, 3, ...)*
- **BArray**< **BArray**< **BFloat64** > > [channelData](#)  
*The raw channel data in a 2 dimensional array, ordered as per channel information in dataInfo.*
- **BDict**< **BString** > [info](#)  
*Extra information on data or ASCII data.*



### 7.31.1 Detailed Description

This class provides the actual seismic data values contained within a single data block.

All seismic data is segmented into blocks that have a timestamp and perhaps other metadata. The [DataBlock](#) contains an array of actual data samples in a 64bit floating pointer format for one or more channels. If the samples are for multiple channels it is assumed that these are synchronously sampled and are normally from a data file that is in the sample multiplexed format (ie. For each point in time there is a set of samples one for each channel). The `startTime` comes from the original blocks start time information. The `endTime` may come from the original blocks end time information, if available in the original data format that the data was imported from or is generated from the `startTime` and the calculated sample rate of the data. For some data types where the sampling rate is a bit variable, the `endTime` fields may be lined up with the next blocks `startTime` field to ensure contiguous data segments. The `info` field contains extra, free string format, metadata on the block if available. This could be quality information from the TapeDigitiser system for example.

### 7.31.2 Constructor & Destructor Documentation

#### 7.31.2.1 DataBlock()

```
Bds::DataBlock::DataBlock (
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BUInt32 channelNumber = 0,
    BUInt32 segmentNumber = 0,
    BArray< BArray< BFloat64 > > channelData = BArray< BArray< BFloat64 > >(),
    BDict< BString > info = BDict< BString >() )
```

### 7.31.3 Member Data Documentation

#### 7.31.3.1 channelData

```
BArray< BArray< BFloat64 > > Bds::DataBlock::channelData
```

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

#### 7.31.3.2 channelNumber

```
BUInt32 Bds::DataBlock::channelNumber
```

The first channel number. (1, 2, 3 ...)



## 7.31.3.3 endTime

**BTimeStamp** Bds::DataBlock::endTime

The End Time the channel was available.

## 7.31.3.4 info

**BDict< BString >** Bds::DataBlock::info

Extra information on data or ASCII data.

## 7.31.3.5 segmentNumber

**BUInt32** Bds::DataBlock::segmentNumber

The segment number. (1, 2, 3, ...)

## 7.31.3.6 startTime

**BTimeStamp** Bds::DataBlock::startTime

The Start Time.

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

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

## 7.32 Bds::DataBlockPos Class Reference

This defines the position of a data block in a file. It is used by the BDS data convertors to order blocks by time.

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

### Public Member Functions

- [DataBlockPos](#) ( **BTimeStamp** startTime=0, **BTimeStamp** endTime=0, **BUInt64** position=0, **BUInt** order=0, int ref=0, **BUInt** numSamples=0)
- int [operator<](#) (const [DataBlockPos](#) &b) const



## Public Attributes

- **BTimeStamp** [startTime](#)
- **BTimeStamp** [endTime](#)
- **BUInt64** [position](#)
- **BUInt** [order](#)
- **int** [ref](#)
- **BUInt** [numSamples](#)

### 7.32.1 Detailed Description

This defines the position of a data block in a file. It is used by the BDS data convertors to order blocks by time.

### 7.32.2 Constructor & Destructor Documentation

#### 7.32.2.1 DataBlockPos()

```
Bds::DataBlockPos::DataBlockPos (
    BTimeStamp startTime = 0,
    BTimeStamp endTime = 0,
    BUInt64 position = 0,
    BUInt order = 0,
    int ref = 0,
    BUInt numSamples = 0 ) [inline]
```

### 7.32.3 Member Function Documentation

#### 7.32.3.1 operator<()

```
int Bds::DataBlockPos::operator< (
    const DataBlockPos & b ) const [inline]
```

### 7.32.4 Member Data Documentation

#### 7.32.4.1 endTime

```
BTimeStamp Bds::DataBlockPos::endTime
```



#### 7.32.4.2 numSamples

**BUInt** Bds::DataBlockPos::numSamples

#### 7.32.4.3 order

**BUInt** Bds::DataBlockPos::order

#### 7.32.4.4 position

**BUInt64** Bds::DataBlockPos::position

#### 7.32.4.5 ref

**int** Bds::DataBlockPos::ref

#### 7.32.4.6 startTime

**BTimeStamp** Bds::DataBlockPos::startTime

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

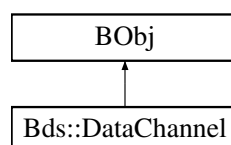
- /src/blacknest/bds/bds/bdsDataLib/[BdsDataFile.h](#)

## 7.33 Bds::DataChannel Class Reference

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

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

Inheritance diagram for Bds::DataChannel:





## Public Member Functions

- **DataChannel** ( **BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), **BString** network= **BString**(), **BString** station= **BString**(), **BString** channel= **BString**(), **BString** source= **BString**(), **BUInt32** numBlocks=0, **BUInt64** numSamples=0, **BFloat64** sampleRate=0, **BUInt32** sampleFormat=0, **BUInt32** dataFileId=0, **BUInt32** dataFileChannel=0, **BString** importFormat= **BString**(), **BString** importFilename= **BString**(), **BTimeStamp** importStartTime= **BTimeStamp**(), **BDict**< **BString** > info= **BDict**< **BString** >())
- **BString** getType ()
- **BError** setMembers ( **BDictString** &members)
- **BError** setMember ( **BString** name, **BString** value)
- **BError** getMembers ( **BDictString** &members)
- **BError** getMember ( **BString** name, **BString** &value)

## Public Attributes

- **BUInt32** id  
*Unique ID when stored in a database or for other uses.*
- **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. This links to the particular [DataFileInfo](#) where the data is stored.*
- **BUInt32** dataFileChannel  
*The Data File Channel number. The channel number within the data file. (1, 2, 3 ...)*
- **BString** importFormat  
*The original data format.*
- **BString** importFilename  
*The original data file name.*
- **BTimeStamp** importStartTime  
*The original import files start time.*
- **BDict**< **BString** > info  
*Extra info on the channel.*



### 7.33.1 Detailed Description

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

This provides information on actual seismic data for a channel that is stored in the BDS system. The data will be stored in a particular file perhaps with other data channels. When known information on the channels numBlocks, numSamples and sampleRate will be provided. Generally this information will only be know if a data files has been imported rather than a live real-time data stream. Generally the seismic data file itself should be interrogated to find the definitive information. The info field provides extra details on the data contents which might come from one of the specfic data import formats.

### 7.33.2 Constructor & Destructor Documentation

#### 7.33.2.1 DataChannel()

```
Bds::DataChannel::DataChannel (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString network = BString(),
    BString station = BString(),
    BString channel = BString(),
    BString source = BString(),
    BUInt32 numBlocks = 0,
    BUInt64 numSamples = 0,
    BFloat64 sampleRate = 0,
    BUInt32 sampleFormat = 0,
    BUInt32 dataFileId = 0,
    BUInt32 dataFileChannel = 0,
    BString importFormat = BString(),
    BString importFilename = BString(),
    BTimeStamp importStartTime = BTimeStamp(),
    BDict< BString > info = BDict< BString >() )
```

### 7.33.3 Member Function Documentation

#### 7.33.3.1 getMember()

```
BError Bds::DataChannel::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.



#### 7.33.3.2 getMembers()

```
BError Bds::DataChannel::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

#### 7.33.3.3 getType()

```
BString Bds::DataChannel::getType ( )
```

#### 7.33.3.4 setMember()

```
BError Bds::DataChannel::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.33.3.5 setMembers()

```
BError Bds::DataChannel::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.33.4 Member Data Documentation

#### 7.33.4.1 channel

```
BString Bds::DataChannel::channel
```

The Channels name.



#### 7.33.4.2 dataFileChannel

**BUInt32** Bds::DataChannel::dataFileChannel

The Data File [Channel](#) number. The channel number within the data file. (1, 2, 3 ...)

#### 7.33.4.3 dataFileId

**BUInt32** Bds::DataChannel::dataFileId

The Data File Id. This links to the particular [DataFileInfo](#) where the data is stored.

#### 7.33.4.4 endTime

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

The End Time.

#### 7.33.4.5 id

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

Unique ID when stored in a database or for other uses.

#### 7.33.4.6 importFilename

**BString** Bds::DataChannel::importFilename

The original data file name.

#### 7.33.4.7 importFormat

**BString** Bds::DataChannel::importFormat

The original data format.



#### 7.33.4.8 importStartTime

**BTimeStamp** Bds::DataChannel::importStartTime

The original import files start time.

#### 7.33.4.9 info

**BDict< BString >** Bds::DataChannel::info

Extra info on the channel.

#### 7.33.4.10 network

**BString** Bds::DataChannel::network

The [Network](#) Name.

#### 7.33.4.11 numBlocks

**BUInt32** Bds::DataChannel::numBlocks

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

#### 7.33.4.12 numSamples

**BUInt64** Bds::DataChannel::numSamples

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

#### 7.33.4.13 sampleFormat

**BUInt32** Bds::DataChannel::sampleFormat

The data sample format.



## 7.33.4.14 sampleRate

**BFloat64** Bds::DataChannel::sampleRate

The data's sample rate.

## 7.33.4.15 source

**BString** Bds::DataChannel::source

The Data [Source](#).

## 7.33.4.16 startTime

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

The Start Time.

## 7.33.4.17 station

**BString** Bds::DataChannel::station

The [Station](#) name.

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

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

## 7.34 Bds::DataCollate Class Reference

Not sure if this is used or what it does.

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

### Public Member Functions

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



### 7.34.1 Detailed Description

Not sure if this is used or what it does.

### 7.34.2 Constructor & Destructor Documentation

#### 7.34.2.1 DataCollate()

```
Bds::DataCollate::DataCollate ( )
```

#### 7.34.2.2 ~DataCollate()

```
Bds::DataCollate::~~DataCollate ( )
```

### 7.34.3 Member Function Documentation

#### 7.34.3.1 addSource()

```
BError Bds::DataCollate::addSource (
    DataFile & dataFile,
    BUInt channel )
```

#### 7.34.3.2 readData()

```
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](#)



## 7.35 Bds::DataError Class Reference

This stores a data error. It includes an error number and a string as well as information on what seismic channel it is for.

```
#include <BdsLib.h>
```

### Public Member Functions

- [DataError](#) ()
- [DataError](#) (int errorNumber, **BString** title, **BString** filename, **BTimeStamp** startTime, **BTimeStamp** endTime, [DataInfo](#) &dataInfo, **BUInt** channel, **BString** description, **BString** user="")
- [DataError](#) & [set](#) (int errorNumber, **BString** title, **BString** importFilename, **BTimeStamp** startTime, **BTimeStamp** endTime, [DataInfo](#) &dataInfo, **BUInt** channel, **BString** description, **BString** user="")
- void [mergeDataInfo](#) (const [DataInfo](#) &dataInfo, **BUInt** channel)
- int [getErrorNumber](#) () const  
*Get The error number.*
- **BString** [getTitle](#) () const  
*Get the title.*
- **BError** [setString](#) (**BString** str)  
*Set from string.*
- **BError** [setStringUser](#) (**BString** str, **BString** user)  
*Set from string given by user on command line.*
- **BString** [getString](#) () const  
*Get error message.*
- int [num](#) () const  
*Get The error number.*
- const char \* [str](#) () const  
*Return a char\* string.*
- [operator int](#) () const  
*Return error number.*

### Public Attributes

- **Blnt32** [oerrorNumber](#)  
*Error number.*
- **BString** [otitle](#)  
*The title.*
- **BString** [odescription](#)  
*The description.*
- **BString** [ofilename](#)  
*The import filename;.*
- **BTimeStamp** [ostartTime](#)  
*The start Time.*
- **BTimeStamp** [oendTime](#)  
*The end Time.*
- **BString** [onetwork](#)  
*The network Name.*
- **BString** [ostation](#)  
*The station/array name.*



- **BString** [ochannel](#)  
*The channel name.*
- **BString** [osource](#)  
*The data [Source](#).*
- **BString** [ouser](#)  
*The user.*

### 7.35.1 Detailed Description

This stores a data error. It includes an error number and a string as well as information on what seismic channel it is for.

### 7.35.2 Constructor & Destructor Documentation

#### 7.35.2.1 `DataError()` [1/2]

```
Bds::DataError::DataError ( )
```

#### 7.35.2.2 `DataError()` [2/2]

```
Bds::DataError::DataError (
    int errorNumber,
    BString title,
    BString filename,
    BTimeStamp startTime,
    BTimeStamp endTime,
    DataInfo & dataInfo,
    BUInt channel,
    BString description,
    BString user = "" )
```

### 7.35.3 Member Function Documentation

#### 7.35.3.1 `getErrorNumber()`

```
int Bds::DataError::getErrorNumber ( ) const
```

Get The error number.



#### 7.35.3.2 getString()

```
BString Bds::DataError::getString ( ) const
```

Get error message.

#### 7.35.3.3 getTitle()

```
BString Bds::DataError::getTitle ( ) const
```

Get the title.

#### 7.35.3.4 mergeDataInfo()

```
void Bds::DataError::mergeDataInfo (
    const DataInfo & dataInfo,
    BUInt channel )
```

#### 7.35.3.5 num()

```
int Bds::DataError::num ( ) const
```

Get The error number.

#### 7.35.3.6 operator int()

```
Bds::DataError::operator int ( ) const
```

Return error number.

#### 7.35.3.7 set()

```
DataError & Bds::DataError::set (
    int errorNumber,
    BString title,
    BString importFilename,
    BTimeStamp startTime,
    BTimeStamp endTime,
    DataInfo & dataInfo,
    BUInt channel,
    BString description,
    BString user = "" )
```



#### 7.35.3.8 setString()

```
BError Bds::DataError::setString (
    BString str )
```

Set from string.

#### 7.35.3.9 setStringUser()

```
BError Bds::DataError::setStringUser (
    BString str,
    BString user )
```

Set from string given by user on command line.

#### 7.35.3.10 str()

```
const char * Bds::DataError::str ( ) const
```

Return a char\* string.

### 7.35.4 Member Data Documentation

#### 7.35.4.1 ochannel

```
BString Bds::DataError::ochannel
```

The channel name.

#### 7.35.4.2 odescription

```
BString Bds::DataError::odescription
```

The description.



#### 7.35.4.3 oendTime

**BTimeStamp** Bds::DataError::oendTime

The end Time.

#### 7.35.4.4 oerrorNumber

**Int32** Bds::DataError::oerrorNumber

Error number.

#### 7.35.4.5 ofilename

**BString** Bds::DataError::ofilename

The import filename;.

#### 7.35.4.6 onetwork

**BString** Bds::DataError::onetwork

The network Name.

#### 7.35.4.7 osource

**BString** Bds::DataError::osource

The data [Source](#).

#### 7.35.4.8 ostartTime

**BTimeStamp** Bds::DataError::ostartTime

The start Time.



#### 7.35.4.9 ostation

**BString** Bds::DataError::ostation

The station/array name.

#### 7.35.4.10 otitle

**BString** Bds::DataError::otitle

The title.

#### 7.35.4.11 ouser

**BString** Bds::DataError::ouser

The user.

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

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

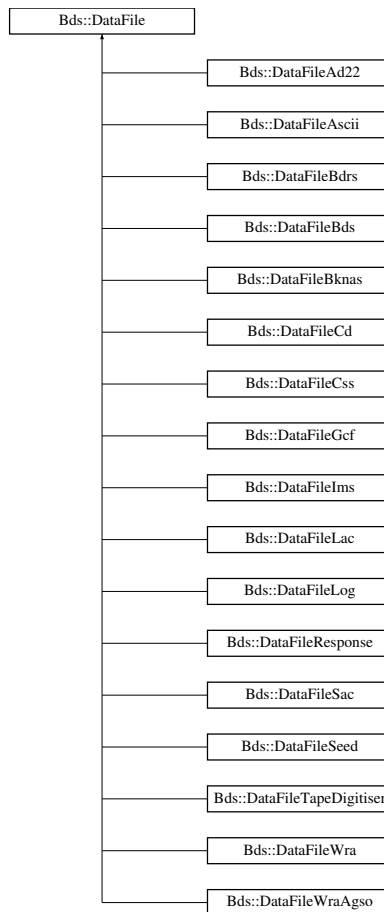
## 7.36 Bds::DataFile Class Reference

This class defines the interface for generic data file access that all of the BDS data conterors share.

```
#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 [WriteOptionsList](#) { [WriteOptionNone](#) = 0x00, [WriteOptionSensorData](#) = 0x01, [WriteOptionNoMetadata](#) = 0x02 }
- enum [ReadOptionsList](#) {  
[ReadOptionNone](#) = 0x00, [ReadOptionValidate](#) = 0x01, [ReadOptionFileNameProcess](#) = 0x02, [ReadOptionValidateCorruptions](#) = 0x04,  
[ReadOptionReorder](#) = 0x08, [ReadOptionDeleteDuplicates](#) = 0x10, [ReadOptionInfoExtra](#) = 0x20,  
[ReadOptionIgnoreSamplerate](#) = 0x40,  
[ReadOptionPrintBlocks](#) = 0x80 }

## 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 **BError** [close](#) ()  
*Close the file.*
- virtual **BError** [setFormat](#) ( **BString** format)



- Set the sub-format.
  - virtual **BString** `getFileName` ()
  - Return the file name.
  - 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` (const **DataInfo** &dataInfo, const **ChannelInfos** &channelInfos, **WriteOptionsList** options=**WriteOptionNone**)
  - Set information on data for write.
  - virtual **BError** `start` ( **BUInt** channel, **BUInt** segment)
  - Start writing next segment of data.
  - virtual **BError** `writeData` (const **DataBlock** & data)
  - Write a block of data.
  - virtual **BError** `end` ()
  - End write segment.
  - virtual **BError** `flush` ()
  - Flush data to disk.
  - virtual **BError** `fileNameProcess` ()
  - Parse the file name for a date/time.
  - virtual **BError** `getFormat` ( **BString** &format)
  - Get sub-format.
  - virtual **BError** `getInfo` (**DataInfo** &dataInfo, **DataFileOptions** options, **BList**< **DataError** > &errors)
  - Get info on data.
  - virtual **BError** `seekBlock` ( **BUInt32** channel, **BUInt** segment, **BTimeStamp** time, **BUInt32** &blockNumber, **BUInt64** &sampleNumber, **DataBlock** & data)
  - Find requested block on given channel given a time.
  - virtual **BError** `readData` ( **BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, **DataBlock** &dataBlock)
  - Read a block.
  - virtual **BError** `getMetaData` (**ChannelInfos** &channelInfos)
  - Return all known MetaData in the file.
  - void `dataErrorFixup` (const **DataInfo** &dataInfo, **BList**< **DataError** > &errors)
  - Fixup data errors, mainly start/end times to be within data.
  - **BInt64** `timeCompare` ( **BTimeStamp** t1, **BTimeStamp** t2, **BUInt** diff)
  - Compare timestamps with a margin.
  - **int** `duplicateCheck` (const **DataBlock** &data1, const **DataBlock** &data2, **BUInt** channel=0)
  - Check if blocks are duplicates.
  - **BUInt64** `getFilePosition` ()

## Static Public Member Functions

- static **DataFormat** `getFormats` ()
- Get list of supported formats.

## Protected Attributes

- **BString** `ofilename`
- **BString** `omode`
- **BTimeStamp** `ofilenameTime`
- **BFile** `ofile`
- **BString** `oformat`



### 7.36.1 Detailed Description

This class defines the interface for generic data file access that all of the BDS data conterors share.

### 7.36.2 Member Enumeration Documentation

#### 7.36.2.1 DataOrder

```
enum Bds::DataFile::DataOrder
```

##### Enumerator

DataOrderUnknown	
DataOrderAll	
DataOrderSample	
DataOrderChannel	

#### 7.36.2.2 Features

```
enum Bds::DataFile::Features
```

##### Enumerator

FeatureNone	
FeatureCanWrite	
FeatureCanRead	

#### 7.36.2.3 ReadOptionsList

```
enum Bds::DataFile::ReadOptionsList
```

##### Enumerator

ReadOptionNone	
ReadOptionValidate	
ReadOptionFileNameProcess	
ReadOptionValidateCorruptions	
ReadOptionReorder	
ReadOptionDeleteDuplicates	
ReadOptionInfoExtra	
ReadOptionIgnoreSamplerate	
ReadOptionPrintBlocks	



#### 7.36.2.4 WriteOptionsList

enum [Bds::DataFile::WriteOptionsList](#)

##### Enumerator

WriteOptionNone	
WriteOptionSensorData	
WriteOptionNoMetadata	

### 7.36.3 Constructor & Destructor Documentation

#### 7.36.3.1 DataFile()

`Bds::DataFile::DataFile ( )`

#### 7.36.3.2 ~DataFile()

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

### 7.36.4 Member Function Documentation

#### 7.36.4.1 close()

**Error** `Bds::DataFile::close ( ) [virtual]`

Close the file.

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



#### 7.36.4.2 dataErrorFixup()

```
void Bds::DataFile::dataErrorFixup (
    const DataInfo & dataInfo,
    BList< DataError > & errors )
```

Fixup data errors, mainly start/end times to be within data.

#### 7.36.4.3 duplicateCheck()

```
int Bds::DataFile::duplicateCheck (
    const DataBlock & data1,
    const DataBlock & data2,
    BUInt channel = 0 )
```

Check if blocks are duplicates.

#### 7.36.4.4 end()

```
BError Bds::DataFile::end ( ) [virtual]
```

End write segment.

Reimplemented in [Bds::DataFileSeed](#), [Bds::DataFileLog](#), [Bds::DataFileAscii](#), and [Bds::DataFileIms](#).

#### 7.36.4.5 fileNameProcess()

```
BError Bds::DataFile::fileNameProcess ( ) [virtual]
```

Parse the file name for a date/time.

#### 7.36.4.6 flush()

```
BError Bds::DataFile::flush ( ) [virtual]
```

Flush data to disk.

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



#### 7.36.4.7 getDataOrder()

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

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

Reimplemented in [Bds::DataFileBds](#), [Bds::DataFileCd](#), [Bds::DataFileCss](#), [Bds::DataFileGcf](#), [Bds::DataFileSeed](#), [Bds::DataFileIms](#), [Bds::DataFileAscii](#), [Bds::DataFileLog](#), [Bds::DataFileWra](#), [Bds::DataFileAd22](#), [Bds::DataFileBdrs](#), [Bds::DataFileLac](#), and [Bds::DataFileWraAgso](#).

#### 7.36.4.8 getFeatures()

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

Get bitmask of supported features.

Reimplemented in [Bds::DataFileCd](#), [Bds::DataFileCss](#), [Bds::DataFileGcf](#), [Bds::DataFileSeed](#), [Bds::DataFileIms](#), [Bds::DataFileAscii](#), [Bds::DataFileLog](#), [Bds::DataFileResponse](#), [Bds::DataFileSac](#), [Bds::DataFileWra](#), [Bds::DataFileAd22](#), [Bds::DataFileBdrs](#), [Bds::DataFileLac](#), and [Bds::DataFileWraAgso](#).

#### 7.36.4.9 getFileName()

```
BString Bds::DataFile::getFileName ( ) [virtual]
```

Return the file name.

#### 7.36.4.10 getFilePosition()

```
BUInt64 Bds::DataFile::getFilePosition ( )
```

#### 7.36.4.11 getFormat()

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

Get sub-format.



#### 7.36.4.12 getFormats()

```
DataFormat Bds::DataFile::getFormats ( ) [static]
```

Get list of supported formats.

#### 7.36.4.13 getInfo()

```
BError Bds::DataFile::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

Reimplemented in [Bds::DataFileBds](#), [Bds::DataFileCd](#), [Bds::DataFileCss](#), [Bds::DataFileGcf](#), [Bds::DataFileSeed](#), [Bds::DataFileTapeDigitiser](#), [Bds::DataFileLog](#), [Bds::DataFileWra](#), [Bds::DataFileAd22](#), [Bds::DataFileBdrs](#), [Bds::DataFileLac](#), and [Bds::DataFileWraAgso](#).

#### 7.36.4.14 getMetaData()

```
BError Bds::DataFile::getMetaData (
    ChannelInfos & channelInfos ) [virtual]
```

Return all known MetaData in the file.

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

#### 7.36.4.15 init()

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

Initialise.

#### 7.36.4.16 open()

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

Open the file for read or write.

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



## 7.36.4.17 readData()

```
BError Bds::DataFile::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

Reimplemented in [Bds::DataFileBds](#), [Bds::DataFileCd](#), [Bds::DataFileCss](#), [Bds::DataFileGcf](#), [Bds::DataFileSeed](#), [Bds::DataFileTapeDigitiser](#), [Bds::DataFileLog](#), [Bds::DataFileWra](#), [Bds::DataFileAd22](#), [Bds::DataFileBdrs](#), [Bds::DataFileLac](#), and [Bds::DataFileWraAgso](#).

## 7.36.4.18 seekBlock()

```
BError Bds::DataFile::seekBlock (
    BUInt32 channel,
    BUInt segment,
    BTimeStamp time,
    BUInt32 & blockNumber,
    BUInt64 & sampleNumber,
    DataBlock & data ) [virtual]
```

Find requested block on given channel given a time.

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

## 7.36.4.19 setFormat()

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

Set the sub-format.

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

## 7.36.4.20 setInfo()

```
BError Bds::DataFile::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options = WriteOptionNone ) [virtual]
```

Set information on data for write.

Reimplemented in [Bds::DataFileBds](#), [Bds::DataFileSeed](#), [Bds::DataFileLog](#), [Bds::DataFileAscii](#), [Bds::DataFileImms](#), [Bds::DataFileResponse](#), [Bds::DataFileBknas](#), and [Bds::DataFileSac](#).



#### 7.36.4.21 start()

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

Start writing next segment of data.

Reimplemented in [Bds::DataFileSeed](#), [Bds::DataFileLog](#), [Bds::DataFileAscii](#), and [Bds::DataFileIms](#).

#### 7.36.4.22 timeCompare()

```
BInt64 Bds::DataFile::timeCompare (
    BTimeStamp t1,
    BTimeStamp t2,
    BUInt diff )
```

Compare timestamps with a margin.

#### 7.36.4.23 writeData()

```
BError Bds::DataFile::writeData (
    const DataBlock & data ) [virtual]
```

Write a block of data.

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

### 7.36.5 Member Data Documentation

#### 7.36.5.1 ofile

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

#### 7.36.5.2 ofileName

```
BString Bds::DataFile::ofileName [protected]
```



### 7.36.5.3 offileNameTime

**BTimeStamp** Bds::DataFile::offileNameTime [protected]

### 7.36.5.4 offormat

**BString** Bds::DataFile::offormat [protected]

### 7.36.5.5 omode

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

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

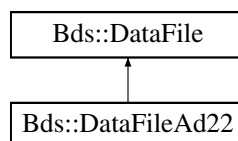
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp](#)

## 7.37 Bds::DataFileAd22 Class Reference

Data file convertor for AD22 format files.

```
#include <BdsDataFileAd22.h>
```

Inheritance diagram for Bds::DataFileAd22:



### Public Member Functions

- [DataFileAd22](#) ()
- 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, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get info on data.*
- **BError** [readData](#) ( **BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, [DataBlock](#) & data)  
*Read a block.*



## Static Public Member Functions

- static [DataFormat](#) `getFormats` ()

## Additional Inherited Members

### 7.37.1 Detailed Description

Data file convertor for AD22 format files.

### 7.37.2 Constructor & Destructor Documentation

#### 7.37.2.1 DataFileAd22()

```
Bds::DataFileAd22::DataFileAd22 ( )
```

### 7.37.3 Member Function Documentation

#### 7.37.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileAd22::getDataOrder ( ) [virtual]
```

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

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

#### 7.37.3.2 getFeatures()

```
int Bds::DataFileAd22::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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



### 7.37.3.3 getFormats()

```
DataFormat Bds::DataFileAd22::getFormats ( ) [static]
```

### 7.37.3.4 getInfo()

```
BError Bds::DataFileAd22::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

### 7.37.3.5 readData()

```
BError Bds::DataFileAd22::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

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

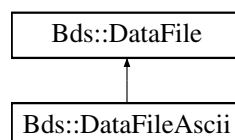
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp](#)

## 7.38 Bds::DataFileAscii Class Reference

Data file convertor for ASCII format files.

```
#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.*
- [DataOrder](#) [getDataOrder](#) ()  
*Get the expected order of writing data, by sample or by channel.*
- int [getFeatures](#) ()  
*Get bitmask of supported features.*
- **BError** [setFormat](#) ( **BString** format)  
*Set the sub-format.*
- **BError** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannellInfos](#) &channellInfos, [WriteOptionsList](#) options=[WriteOptionSensorData](#))  
*Set information on data for write.*
- **BError** [start](#) ( **BUInt** channel, **BUInt** segment)  
*Start writing next segment of data.*
- **BError** [writeData](#) (const [DataBlock](#) & data)  
*Write a block of data.*
- **BError** [end](#) ()  
*End write segment.*

## Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()

## Additional Inherited Members

### 7.38.1 Detailed Description

Data file convertor for ASCII format files.

### 7.38.2 Constructor & Destructor Documentation

#### 7.38.2.1 DataFileAscii()

```
Bds::DataFileAscii::DataFileAscii ( )
```

### 7.38.3 Member Function Documentation



#### 7.38.3.1 end()

```
BError Bds::DataFileAscii::end ( ) [virtual]
```

End write segment.

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

#### 7.38.3.2 getDataOrder()

```
DataFile::DataOrder Bds::DataFileAscii::getDataOrder ( ) [virtual]
```

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

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

#### 7.38.3.3 getFeatures()

```
int Bds::DataFileAscii::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

#### 7.38.3.4 getFormats()

```
DataFormat Bds::DataFileAscii::getFormats ( ) [static]
```

#### 7.38.3.5 open()

```
BError Bds::DataFileAscii::open (
    BString fileName,
    BString mode ) [virtual]
```

Open the file for read or write.

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



#### 7.38.3.6 setFormat()

```
BError Bds::DataFileAscii::setFormat (
    BString format ) [virtual]
```

Set the sub-format.

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

#### 7.38.3.7 setInfo()

```
BError Bds::DataFileAscii::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options = WriteOptionSensorData ) [virtual]
```

Set information on data for write.

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

#### 7.38.3.8 start()

```
BError Bds::DataFileAscii::start (
    BUInt channel,
    BUInt segment ) [virtual]
```

Start writing next segment of data.

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

#### 7.38.3.9 writeData()

```
BError Bds::DataFileAscii::writeData (
    const DataBlock & data ) [virtual]
```

Write a block of data.

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

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](#)

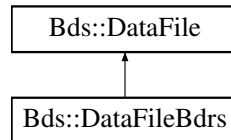


## 7.39 Bds::DataFileBdrs Class Reference

Data file convertor for BDRS format files.

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

Inheritance diagram for Bds::DataFileBdrs:



### Public Member Functions

- [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, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get info on data.*
- **BError** [readData](#) ( **BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, [DataBlock](#) & data)  
*Read a block.*

### Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()

### Additional Inherited Members

#### 7.39.1 Detailed Description

Data file convertor for BDRS format files.

#### 7.39.2 Constructor & Destructor Documentation

##### 7.39.2.1 DataFileBdrs()

```
Bds::DataFileBdrs::DataFileBdrs ( )
```



### 7.39.3 Member Function Documentation

#### 7.39.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileBdrs::getDataOrder ( ) [virtual]
```

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

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

#### 7.39.3.2 getFeatures()

```
int Bds::DataFileBdrs::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

#### 7.39.3.3 getFormats()

```
DataFormat Bds::DataFileBdrs::getFormats ( ) [static]
```

#### 7.39.3.4 getInfo()

```
BError Bds::DataFileBdrs::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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



### 7.39.3.5 readData()

```
BError Bds::DataFileBdrs::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

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

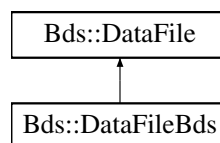
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp](#)

## 7.40 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](#) = 256 }
- 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.*
- **BError** [flush](#) ()  
*Flush any data to disk even if blocks are not full.*
- **BError** [close](#) ()  
*Close file.*
- **BError** [setFormat](#) ( **BString** format)  
*Sets the sub-format.*



- **BError** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannellInfos](#) &channellInfos, [WriteOptionsList](#) options=[WriteOptionSensorData](#))  
*Sets the information.*
- **BError** [writeData](#) (const [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, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get information on open file.*
- **BError** [seekBlock](#) ( **BUint32** channel, **BUint** segment, **BTimeStamp** time, **BUint32** &blockNumber, **BUint64** &sampleNumber, [DataBlock](#) &dataBlock)  
*Find the block that contains the samples for the time requested.*
- **BError** [readData](#) ( **BUint32** channel, **BUint** segment, **BUint32** blockNumber, [DataBlock](#) &dataBlock)  
*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 [DataFormat](#) [getFormats](#) ()  
*Get the names of the supported formats.*

### Additional Inherited Members

#### 7.40.1 Detailed Description

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

#### 7.40.2 Member Enumeration Documentation

##### 7.40.2.1 anonymous enum

anonymous enum

##### Enumerator

StreamsMax	
------------	--



#### 7.40.2.2 anonymous enum

anonymous enum

##### Enumerator

DefaultBlockSize	
------------------	--

#### 7.40.2.3 PackFormat

enum [Bds::DataFileBds::PackFormat](#)

##### Enumerator

PackFormat_Unknown	
PackFormat_SM	
PackFormat_CM	
PackFormat_SM_CC	

### 7.40.3 Constructor & Destructor Documentation

#### 7.40.3.1 DataFileBds()

```
Bds::DataFileBds::DataFileBds ( )
```

#### 7.40.3.2 ~DataFileBds()

```
Bds::DataFileBds::~~DataFileBds ( )
```

### 7.40.4 Member Function Documentation

#### 7.40.4.1 close()

```
BErr Bds::DataFileBds::close ( ) [virtual]
```

Close file.

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



#### 7.40.4.2 flush()

**BEError** Bds::DataFileBds::flush ( ) [virtual]

Flush any data to disk even if blocks are not full.

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

#### 7.40.4.3 getDataOrder()

[DataFile::DataOrder](#) Bds::DataFileBds::getDataOrder ( ) [virtual]

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

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

#### 7.40.4.4 getDiskBlockSize()

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

Returns the data block size in bytes.

#### 7.40.4.5 getFormats()

[DataFormat](#) Bds::DataFileBds::getFormats ( ) [static]

Get the names of the supported formats.

#### 7.40.4.6 getInfo()

**BEError** Bds::DataFileBds::getInfo (   
    [DataInfo](#) & dataInfo,   
    [DataFileOptions](#) options,   
    BList< [DataError](#) > & errors ) [virtual]

Get information on open file.

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



#### 7.40.4.7 open()

```
BError Bds::DataFileBds::open (
    BString fileName,
    BString mode ) [virtual]
```

Open the file for reading or writing.

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

#### 7.40.4.8 readData()

```
BError Bds::DataFileBds::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

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

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

#### 7.40.4.9 seekBlock()

```
BError Bds::DataFileBds::seekBlock (
    BUInt32 channel,
    BUInt segment,
    BTimeStamp time,
    BUInt32 & blockNumber,
    BUInt64 & sampleNumber,
    DataBlock & dataBlock ) [virtual]
```

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

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

#### 7.40.4.10 setDiskBlockSize()

```
BError Bds::DataFileBds::setDiskBlockSize (
    BUInt32 blockSize )
```

Sets up file/stream block size.



## 7.40.4.11 setFormat()

```
BError Bds::DataFileBds::setFormat (
    BString format ) [virtual]
```

Sets the sub-format.

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

## 7.40.4.12 setInfo()

```
BError Bds::DataFileBds::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options = WriteOptionSensorData ) [virtual]
```

Sets the information.

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

## 7.40.4.13 writeData()

```
BError Bds::DataFileBds::writeData (
    const DataBlock & data ) [virtual]
```

Writes a data block to the file.

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

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

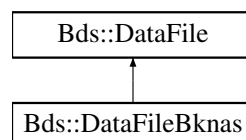
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp](#)

## 7.41 Bds::DataFileBknas Class Reference

Data file convertor for BKNAS format files.

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

Inheritance diagram for Bds::DataFileBknas:





## Public Member Functions

- [DataFileBknas](#) ()
- **Error** [open](#) ( **BString** fileName, **BString** mode)  
*Open the file for read or write.*
- **Error** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannellInfos](#) &channellInfos, [WriteOptionsList](#) options=[WriteOptionSensorData](#))  
*Set information on data for write.*
- **Error** [writeData](#) (const [DataBlock](#) & data)  
*Write a block of data.*

## Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()

## Additional Inherited Members

### 7.41.1 Detailed Description

Data file convertor for BKNAS format files.

### 7.41.2 Constructor & Destructor Documentation

#### 7.41.2.1 DataFileBknas()

```
Bds::DataFileBknas::DataFileBknas ( )
```

### 7.41.3 Member Function Documentation

#### 7.41.3.1 getFormats()

```
DataFormat Bds::DataFileBknas::getFormats ( ) [static]
```



## 7.41.3.2 open()

```
BError Bds::DataFileBknas::open (
    BString fileName,
    BString mode ) [virtual]
```

Open the file for read or write.

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

## 7.41.3.3 setInfo()

```
BError Bds::DataFileBknas::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options = WriteOptionSensorData ) [virtual]
```

Set information on data for write.

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

## 7.41.3.4 writeData()

```
BError Bds::DataFileBknas::writeData (
    const DataBlock & data ) [virtual]
```

Write a block of data.

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

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

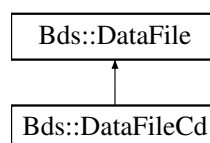
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp](#)

## 7.42 Bds::DataFileCd Class Reference

Data file convertor for CD1.0 and CD1.1 file formats.

```
#include <BdsDataFileCd.h>
```

Inheritance diagram for Bds::DataFileCd:





## Public Member Functions

- [DataFileCd](#) ()
- 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, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get info on data.*
- **BError** [readData](#) ( **BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, [DataBlock](#) & data)  
*Read a block.*

## Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()

## Additional Inherited Members

### 7.42.1 Detailed Description

Data file convertor for CD1.0 and CD1.1 file formats.

### 7.42.2 Constructor & Destructor Documentation

#### 7.42.2.1 DataFileCd()

```
Bds::DataFileCd::DataFileCd ( )
```

### 7.42.3 Member Function Documentation

#### 7.42.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileCd::getDataOrder ( ) [virtual]
```

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

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



## 7.42.3.2 getFeatures()

```
int Bds::DataFileCd::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

## 7.42.3.3 getFormats()

```
DataFormat Bds::DataFileCd::getFormats ( ) [static]
```

## 7.42.3.4 getInfo()

```
BError Bds::DataFileCd::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

## 7.42.3.5 readData()

```
BError Bds::DataFileCd::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

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

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

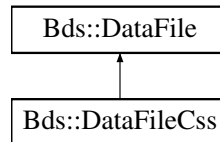


## 7.43 Bds::DataFileCss Class Reference

Data file convertor for CSS format files.

```
#include <BdsDataFileCss.h>
```

Inheritance diagram for Bds::DataFileCss:



### Public Member Functions

- [DataFileCss](#) ()
- 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, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get info on data.*
- **BError** [readData](#) ( **BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, [DataBlock](#) & data)  
*Read a block.*

### Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()

### Additional Inherited Members

#### 7.43.1 Detailed Description

Data file convertor for CSS format files.

#### 7.43.2 Constructor & Destructor Documentation

##### 7.43.2.1 DataFileCss()

```
Bds::DataFileCss::DataFileCss ( )
```



### 7.43.3 Member Function Documentation

#### 7.43.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileCss::getDataOrder ( ) [virtual]
```

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

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

#### 7.43.3.2 getFeatures()

```
int Bds::DataFileCss::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

#### 7.43.3.3 getFormats()

```
DataFormat Bds::DataFileCss::getFormats ( ) [static]
```

#### 7.43.3.4 getInfo()

```
BError Bds::DataFileCss::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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



### 7.43.3.5 readData()

```
BError Bds::DataFileCss::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

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

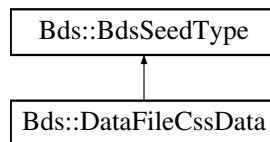
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp](#)

## 7.44 Bds::DataFileCssData Class Reference

[DataFileCss](#) internal CSS data type.

```
#include <BdsDataFileCss.h>
```

Inheritance diagram for Bds::DataFileCssData:



### Public Member Functions

- [DataFileCssData](#) ()
- [~DataFileCssData](#) ()
- **BError** [set](#) ( **BString** line)

### Public Attributes

- **BString** [sta](#)
- **BString** [chan](#)
- double [startTime](#)
- int [wfid](#)
- int [chanid](#)
- int [jdate](#)
- double [endTime](#)
- int [nsamp](#)
- double [sampleRate](#)
- double [calibrationFactor](#)
- double [calibrationFreq](#)



- **BString** [instType](#)
- **BString** [segtype](#)
- **BString** [datatype](#)
- **BString** [clip](#)
- **BString** [dirName](#)
- **BString** [fileName](#)
- **BUInt32** [fileOffset](#)
- **int** [commId](#)
- **BString** [loadDate](#)
- **BFile** \* [file](#)
- **BUInt32** [sampleFormat](#)
- **BUInt32** [sampleSize](#)
- **int** [sampleBigEndian](#)

### 7.44.1 Detailed Description

[DataFileCss](#) internal CSS data type.

### 7.44.2 Constructor & Destructor Documentation

#### 7.44.2.1 DataFileCssData()

```
Bds::DataFileCssData::DataFileCssData ( )
```

#### 7.44.2.2 ~DataFileCssData()

```
Bds::DataFileCssData::~~DataFileCssData ( )
```

### 7.44.3 Member Function Documentation

#### 7.44.3.1 set()

```
BError Bds::DataFileCssData::set (
    BString line )
```

### 7.44.4 Member Data Documentation



#### 7.44.4.1 calibrationFactor

```
double Bds::DataFileCssData::calibrationFactor
```

#### 7.44.4.2 calibrationFreq

```
double Bds::DataFileCssData::calibrationFreq
```

#### 7.44.4.3 chan

```
BString Bds::DataFileCssData::chan
```

#### 7.44.4.4 chanid

```
int Bds::DataFileCssData::chanid
```

#### 7.44.4.5 clip

```
BString Bds::DataFileCssData::clip
```

#### 7.44.4.6 commId

```
int Bds::DataFileCssData::commId
```

#### 7.44.4.7 datatype

```
BString Bds::DataFileCssData::datatype
```

#### 7.44.4.8 dirName

```
BString Bds::DataFileCssData::dirName
```



#### 7.44.4.9 endTime

double Bds::DataFileCssData::endTime

#### 7.44.4.10 file

**BFile\*** Bds::DataFileCssData::file

#### 7.44.4.11 fileName

**BString** Bds::DataFileCssData::fileName

#### 7.44.4.12 fileOffset

**BUInt32** Bds::DataFileCssData::fileOffset

#### 7.44.4.13 instType

**BString** Bds::DataFileCssData::instType

#### 7.44.4.14 jdate

int Bds::DataFileCssData::jdate

#### 7.44.4.15 loadDate

**BString** Bds::DataFileCssData::loadDate

#### 7.44.4.16 nsamp

int Bds::DataFileCssData::nsamp



#### 7.44.4.17 sampleBigEndian

```
int Bds::DataFileCssData::sampleBigEndian
```

#### 7.44.4.18 sampleFormat

```
BUInt32 Bds::DataFileCssData::sampleFormat
```

#### 7.44.4.19 sampleRate

```
double Bds::DataFileCssData::sampleRate
```

#### 7.44.4.20 sampleSize

```
BUInt32 Bds::DataFileCssData::sampleSize
```

#### 7.44.4.21 segtype

```
BString Bds::DataFileCssData::segtype
```

#### 7.44.4.22 sta

```
BString Bds::DataFileCssData::sta
```

#### 7.44.4.23 startTime

```
double Bds::DataFileCssData::startTime
```



## 7.44.4.24 wfid

```
int Bds::DataFileCssData::wfid
```

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

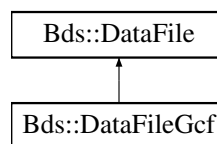
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp](#)

## 7.45 Bds::DataFileGcf Class Reference

Data file convertor for GCF format files.

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

Inheritance diagram for Bds::DataFileGcf:



### Public Member Functions

- [DataFileGcf](#) ()
- [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, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get info on data.*
- **BError** [readData](#) ( **BUint32** channel, **BUint** segment, **BUint32** blockNumber, [DataBlock](#) & data)  
*Read a block.*

### Static Public Member Functions

- static [DataFormat getFormats](#) ()

### Additional Inherited Members

#### 7.45.1 Detailed Description

Data file convertor for GCF format files.



## 7.45.2 Constructor & Destructor Documentation

### 7.45.2.1 DataFileGcf()

```
Bds::DataFileGcf::DataFileGcf ( )
```

## 7.45.3 Member Function Documentation

### 7.45.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileGcf::getDataOrder ( ) [virtual]
```

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

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

### 7.45.3.2 getFeatures()

```
int Bds::DataFileGcf::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

### 7.45.3.3 getFormats()

```
DataFormat Bds::DataFileGcf::getFormats ( ) [static]
```

### 7.45.3.4 getInfo()

```
BError Bds::DataFileGcf::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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



## 7.45.3.5 readData()

```
BError Bds::DataFileGcf::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

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

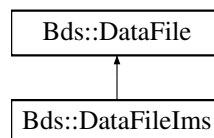
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp](#)

## 7.46 Bds::DataFileIms Class Reference

Data file convertor for IMS format files.

```
#include <BdsDataFileIms.h>
```

Inheritance diagram for Bds::DataFileIms:



### Public Member Functions

- [DataFileIms](#) ()
- **BError** [open](#) ( **BString** fileName, **BString** mode)  
*Open the file for read or write.*
- **BError** [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](#) (const [DataInfo](#) &dataInfo, const [ChannellInfos](#) &channellInfos, [WriteOptionsList](#) options=[WriteOptionNone](#))  
*Set information on data for write.*
- **BError** [start](#) ( **BUInt** channel, **BUInt** segment)  
*Start writing next segment of data.*
- **BError** [writeData](#) (const [DataBlock](#) & data)  
*Write a block of data.*
- **BError** [end](#) ()  
*End write segment.*
- **BError** [getMetaData](#) ([ChannellInfos](#) &channellInfos)  
*Return all known MetaData in the file.*



## Static Public Member Functions

- static [DataFormat](#) `getFormats` ()

## Additional Inherited Members

### 7.46.1 Detailed Description

Data file convertor for IMS format files.

### 7.46.2 Constructor & Destructor Documentation

#### 7.46.2.1 `DataFileIms()`

```
Bds::DataFileIms::DataFileIms ( )
```

### 7.46.3 Member Function Documentation

#### 7.46.3.1 `close()`

```
BEError Bds::DataFileIms::close ( ) [virtual]
```

Close the file.

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

#### 7.46.3.2 `end()`

```
BEError Bds::DataFileIms::end ( ) [virtual]
```

End write segment.

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



### 7.46.3.3 getDataOrder()

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

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

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

### 7.46.3.4 getFeatures()

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

Get bitmask of supported features.

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

### 7.46.3.5 getFormats()

```
DataFormat Bds::DataFileIms::getFormats ( ) [static]
```

### 7.46.3.6 getMetaData()

```
BError Bds::DataFileIms::getMetaData (
    ChannelInfos & channelInfos ) [virtual]
```

Return all known MetaData in the file.

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

### 7.46.3.7 open()

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

Open the file for read or write.

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



## 7.46.3.8 setInfo()

```
BError Bds::DataFileIms::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options = WriteOptionNone ) [virtual]
```

Set information on data for write.

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

## 7.46.3.9 start()

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

Start writing next segment of data.

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

## 7.46.3.10 writeData()

```
BError Bds::DataFileIms::writeData (
    const DataBlock & data ) [virtual]
```

Write a block of data.

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

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

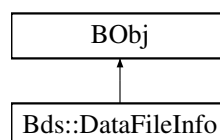
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp](#)

## 7.47 Bds::DataFileInfo Class Reference

This class defines information on a sensor data file.

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

Inheritance diagram for Bds::DataFileInfo:





## Public Member Functions

- **DataFileInfo** ( **BUInt32** *id*=0, **BTimeStamp** *startTime*= **BTimeStamp**(), **BTimeStamp** *endTime*= **BTimeStamp**(), **BString** *location*= **BString**(), **BString** *format*= **BString**(), **BString** *url*= **BString**(), **BString** *comment*= **BString**(), **BUInt32** *importUserId*=0, **BTimeStamp** *importTime*= **BTimeStamp**(), **BString** *state*= **BString**() )
- **BString** *getType* ()
- **BError** *setMembers* ( **BDictString** &members)
- **BError** *setMember* ( **BString** name, **BString** value)
- **BError** *getMembers* ( **BDictString** &members)
- **BError** *getMember* ( **BString** name, **BString** &value)

## Public Attributes

- **BUInt32** *id*  
*Unique ID when stored in a database or for other uses.*
- **BTimeStamp** *startTime*  
*The Start Time.*
- **BTimeStamp** *endTime*  
*The End Time.*
- **BString** *location*  
*The storage location.*
- **BString** *format*  
*The data format.*
- **BString** *url*  
*The URL for file access.*
- **BString** *comment*  
*A comment on the file.*
- **BUInt32** *importUserId*  
*The user ID of the importing user.*
- **BTimeStamp** *importTime*  
*The Time the data was imported.*
- **BString** *state*  
*Status info on the import (importing, realtime, failed, ok etc)*

### 7.47.1 Detailed Description

This class defines information on a sensor data file.

The raw sensor data for a seismic channel is stored in files in the BDS system. This class defines the database entry that describes this file, its storage location and status. A single file can store one or more channels of seismic data in different formats.

### 7.47.2 Constructor & Destructor Documentation



### 7.47.2.1 DataFileInfo()

```
Bds::DataFileInfo::DataFileInfo (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString location = BString(),
    BString format = BString(),
    BString url = BString(),
    BString comment = BString(),
    BUInt32 importUserId = 0,
    BTimeStamp importTime = BTimeStamp(),
    BString state = BString() )
```

## 7.47.3 Member Function Documentation

### 7.47.3.1 getMember()

```
BError Bds::DataFileInfo::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

### 7.47.3.2 getMembers()

```
BError Bds::DataFileInfo::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.47.3.3 getType()

```
BString Bds::DataFileInfo::getType ( )
```

### 7.47.3.4 setMember()

```
BError Bds::DataFileInfo::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.



#### 7.47.3.5 setMembers()

```
BError Bds::DataFileInfo::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.47.4 Member Data Documentation

#### 7.47.4.1 comment

```
BString Bds::DataFileInfo::comment
```

A comment on the file.

#### 7.47.4.2 endTime

```
BTimeStamp Bds::DataFileInfo::endTime
```

The End Time.

#### 7.47.4.3 format

```
BString Bds::DataFileInfo::format
```

The data format.

#### 7.47.4.4 id

```
BUInt32 Bds::DataFileInfo::id
```

Unique ID when stored in a database or for other uses.

#### 7.47.4.5 importTime

```
BTimeStamp Bds::DataFileInfo::importTime
```

The Time the data was imported.



#### 7.47.4.6 importUserId

**BUInt32** Bds::DataFileInfo::importUserId

The user ID of the importing user.

#### 7.47.4.7 location

**BString** Bds::DataFileInfo::location

The storage location.

#### 7.47.4.8 startTime

**BTimeStamp** Bds::DataFileInfo::startTime

The Start Time.

#### 7.47.4.9 state

**BString** Bds::DataFileInfo::state

Status info on the import (importing, realtime, failed, ok etc)

#### 7.47.4.10 url

**BString** Bds::DataFileInfo::url

The URL for file access.

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

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

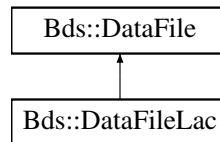


## 7.48 Bds::DataFileLac Class Reference

Data file convertor for LAC format files.

```
#include <BdsDataFileLac.h>
```

Inheritance diagram for Bds::DataFileLac:



### Public Member Functions

- [DataFileLac](#) ()
- 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, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get info on data.*
- **BError** [readData](#) ( **BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, [DataBlock](#) & data)  
*Read a block.*

### Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()

### Additional Inherited Members

#### 7.48.1 Detailed Description

Data file convertor for LAC format files.

#### 7.48.2 Constructor & Destructor Documentation

##### 7.48.2.1 DataFileLac()

```
Bds::DataFileLac::DataFileLac ( )
```



### 7.48.3 Member Function Documentation

#### 7.48.3.1 `getDataOrder()`

```
DataFile::DataOrder Bds::DataFileLac::getDataOrder ( ) [virtual]
```

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

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

#### 7.48.3.2 `getFeatures()`

```
int Bds::DataFileLac::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

#### 7.48.3.3 `getFormats()`

```
DataFormat Bds::DataFileLac::getFormats ( ) [static]
```

#### 7.48.3.4 `getInfo()`

```
BError Bds::DataFileLac::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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



## 7.48.3.5 readData()

```
BError Bds::DataFileLac::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

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

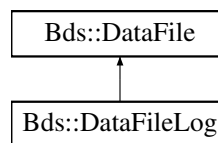
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp

## 7.49 Bds::DataFileLog Class Reference

Data file convertor for LOG format files.

```
#include <BdsDataFileLog.h>
```

Inheritance diagram for Bds::DataFileLog:



### Public Member Functions

- [DataFileLog](#) ()
- **BError** [open](#) ( **BString** fileName, **BString** mode)  
*Open the file for read or write.*
- [DataOrder](#) [getDataOrder](#) ()  
*Get the expected order of writing data, by sample or by channel.*
- int [getFeatures](#) ()  
*Get bitmask of supported features.*
- **BError** [getInfo](#) ([DataInfo](#) &dataInfo, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get info on data.*
- **BError** [readData](#) ( **BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, [DataBlock](#) & data)  
*Read a block.*
- **BError** [setFormat](#) ( **BString** format)  
*Set the sub-format.*
- **BError** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannellInfos](#) &channellInfos, [WriteOptionsList](#) options=[WriteOptionSensorData](#))  
*Set information on data for write.*
- **BError** [start](#) ( **BUInt** channel, **BUInt** segment)  
*Start writing next segment of data.*
- **BError** [writeData](#) (const [DataBlock](#) & data)  
*Write a block of data.*
- **BError** [end](#) ()  
*End write segment.*



## Static Public Member Functions

- static [DataFormat](#) `getFormats` ()

## Additional Inherited Members

### 7.49.1 Detailed Description

Data file convertor for LOG format files.

### 7.49.2 Constructor & Destructor Documentation

#### 7.49.2.1 `DataFileLog()`

```
Bds::DataFileLog::DataFileLog ( )
```

### 7.49.3 Member Function Documentation

#### 7.49.3.1 `end()`

```
BEError Bds::DataFileLog::end ( ) [virtual]
```

End write segment.

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

#### 7.49.3.2 `getDataOrder()`

```
DataFile::DataOrder Bds::DataFileLog::getDataOrder ( ) [virtual]
```

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

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



#### 7.49.3.3 getFeatures()

```
int Bds::DataFileLog::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

#### 7.49.3.4 getFormats()

```
DataFormat Bds::DataFileLog::getFormats ( ) [static]
```

#### 7.49.3.5 getInfo()

```
BError Bds::DataFileLog::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

#### 7.49.3.6 open()

```
BError Bds::DataFileLog::open (
    BString fileName,
    BString mode ) [virtual]
```

Open the file for read or write.

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

#### 7.49.3.7 readData()

```
BError Bds::DataFileLog::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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



#### 7.49.3.8 setFormat()

```
BError Bds::DataFileLog::setFormat (
    BString format ) [virtual]
```

Set the sub-format.

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

#### 7.49.3.9 setInfo()

```
BError Bds::DataFileLog::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options = WriteOptionSensorData ) [virtual]
```

Set information on data for write.

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

#### 7.49.3.10 start()

```
BError Bds::DataFileLog::start (
    BUInt channel,
    BUInt segment ) [virtual]
```

Start writing next segment of data.

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

#### 7.49.3.11 writeData()

```
BError Bds::DataFileLog::writeData (
    const DataBlock & data ) [virtual]
```

Write a block of data.

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

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

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



## 7.50 Bds::DataFileOptions Class Reference

This defines a list of BDS data convtor options.

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

### Public Member Functions

- [DataFileOptions](#) (int options=0)
- [operator int](#) ()
- [DataFileOptions](#) & [operator|=](#) (int o)

### Public Attributes

- int [optionList](#)
- **BArray**< **BUInt** > [oignoreBlockList](#)

#### 7.50.1 Detailed Description

This defines a list of BDS data convtor options.

#### 7.50.2 Constructor & Destructor Documentation

##### 7.50.2.1 DataFileOptions()

```
Bds::DataFileOptions::DataFileOptions (
    int options = 0 ) [inline]
```

#### 7.50.3 Member Function Documentation

##### 7.50.3.1 operator int()

```
Bds::DataFileOptions::operator int ( ) [inline]
```

##### 7.50.3.2 operator" |=()

```
DataFileOptions& Bds::DataFileOptions::operator|= (
    int o ) [inline]
```



## 7.50.4 Member Data Documentation

### 7.50.4.1 oignoreBlockList

```
BArray< BUInt> Bds::DataFileOptions::oignoreBlockList
```

### 7.50.4.2 ooptionList

```
int Bds::DataFileOptions::ooptionList
```

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

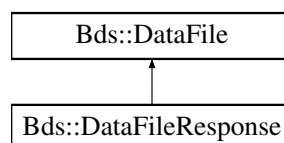
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h](#)

## 7.51 Bds::DataFileResponse Class Reference

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

```
#include <BdsDataFileResponse.h>
```

Inheritance diagram for Bds::DataFileResponse:



### Public Member Functions

- [DataFileResponse](#) ()
- int [getFeatures](#) ()  
*Get bitmask of supported features.*
- **BError** [getMetaData](#) ([ChannelInfos](#) &channelInfos)  
*Return all known MetaData in the file.*
- **BError** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannelInfos](#) &channelInfos, [WriteOptionsList](#) options)  
*Set information on data for write.*

### Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()



## Additional Inherited Members

### 7.51.1 Detailed Description

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

### 7.51.2 Constructor & Destructor Documentation

#### 7.51.2.1 DataFileResponse()

```
Bds::DataFileResponse::DataFileResponse ( )
```

### 7.51.3 Member Function Documentation

#### 7.51.3.1 getFeatures()

```
int Bds::DataFileResponse::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

#### 7.51.3.2 getFormats()

```
DataFormat Bds::DataFileResponse::getFormats ( ) [static]
```

#### 7.51.3.3 getMetaData()

```
BEError Bds::DataFileResponse::getMetaData (
    ChannelInfos & channelInfos ) [virtual]
```

Return all known MetaData in the file.

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



#### 7.51.3.4 setInfo()

```
BError Bds::DataFileResponse::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options ) [virtual]
```

Set information on data for write.

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

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

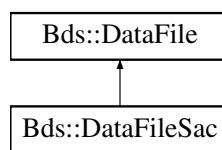
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp](#)

## 7.52 Bds::DataFileSac Class Reference

Data file convertor for SAC format files.

```
#include <BdsDataFileSac.h>
```

Inheritance diagram for Bds::DataFileSac:



### Public Member Functions

- [DataFileSac](#) ()
- int [getFeatures](#) ()  
*Get bitmask of supported features.*
- **BError** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannelInfos](#) &channelInfos, [WriteOptionsList](#) options=[WriteOptionNone](#))  
*Set information on data for write.*

### Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()

### Additional Inherited Members

#### 7.52.1 Detailed Description

Data file convertor for SAC format files.



## 7.52.2 Constructor & Destructor Documentation

### 7.52.2.1 DataFileSac()

```
Bds::DataFileSac::DataFileSac ( )
```

## 7.52.3 Member Function Documentation

### 7.52.3.1 getFeatures()

```
int Bds::DataFileSac::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

### 7.52.3.2 getFormats()

```
DataFormat Bds::DataFileSac::getFormats ( ) [static]
```

### 7.52.3.3 setInfo()

```
BError Bds::DataFileSac::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options = WriteOptionNone ) [virtual]
```

Set information on data for write.

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

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

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

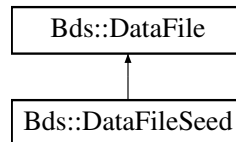


## 7.53 Bds::DataFileSeed Class Reference

Data file convertor for SEED file formats.

```
#include <BdsDataFileSeed.h>
```

Inheritance diagram for Bds::DataFileSeed:



### Public Member Functions

- [DataFileSeed](#) ()
- [~DataFileSeed](#) ()
- **BError** [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** [setFormat](#) ( **BString** format)  
*Set the sub-format.*
- **BError** [getInfo](#) ( [DataInfo](#) &dataInfo, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get info on data.*
- **BError** [readData](#) ( **BUint32** channel, **BUint** segment, **BUint32** blockNumber, [DataBlock](#) & data)  
*Read a block.*
- **BError** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannellInfos](#) &channellInfos, [WriteOptionsList](#) options=[WriteOptionSensorData](#))  
*Set information on data for write.*
- **BError** [start](#) ( **BUint** channel, **BUint** segment)  
*Start writing next segment of data.*
- **BError** [writeData](#) (const [DataBlock](#) & data)  
*Write a block of data.*
- **BError** [end](#) ()  
*End write segment.*
- void [msrFileWrite](#) (void \* data, int len)

### Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()

### Static Public Attributes

- static **BError** [omsrErr](#)  
*MSR processing error.*
- static int [onoLock](#)  
*Disable libmseed locking.*



## Additional Inherited Members

### 7.53.1 Detailed Description

Data file convertor for SEED file formats.

### 7.53.2 Constructor & Destructor Documentation

#### 7.53.2.1 DataFileSeed()

```
Bds::DataFileSeed::DataFileSeed ( )
```

#### 7.53.2.2 ~DataFileSeed()

```
Bds::DataFileSeed::~~DataFileSeed ( )
```

### 7.53.3 Member Function Documentation

#### 7.53.3.1 close()

```
BEError Bds::DataFileSeed::close ( ) [virtual]
```

Close the file.

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

#### 7.53.3.2 end()

```
BEError Bds::DataFileSeed::end ( ) [virtual]
```

End write segment.

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



#### 7.53.3.3 `getDataOrder()`

```
DataFile::DataOrder Bds::DataFileSeed::getDataOrder ( ) [virtual]
```

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

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

#### 7.53.3.4 `getFeatures()`

```
int Bds::DataFileSeed::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

#### 7.53.3.5 `getFormats()`

```
DataFormat Bds::DataFileSeed::getFormats ( ) [static]
```

#### 7.53.3.6 `getInfo()`

```
BError Bds::DataFileSeed::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

#### 7.53.3.7 `msrFileWrite()`

```
void Bds::DataFileSeed::msrFileWrite (
    void * data,
    int len )
```



#### 7.53.3.8 readData()

```
BError Bds::DataFileSeed::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

#### 7.53.3.9 setFormat()

```
BError Bds::DataFileSeed::setFormat (
    BString format ) [virtual]
```

Set the sub-format.

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

#### 7.53.3.10 setInfo()

```
BError Bds::DataFileSeed::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options = WriteOptionSensorData ) [virtual]
```

Set information on data for write.

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

#### 7.53.3.11 start()

```
BError Bds::DataFileSeed::start (
    BUInt channel,
    BUInt segment ) [virtual]
```

Start writing next segment of data.

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



### 7.53.3.12 writeData()

```
BError Bds::DataFileSeed::writeData (
    const DataBlock & data ) [virtual]
```

Write a block of data.

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

## 7.53.4 Member Data Documentation

### 7.53.4.1 omsrErr

```
BError Bds::DataFileSeed::omsrErr [static]
```

MSR processing error.

### 7.53.4.2 onoLock

```
int Bds::DataFileSeed::onoLock [static]
```

Disable libmseed locking.

Disable libmseed lock, for sequential programs.

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

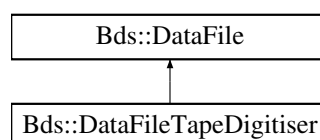
- [/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp](#)

## 7.54 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](#) ()
- **Error** [open](#) ( [BString](#) fileName, [BString](#) mode)  
*Open the file for reading or writing.*
- **Error** [getInfo](#) ([DataInfo](#) &dataInfo, [DataFileOptions](#) options, [BList](#)< [DataError](#) > &errors)  
*Get info on data.*
- **Error** [readData](#) ( [BUInt32](#) channel, [BUInt](#) segment, [BUInt32](#) blockNumber, [DataBlock](#) & data)  
*Read a block.*

## Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()

## Additional Inherited Members

### 7.54.1 Detailed Description

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

### 7.54.2 Constructor & Destructor Documentation

#### 7.54.2.1 DataFileTapeDigitiser()

```
Bds::DataFileTapeDigitiser::DataFileTapeDigitiser ( )
```

### 7.54.3 Member Function Documentation

#### 7.54.3.1 getFormats()

```
DataFormat Bds::DataFileTapeDigitiser::getFormats ( ) [static]
```

#### 7.54.3.2 getInfo()

```
Error Bds::DataFileTapeDigitiser::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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



### 7.54.3.3 open()

```
BError Bds::DataFileTapeDigitiser::open (
    BString fileName,
    BString mode ) [virtual]
```

Open the file for reading or writing.

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

### 7.54.3.4 readData()

```
BError Bds::DataFileTapeDigitiser::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

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

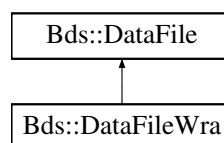
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp](#)

## 7.55 Bds::DataFileWra Class Reference

Data file convertor for WRA format files.

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

Inheritance diagram for Bds::DataFileWra:



### Public Member Functions

- [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, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get info on data.*
- **BError** [readData](#) ( **BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, [DataBlock](#) & data)  
*Read a block.*



## Static Public Member Functions

- static [DataFormat](#) `getFormats` ()

## Additional Inherited Members

### 7.55.1 Detailed Description

Data file convertor for WRA format files.

### 7.55.2 Constructor & Destructor Documentation

#### 7.55.2.1 DataFileWra()

```
Bds::DataFileWra::DataFileWra ( )
```

### 7.55.3 Member Function Documentation

#### 7.55.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileWra::getDataOrder ( ) [virtual]
```

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

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

#### 7.55.3.2 getFeatures()

```
int Bds::DataFileWra::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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



### 7.55.3.3 getFormats()

```
DataFormat Bds::DataFileWra::getFormats ( ) [static]
```

### 7.55.3.4 getInfo()

```
BError Bds::DataFileWra::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

### 7.55.3.5 readData()

```
BError Bds::DataFileWra::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

### 7.55.3.6 setFormat()

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

Set the sub-format.

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

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

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

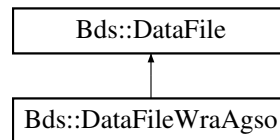


## 7.56 Bds::DataFileWraAgso Class Reference

Data file convertor for WRA AGSO format files.

```
#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, [DataFileOptions](#) options, **BList**< [DataError](#) > &errors)  
*Get info on data.*
- **BError** [readData](#) ( **BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, [DataBlock](#) & data)  
*Read a block.*

### Static Public Member Functions

- static [DataFormat](#) [getFormats](#) ()

### Additional Inherited Members

#### 7.56.1 Detailed Description

Data file convertor for WRA AGSO format files.

#### 7.56.2 Constructor & Destructor Documentation

##### 7.56.2.1 DataFileWraAgso()

```
Bds::DataFileWraAgso::DataFileWraAgso ( )
```



## 7.56.3 Member Function Documentation

### 7.56.3.1 `getDataOrder()`

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

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

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

### 7.56.3.2 `getFeatures()`

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

Get bitmask of supported features.

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

### 7.56.3.3 `getFormats()`

```
DataFormat Bds::DataFileWraAgso::getFormats ( ) [static]
```

### 7.56.3.4 `getInfo()`

```
BError Bds::DataFileWraAgso::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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



## 7.56.3.5 readData()

```
BError Bds::DataFileWraAgso::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

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](#)

## 7.57 Bds::DataFormat Class Reference

This holds information on a seismic data format.

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

### Public Member Functions

- [DataFormat](#) ( **BList**< **BString** > names= **BList**< **BString** >(), **BString** description= **BString**(), **BInt32** dataRead=0, **BInt32** dataWrite=0, **BInt32** metaDataRead=0, **BInt32** metaDataWrite=0, **BString** extension= **BString**())

### Public Attributes

- **BList**< **BString** > [names](#)  
*The format names.*
- **BString** [description](#)  
*The description.*
- **BInt32** [dataRead](#)  
*Ability to read data.*
- **BInt32** [dataWrite](#)  
*Ability to write data.*
- **BInt32** [metaDataRead](#)  
*MetaData read supported.*
- **BInt32** [metaDataWrite](#)  
*MetaData write supported.*
- **BString** [extension](#)  
*Default filename extension.*



### 7.57.1 Detailed Description

This holds information on a seismic data format.

It is used by the BDS data convertors to define which data formats they support..

### 7.57.2 Constructor & Destructor Documentation

#### 7.57.2.1 DataFormat()

```
Bds::DataFormat::DataFormat (
    BList< BString > names = BList< BString >(),
    BString description = BString(),
    BInt32 dataRead = 0,
    BInt32 dataWrite = 0,
    BInt32 metaDataRead = 0,
    BInt32 metaDataWrite = 0,
    BString extension = BString() )
```

### 7.57.3 Member Data Documentation

#### 7.57.3.1 dataRead

**BInt32** Bds::DataFormat::dataRead

Ability to read data.

#### 7.57.3.2 dataWrite

**BInt32** Bds::DataFormat::dataWrite

Ability to write data.

#### 7.57.3.3 description

**BString** Bds::DataFormat::description

The description.



## 7.57.3.4 extension

**BString** Bds::DataFormat::extension

Default filename extension.

## 7.57.3.5 metaDataRead

**BInt32** Bds::DataFormat::metaDataRead

MetaData read supported.

## 7.57.3.6 metaDataWrite

**BInt32** Bds::DataFormat::metaDataWrite

MetaData write supported.

## 7.57.3.7 names

**BList< BString >** Bds::DataFormat::names

The format names.

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

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

## 7.58 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< DataFormat >** &formats)
- **BError** formatGet ( **BString** format, [DataFile](#) \*&dataFile)



## Protected Member Functions

- int `findFormat` (`DataFormat` dataFormat, `BString` string)

### 7.58.1 Detailed Description

This class defines the interface for generic data file access.

It allows programs to get a list of all of the supported data formats and to get a suitable BDS data convertor for accessing the file.

### 7.58.2 Constructor & Destructor Documentation

#### 7.58.2.1 DataFormats()

```
Bds::DataFormats::DataFormats ( )
```

#### 7.58.2.2 ~DataFormats()

```
Bds::DataFormats::~~DataFormats ( )
```

### 7.58.3 Member Function Documentation

#### 7.58.3.1 findFormat()

```
int Bds::DataFormats::findFormat (
    DataFormat dataFormat,
    BString string ) [protected]
```

#### 7.58.3.2 formatGet()

```
BError Bds::DataFormats::formatGet (
    BString format,
    DataFile *& dataFile )
```



## 7.58.3.3 formatList()

```
BError Bds::DataFormats::formatList (
    BList< DataFormat > & formats )
```

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](#)

## 7.59 Bds::DataHandle Class Reference

This defines a handle to a sensor data stream/file when opened for read or write.

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

### Public Member Functions

- [DataHandle](#) ( [BUInt32](#) handle=0, [BUInt32](#) dataFileId=0)

### Public Attributes

- [BUInt32](#) handle  
*Opaque file handle.*
- [BUInt32](#) dataFileId  
*The data file ID if opened for write.*

### 7.59.1 Detailed Description

This defines a handle to a sensor data stream/file when opened for read or write.

### 7.59.2 Constructor & Destructor Documentation

## 7.59.2.1 DataHandle()

```
Bds::DataHandle::DataHandle (
    BUInt32 handle = 0,
    BUInt32 dataFileId = 0 )
```

### 7.59.3 Member Data Documentation



### 7.59.3.1 dataFileId

**BUInt32** Bds::DataHandle::dataFileId

The data file ID if opened for write.

### 7.59.3.2 handle

**BUInt32** Bds::DataHandle::handle

Opaque file handle.

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

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

## 7.60 Bds::DataInfo Class Reference

This class defines information on a set of data.

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

### Public Member Functions

- **DataInfo** ( **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), **BString** array= **BString**(), **BString** description= **BString**(), **BUInt32** synchronous=0, **BArray**< **BArray**< **DataChannel** > > channels= **BArray**< **BArray**< **DataChannel** > >(), **BDict**< **BString** > info= **BDict**< **BString** >(), **BDict**< **BString** > infoExtra= **BDict**< **BString** >(), **BList**< **BString** > warnings= **BList**< **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  
*Info on the set of channels.*
- **BDict**< **BString** > infoExtra  
*Extra Info on the set of channels. Used for extended error/logging information.*
- **BList**< **BString** > warnings  
*Warnings on the data set.*



## 7.60.1 Detailed Description

This class defines information on a set of data.

This describes a set of seismic data. It returns basic information when performing a selection of data or detailed information when enquiring information from an actual seismic data file. All of the detailed information comes from the data files themselves. This includes the info, infoExtra and warnings information. The channels array contains an array of data segments per channel. For basic information this could be a single segment over a time period. However, when enquiring detailed information from a file it will contain an entry per contiguous data segment in the file.

## 7.60.2 Constructor & Destructor Documentation

### 7.60.2.1 DataInfo()

```
Bds::DataInfo::DataInfo (
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString array = BString(),
    BString description = BString(),
    BUInt32 synchronous = 0,
    BArray< BArray< DataChannel > > channels = BArray< BArray<DataChannel > >(),
    BDict< BString > info = BDict< BString >(),
    BDict< BString > infoExtra = BDict< BString >(),
    BList< BString > warnings = BList< BString >() )
```

## 7.60.3 Member Data Documentation

### 7.60.3.1 array

**BString** Bds::DataInfo::array

The Seismic Array that all of the channels are from, if just one.

### 7.60.3.2 channels

**BArray< BArray<DataChannel > >** Bds::DataInfo::channels

The Data channels. Each channel can have multiple segments of data.



### 7.60.3.3 description

**BString** Bds::DataInfo::description

The Comment.

### 7.60.3.4 endTime

**BTimeStamp** Bds::DataInfo::endTime

The End Time.

### 7.60.3.5 info

**BDict**< **BString** > Bds::DataInfo::info

Info on the set of channels.

### 7.60.3.6 infoExtra

**BDict**< **BString** > Bds::DataInfo::infoExtra

Extra Info on the set of channels. Used for extended error/logging information.

### 7.60.3.7 startTime

**BTimeStamp** Bds::DataInfo::startTime

The Start Time.

### 7.60.3.8 synchronous

**BUInt32** Bds::DataInfo::synchronous

The channels are synchronously sampled.



## 7.60.3.9 warnings

```
BList< BString > Bds::DataInfo::warnings
```

Warnings on the data set.

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

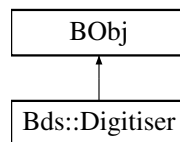
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.61 Bds::Digitiser Class Reference

This class defines a seismic [Digitiser](#).

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

Inheritance diagram for Bds::Digitiser:



### Public Member Functions

- [Digitiser](#) ( [BUInt32](#) id=0, [BTimeStamp](#) startTime= [BTimeStamp](#)(), [BTimeStamp](#) endTime= [BTime](#)↵  
[Stamp](#)(), [BString](#) name= [BString](#)(), [BString](#) type= [BString](#)(), [BString](#) serialNumber= [BString](#)(), [B](#)↵  
[UInt32](#) numberChannels=0, [BFloat64](#) baseSamplingFrequency=0, [BFloat64](#) initialSamplingFrequency=0,  
[BFloat64](#) gain=0, [BInt32](#) shared=0)
- [BString](#) [getType](#) ()
- [BError](#) [setMembers](#) ( [BDictString](#) &members)
- [BError](#) [setMember](#) ( [BString](#) name, [BString](#) value)
- [BError](#) [getMembers](#) ( [BDictString](#) &members)
- [BError](#) [getMember](#) ( [BString](#) name, [BString](#) &value)

### 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)*
- **BInt32** [shared](#)  
*This digitiser is shared.*

### 7.61.1 Detailed Description

This class defines a seismic [Digitiser](#).

This just stores information on the seismic instrument's digitiser. Its contents is generally for information only.

### 7.61.2 Constructor & Destructor Documentation

#### 7.61.2.1 Digitiser()

```
Bds::Digitiser::Digitiser (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString name = BString(),
    BString type = BString(),
    BString serialNumber = BString(),
    BUInt32 numberChannels = 0,
    BFloat64 baseSamplingFrequency = 0,
    BFloat64 initialSamplingFrequency = 0,
    BFloat64 gain = 0,
    BInt32 shared = 0 )
```

### 7.61.3 Member Function Documentation

#### 7.61.3.1 getMember()

```
BError Bds::Digitiser::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.



### 7.61.3.2 getMembers()

```
BError Bds::Digitiser::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.61.3.3 getType()

```
BString Bds::Digitiser::getType ( )
```

### 7.61.3.4 setMember()

```
BError Bds::Digitiser::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

### 7.61.3.5 setMembers()

```
BError Bds::Digitiser::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

## 7.61.4 Member Data Documentation

### 7.61.4.1 baseSamplingFrequency

```
BFloat64 Bds::Digitiser::baseSamplingFrequency
```

The base sampling frequency.



#### 7.61.4.2 endTime

**BTimeStamp** Bds::Digitiser::endTime

The End Time the channel was available.

#### 7.61.4.3 gain

**BFloat64** Bds::Digitiser::gain

The overall gain of the digitiser at the manufacturers calibration frequency. (For information only)

#### 7.61.4.4 id

**BUInt32** Bds::Digitiser::id

The ID.

#### 7.61.4.5 initialSamplingFrequency

**BFloat64** Bds::Digitiser::initialSamplingFrequency

The initial pre-decimation sampling frequency.

#### 7.61.4.6 name

**BString** Bds::Digitiser::name

The Digitisers name.

#### 7.61.4.7 numberChannels

**BUInt32** Bds::Digitiser::numberChannels

The number of supported channels.



#### 7.61.4.8 serialNumber

**BString** Bds::Digitiser::serialNumber

The digitisers's serial number.

#### 7.61.4.9 shared

**Int32** Bds::Digitiser::shared

This digitiser is shared.

#### 7.61.4.10 startTime

**TimeStamp** Bds::Digitiser::startTime

The Start Time.

#### 7.61.4.11 type

**BString** Bds::Digitiser::type

The Digitisers type.

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

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

## 7.62 Bds::Fap Class Reference

This class defines an entry in an Amplitude/Phase [Response](#) table.

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

### Public Member Functions

- [Fap](#) ( **BFloat64** frequency=0, **BFloat64** amplitude=0, **BFloat64** phase=0)



## Public Attributes

- **BFloat64** [frequency](#)  
*The frequency.*
- **BFloat64** [amplitude](#)  
*The Amplitude.*
- **BFloat64** [phase](#)  
*The Phase.*

### 7.62.1 Detailed Description

This class defines an entry in an Amplitude/Phase [Response](#) table.

### 7.62.2 Constructor & Destructor Documentation

#### 7.62.2.1 Fap()

```
Bds::Fap::Fap (  
    BFloat64 frequency = 0,  
    BFloat64 amplitude = 0,  
    BFloat64 phase = 0 )
```

### 7.62.3 Member Data Documentation

#### 7.62.3.1 amplitude

```
BFloat64 Bds::Fap::amplitude
```

The Amplitude.

#### 7.62.3.2 frequency

```
BFloat64 Bds::Fap::frequency
```

The frequency.



## 7.62.3.3 phase

```
BFloat64 Bds::Fap::phase
```

The Phase.

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

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

## 7.63 Bds::Fir Class Reference

This class defines an FIR response table.

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

### Public Member Functions

- [Fir](#) ( [BArray](#)< [FirEntry](#) > [b](#)= [BArray](#)< [FirEntry](#) >(), [BArray](#)< [FirEntry](#) > [a](#)= [BArray](#)< [FirEntry](#) >() )

### Public Attributes

- [BArray](#)< [FirEntry](#) > [b](#)  
*Numerator.*
- [BArray](#)< [FirEntry](#) > [a](#)  
*Denominator.*

### 7.63.1 Detailed Description

This class defines an FIR response table.

This has an array of the A and B coefficients.

### 7.63.2 Constructor & Destructor Documentation

#### 7.63.2.1 Fir()

```
Bds::Fir::Fir (
    BArray< FirEntry > b = BArray<FirEntry >(),
    BArray< FirEntry > a = BArray<FirEntry >() )
```



### 7.63.3 Member Data Documentation

#### 7.63.3.1 a

**BArray**<[FirEntry](#) > Bds::Fir::a

Denominator.

#### 7.63.3.2 b

**BArray**<[FirEntry](#) > Bds::Fir::b

Numerator.

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

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

## 7.64 Bds::FirEntry Class Reference

This class defines an entry in a FIR coefficient table.

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

### Public Member Functions

- [FirEntry](#) ( **BFloat64** [coefficient](#)=0, **BFloat64** [error](#)=0)

### Public Attributes

- **BFloat64** [coefficient](#)  
*Value.*
- **BFloat64** [error](#)  
*Error.*

#### 7.64.1 Detailed Description

This class defines an entry in a FIR coefficient table.



## 7.64.2 Constructor & Destructor Documentation

### 7.64.2.1 FirEntry()

```
Bds::FirEntry::FirEntry (
    BFloat64 coefficient = 0,
    BFloat64 error = 0 )
```

## 7.64.3 Member Data Documentation

### 7.64.3.1 coefficient

```
BFloat64 Bds::FirEntry::coefficient
```

Value.

### 7.64.3.2 error

```
BFloat64 Bds::FirEntry::error
```

Error.

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

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

## 7.65 Bds::GcfChannel Struct Reference

[DataFileGcf](#) internal GCF channel information.

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

### Public Attributes

- BString [systemId](#)
- BString [streamId](#)
- BUInt [type](#)
- BUInt [sampleRate](#)
- BUInt [format](#)
- BUInt32 [channel](#)



### 7.65.1 Detailed Description

[DataFileGcf](#) internal GCF channel information.

### 7.65.2 Member Data Documentation

#### 7.65.2.1 channel

**BUInt32** Bds::GcfChannel::channel

#### 7.65.2.2 format

**BUInt** Bds::GcfChannel::format

#### 7.65.2.3 sampleRate

**BUInt** Bds::GcfChannel::sampleRate

#### 7.65.2.4 streamId

**BString** Bds::GcfChannel::streamId

#### 7.65.2.5 systemId

**BString** Bds::GcfChannel::systemId

#### 7.65.2.6 type

**BUInt** Bds::GcfChannel::type

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

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

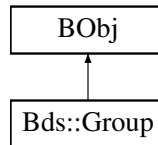


## 7.66 Bds::Group Class Reference

This holds information on a user security group.

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

Inheritance diagram for Bds::Group:



### Public Member Functions

- **Group** ( **BUInt32** id=0, **BString** group= **BString**(), **BString** description= **BString**())
- **BString** getType ()
- **BError** setMembers ( **BDictString** &members)
- **BError** setMember ( **BString** name, **BString** value)
- **BError** getMembers ( **BDictString** &members)
- **BError** getMember ( **BString** name, **BString** &value)

### Public Attributes

- **BUInt32** id  
*The unique id.*
- **BString** group  
*The **Group** name.*
- **BString** description  
*The Groups description.*

### 7.66.1 Detailed Description

This holds information on a user security group.

The BDS has the concept of a security group that users can belong to. This class defines that security group as stored in the database.

### 7.66.2 Constructor & Destructor Documentation

#### 7.66.2.1 Group()

```

Bds::Group::Group (
    BUInt32 id = 0,
    BString group = BString(),
    BString description = BString() )
  
```



### 7.66.3 Member Function Documentation

#### 7.66.3.1 getMember()

```
BError Bds::Group::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.66.3.2 getMembers()

```
BError Bds::Group::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

#### 7.66.3.3 getType()

```
BString Bds::Group::getType ( )
```

#### 7.66.3.4 setMember()

```
BError Bds::Group::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.66.3.5 setMembers()

```
BError Bds::Group::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.



## 7.66.4 Member Data Documentation

### 7.66.4.1 description

**BString** Bds::Group::description

The Groups description.

### 7.66.4.2 group

**BString** Bds::Group::group

The [Group](#) name.

### 7.66.4.3 id

**BUInt32** Bds::Group::id

The unique id.

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

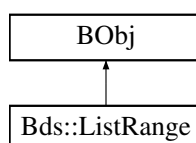
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.67 Bds::ListRange Class Reference

This class defines an integer based range.

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

Inheritance diagram for Bds::ListRange:





## Public Member Functions

- `ListRange` ( `BUInt32 start=0`, `BUInt32 number=0`, `BInt32 reverse=0`)
- `BString getType` ()
- `BError setMembers` ( `BDictString &members`)
- `BError setMember` ( `BString name`, `BString value`)
- `BError getMembers` ( `BDictString &members`)
- `BError getMember` ( `BString name`, `BString &value`)

## Public Attributes

- `BUInt32 start`  
*The start position.*
- `BUInt32 number`  
*The number of items.*
- `BInt32 reverse`  
*List from end.*

### 7.67.1 Detailed Description

This class defines an integer based range.

It is used for limit the number of items returned in selections etc.

### 7.67.2 Constructor & Destructor Documentation

#### 7.67.2.1 ListRange()

```
Bds::ListRange::ListRange (
    BUInt32 start = 0,
    BUInt32 number = 0,
    BInt32 reverse = 0 )
```

### 7.67.3 Member Function Documentation

#### 7.67.3.1 getMember()

```
BError Bds::ListRange::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from `BObj`.



### 7.67.3.2 getMembers()

```
BError Bds::ListRange::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.67.3.3 getType()

```
BString Bds::ListRange::getType ( )
```

### 7.67.3.4 setMember()

```
BError Bds::ListRange::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

### 7.67.3.5 setMembers()

```
BError Bds::ListRange::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

## 7.67.4 Member Data Documentation

### 7.67.4.1 number

```
BUInt32 Bds::ListRange::number
```

The number of items.



#### 7.67.4.2 reverse

**BInt32** Bds::ListRange::reverse

List from end.

#### 7.67.4.3 start

**BUInt32** Bds::ListRange::start

The start position.

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

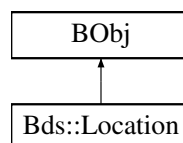
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.68 Bds::Location Class Reference

This class defines the physical location of a [Station](#).

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

Inheritance diagram for Bds::Location:



### Public Member Functions

- [Location](#) ( **BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTime**↵  
Stamp(), **BString** network= **BString**(), **BString** station= **BString**(), **BString** datum= **BString**(), **B**↵  
**Float64** longitude=0, **BFloat64** latitude=0, **BFloat64** elevation=0, **BFloat64** arrayOffsetEast=0, **BFloat64**  
arrayOffsetNorth=0)
- **BString** getType ()
- **BError** setMembers ( **BDictString** &members)
- **BError** setMember ( **BString** name, **BString** value)
- **BError** getMembers ( **BDictString** &members)
- **BError** getMember ( **BString** name, **BString** &value)



## 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.*

### 7.68.1 Detailed Description

This class defines the physical location of a [Station](#).

This defines the physical location of the station using WGS84 longitude and latitude parameters. It also defines the stations elevation and if part of a seismic array, the offset with respect to the arrays centre location.

### 7.68.2 Constructor & Destructor Documentation

#### 7.68.2.1 Location()

```
Bds::Location::Location (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString network = BString(),
    BString station = BString(),
    BString datum = BString(),
    BFloat64 longitude = 0,
    BFloat64 latitude = 0,
    BFloat64 elevation = 0,
    BFloat64 arrayOffsetEast = 0,
    BFloat64 arrayOffsetNorth = 0 )
```



## 7.68.3 Member Function Documentation

### 7.68.3.1 getMember()

```
BError Bds::Location::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

### 7.68.3.2 getMembers()

```
BError Bds::Location::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.68.3.3 getType()

```
BString Bds::Location::getType ( )
```

### 7.68.3.4 setMember()

```
BError Bds::Location::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

### 7.68.3.5 setMembers()

```
BError Bds::Location::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.



## 7.68.4 Member Data Documentation

### 7.68.4.1 arrayOffsetEast

**BFloat64** Bds::Location::arrayOffsetEast

The Array offset in in an array in an easterly direction.

### 7.68.4.2 arrayOffsetNorth

**BFloat64** Bds::Location::arrayOffsetNorth

The Array offset in in an array in a northerly direction.

### 7.68.4.3 datum

**BString** Bds::Location::datum

The locations Datum.

### 7.68.4.4 elevation

**BFloat64** Bds::Location::elevation

The ground level elevation in meters from the WGS84 ellipsoid (Sea level)

### 7.68.4.5 endTime

**BTimeStamp** Bds::Location::endTime

The End Time the channel was available.



#### 7.68.4.6 id

**BUInt32** Bds::Location::id

The ID.

#### 7.68.4.7 latitude

**BFloat64** Bds::Location::latitude

The Latitude in degrees using the WGS84 datum.

#### 7.68.4.8 longitude

**BFloat64** Bds::Location::longitude

The longitude in degrees using the WGS84 datum.

#### 7.68.4.9 network

**BString** Bds::Location::network

The Network/Organisation Name.

#### 7.68.4.10 startTime

**BTimeStamp** Bds::Location::startTime

The Start Time.

#### 7.68.4.11 station

**BString** Bds::Location::station

The station this location is for.

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

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

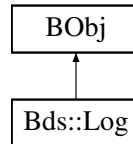


## 7.69 Bds::Log Class Reference

This holds information on a [Log](#) entry.

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

Inheritance diagram for Bds::Log:



### Public Member Functions

- [Log](#) ( [BUInt32](#) id=0, [BTimeStamp](#) time= [BTimeStamp](#)(), [BString](#) type= [BString](#)(), [BUInt32](#) priority=0, [BString](#) subSystem= [BString](#)(), [BString](#) title= [BString](#)(), [BString](#) description= [BString](#)())
- [BString](#) getType ()
- [BError](#) setMembers ( [BDictString](#) &members)
- [BError](#) setMember ( [BString](#) name, [BString](#) value)
- [BError](#) getMembers ( [BDictString](#) &members)
- [BError](#) getMember ( [BString](#) name, [BString](#) &value)

### Public Attributes

- [BUInt32](#) id  
*The unique id.*
- [BTimeStamp](#) time  
*The Time.*
- [BString](#) type  
*The Type.*
- [BUInt32](#) priority  
*The priority 0 to 5.*
- [BString](#) subSystem  
*The SubSystem.*
- [BString](#) title  
*The Changes title.*
- [BString](#) description  
*The Description of the change.*

#### 7.69.1 Detailed Description

This holds information on a [Log](#) entry.

[Log](#) entries are added automatically and manually to the system. A system administrator can view these logs.



## 7.69.2 Constructor & Destructor Documentation

### 7.69.2.1 Log()

```
Bds::Log::Log (
    BUInt32 id = 0,
    BTimeStamp time = BTimeStamp(),
    BString type = BString(),
    BUInt32 priority = 0,
    BString subSystem = BString(),
    BString title = BString(),
    BString description = BString() )
```

## 7.69.3 Member Function Documentation

### 7.69.3.1 getMember()

```
BError Bds::Log::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

### 7.69.3.2 getMembers()

```
BError Bds::Log::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.69.3.3 getType()

```
BString Bds::Log::getType ( )
```



#### 7.69.3.4 setMember()

```
BError Bds::Log::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.69.3.5 setMembers()

```
BError Bds::Log::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.69.4 Member Data Documentation

#### 7.69.4.1 description

```
BString Bds::Log::description
```

The Description of the change.

#### 7.69.4.2 id

```
BUInt32 Bds::Log::id
```

The unique id.

#### 7.69.4.3 priority

```
BUInt32 Bds::Log::priority
```

The priority 0 to 5.



#### 7.69.4.4 subSystem

**BString** Bds::Log::subSystem

The SubSystem.

#### 7.69.4.5 time

**BTimeStamp** Bds::Log::time

The Time.

#### 7.69.4.6 title

**BString** Bds::Log::title

The Changes title.

#### 7.69.4.7 type

**BString** Bds::Log::type

The Type.

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

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

## 7.70 Bds::LogSelect Class Reference

This defines the selection cirteria when requesting a set of log entries.

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

### Public Member Functions

- [LogSelect](#) ( **BTimeStamp** startTime= **BTimeStamp**(), **BString** type= **BString**(), **BUInt32** priority=0, **BString** subSystem= **BString**())



## Public Attributes

- **BTimeStamp** [startTime](#)  
*The start time.*
- **BString** [type](#)  
*The Type.*
- **BUInt32** [priority](#)  
*The priority 0 to 5.*
- **BString** [subSystem](#)  
*The SubSystem.*

### 7.70.1 Detailed Description

This defines the selection criteria when requesting a set of log entries.

### 7.70.2 Constructor & Destructor Documentation

#### 7.70.2.1 LogSelect()

```
Bds::LogSelect::LogSelect (
    BTimeStamp startTime = BTimeStamp(),
    BString type = BString(),
    BUInt32 priority = 0,
    BString subSystem = BString() )
```

### 7.70.3 Member Data Documentation

#### 7.70.3.1 priority

**BUInt32** Bds::LogSelect::priority

The priority 0 to 5.

#### 7.70.3.2 startTime

**BTimeStamp** Bds::LogSelect::startTime

The start time.



### 7.70.3.3 subSystem

**BString** Bds::LogSelect::subSystem

The SubSystem.

### 7.70.3.4 type

**BString** Bds::LogSelect::type

The Type.

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

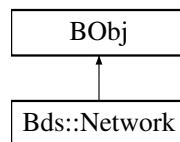
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.71 Bds::Network Class Reference

This class defines a seismic [Network](#) organisation.

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

Inheritance diagram for Bds::Network:



### Public Member Functions

- [Network](#) ( **BUInt32** id=0, **BString** network= **BString**(), **BString** description= **BString**(), **BList**< **BString** > stations= **BList**< **BString** >())
- **BString** [getType](#) ()
- **BError** [setMembers](#) ( **BDictString** &members)
- **BError** [setMember](#) ( **BString** name, **BString** value)
- **BError** [getMembers](#) ( **BDictString** &members)
- **BError** [getMember](#) ( **BString** name, **BString** &value)

### Public Attributes

- **BUInt32** [id](#)  
*Unique ID when stored in a database or for other uses.*
- **BString** [network](#)  
*The name.*
- **BString** [description](#)  
*The organisations description.*
- **BList**< **BString** > [stations](#)  
*The list of arrays/stations the [Network](#) uses.*



### 7.71.1 Detailed Description

This class defines a seismic [Network](#) organisation.

Typical Seismic Networks are "BN", IDC" etc.

### 7.71.2 Constructor & Destructor Documentation

#### 7.71.2.1 Network()

```
Bds::Network::Network (
    BUInt32 id = 0,
    BString network = BString(),
    BString description = BString(),
    BList< BString > stations = BList< BString >() )
```

### 7.71.3 Member Function Documentation

#### 7.71.3.1 getMember()

```
BError Bds::Network::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.71.3.2 getMembers()

```
BError Bds::Network::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

#### 7.71.3.3 getType()

```
BString Bds::Network::getType ( )
```



#### 7.71.3.4 setMember()

```
BError Bds::Network::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.71.3.5 setMembers()

```
BError Bds::Network::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.71.4 Member Data Documentation

#### 7.71.4.1 description

```
BString Bds::Network::description
```

The organisations description.

#### 7.71.4.2 id

```
BUInt32 Bds::Network::id
```

Unique ID when stored in a database or for other uses.

#### 7.71.4.3 network

```
BString Bds::Network::network
```

The name.



## 7.71.4.4 stations

```
BList< BString > Bds::Network::stations
```

The list of arrays/stations the [Network](#) uses.

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

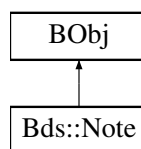
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.72 Bds::Note Class Reference

This holds information on a [Note](#) for general information.

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

Inheritance diagram for Bds::Note:



### Public Member Functions

- [Note](#) ( **BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), **BString** network= **BString**(), **BString** station= **BString**(), **BString** channel= **BString**(), **BString** source= **BString**(), **BString** type= **BString**(), **BString** user= **BString**(), **BTimeStamp** timeAdded= **BTimeStamp**(), **BInt32** errorNumber=0, **BString** title= **BString**(), **BString** description= **BString**(), **BString** docFormat= **BString**(), **BString** docUrl= **BString**(), **BUInt32** dataFileId=0, **BString** importFilename= **BString**())
- **BString** getType ()
- **BError** setMembers ( **BDictString** &members)
- **BError** setMember ( **BString** name, **BString** value)
- **BError** getMembers ( **BDictString** &members)
- **BError** getMember ( **BString** name, **BString** &value)

### Public Attributes

- **BUInt32** id  
*The unique id.*
- **BTimeStamp** startTime  
*The Start Time note is for.*
- **BTimeStamp** endTime  
*The End Time note is for.*
- **BString** network  
*The [Network](#) Name.*
- **BString** station



- The Station/Array name.*
  - **BString** [channel](#)
- The Channels name.*
  - **BString** [source](#)
- The Data [Source](#).*
  - **BString** [type](#)
- The Type (note, warning, error ...)*
  - **BString** [user](#)
- The user.*
  - **BTimeStamp** [timeAdded](#)
- The Time Entered.*
  - **BInt32** [errorNumber](#)
- Error number if error.*
  - **BString** [title](#)
- The title.*
  - **BString** [description](#)
- The Description.*
  - **BString** [docFormat](#)
- Document format if any.*
  - **BString** [docUrl](#)
- Document Url if any.*
  - **BUInt32** [dataFileId](#)
- The data file id associated with this note.*
  - **BString** [importFilename](#)
- The import filename.*

### 7.72.1 Detailed Description

This holds information on a [Note](#) for general information.

Normally a [Note](#) can be added for a particular set of data over a particular time period. These notes are sometimes added automatically during a data import process or by a user. A data user can then ask for any notes for a particular set of data.

### 7.72.2 Constructor & Destructor Documentation

#### 7.72.2.1 [Note\(\)](#)

```
Bds::Note::Note (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString network = BString(),
    BString station = BString(),
    BString channel = BString(),
    BString source = BString(),
    BString type = BString(),
```



```
BString user = BString(),
BTimeStamp timeAdded = BTimeStamp(),
BInt32 errorNumber = 0,
BString title = BString(),
BString description = BString(),
BString docFormat = BString(),
BString docUrl = BString(),
BUInt32 dataFileId = 0,
BString importFilename = BString() )
```

### 7.72.3 Member Function Documentation

#### 7.72.3.1 getMember()

```
BError Bds::Note::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.72.3.2 getMembers()

```
BError Bds::Note::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

#### 7.72.3.3 getType()

```
BString Bds::Note::getType ( )
```

#### 7.72.3.4 setMember()

```
BError Bds::Note::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.



#### 7.72.3.5 setMembers()

```
BError Bds::Note::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.72.4 Member Data Documentation

#### 7.72.4.1 channel

```
BString Bds::Note::channel
```

The Channels name.

#### 7.72.4.2 dataFileId

```
BUInt32 Bds::Note::dataFileId
```

The data file id associated with this note.

#### 7.72.4.3 description

```
BString Bds::Note::description
```

The Description.

#### 7.72.4.4 docFormat

```
BString Bds::Note::docFormat
```

Document format if any.

#### 7.72.4.5 docUrl

```
BString Bds::Note::docUrl
```

Document Url if any.



#### 7.72.4.6 endTime

**BTimeStamp** Bds::Note::endTime

The End Time note is for.

#### 7.72.4.7 errorNumber

**BInt32** Bds::Note::errorNumber

Error number if error.

#### 7.72.4.8 id

**BUInt32** Bds::Note::id

The unique id.

#### 7.72.4.9 importFilename

**BString** Bds::Note::importFilename

The import filename.

#### 7.72.4.10 network

**BString** Bds::Note::network

The [Network](#) Name.

#### 7.72.4.11 source

**BString** Bds::Note::source

The Data [Source](#).



#### 7.72.4.12 `startTime`

**BTimeStamp** `Bds::Note::startTime`

The Start Time note is for.

#### 7.72.4.13 `station`

**BString** `Bds::Note::station`

The Station/Array name.

#### 7.72.4.14 `timeAdded`

**BTimeStamp** `Bds::Note::timeAdded`

The Time Entered.

#### 7.72.4.15 `title`

**BString** `Bds::Note::title`

The title.

#### 7.72.4.16 `type`

**BString** `Bds::Note::type`

The Type (note, warning, error ...)

#### 7.72.4.17 `user`

**BString** `Bds::Note::user`

The user.

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

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



## 7.73 Bds::Point Class Reference

This class defines an X,Y location.

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

### Public Member Functions

- [Point](#) ( [BFloat64](#) x=0, [BFloat64](#) y=0)

### Public Attributes

- [BFloat64](#) [x](#)
- [BFloat64](#) [y](#)

### 7.73.1 Detailed Description

This class defines an X,Y location.

The class simply stores the x and y point values.

### 7.73.2 Constructor & Destructor Documentation

#### 7.73.2.1 Point()

```
Bds::Point::Point (
    BFloat64 x = 0,
    BFloat64 y = 0 )
```

### 7.73.3 Member Data Documentation

#### 7.73.3.1 x

[BFloat64](#) [Bds::Point::x](#)



### 7.73.3.2 y

```
BFloat64 Bds::Point::y
```

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

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

## 7.74 Bds::PoleZero Class Reference

This class defines a Pole/Zero [Response](#).

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

### Public Member Functions

- [PoleZero](#) ( **BArray**< **BComplex** > [poles](#)= **BArray**< **BComplex** >(), **BArray**< **BComplex** > [zeros](#)= **BArray**< **BComplex** >())

### Public Attributes

- **BArray**< **BComplex** > [poles](#)  
*Poles.*
- **BArray**< **BComplex** > [zeros](#)  
*Zeros.*

### 7.74.1 Detailed Description

This class defines a Pole/Zero [Response](#).

It consists of an array of Complex poles and an array of Complex zeros.

### 7.74.2 Constructor & Destructor Documentation

#### 7.74.2.1 PoleZero()

```
Bds::PoleZero::PoleZero (
    BArray< BComplex > poles = BArray< BComplex >(),
    BArray< BComplex > zeros = BArray< BComplex >() )
```



### 7.74.3 Member Data Documentation

#### 7.74.3.1 poles

```
BArray< BComplex > Bds::PoleZero::poles
```

Poles.

#### 7.74.3.2 zeros

```
BArray< BComplex > Bds::PoleZero::zeros
```

Zeros.

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

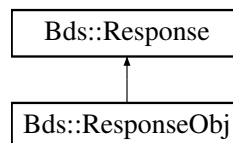
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.75 Bds::Response Class Reference

This class defines a seismic [Response](#) characteristic.

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

Inheritance diagram for Bds::Response:



### Public Member Functions

- [Response](#) ( [BUInt32](#) id=0, [BTimeStamp](#) startTime= [BTimeStamp](#)(), [BTimeStamp](#) endTime= [BTime](#)↵  
Stamp(), [BString](#) network= [BString](#)(), [BString](#) station= [BString](#)(), [BString](#) channel= [BString](#)(), [B](#)↵  
String source= [BString](#)(), [BUInt32](#) stage=0, [BString](#) name= [BString](#)(), [BString](#) type= [BString](#)(), [PoleZero](#)  
poleZeros=[PoleZero](#)(), [BArray](#)< [Fap](#) > faps= [BArray](#)< [Fap](#) >(), [Fir](#) fir=[Fir](#)(), [BFloat64](#) gain=0, [BFloat64](#)  
gainFrequency=0, [BString](#) stageType= [BString](#)(), [BFloat64](#) decimation=0, [BString](#) symmetry= [BString](#)(),  
[BString](#) description= [BString](#)(), [BInt32](#) measured=0, [BFloat64](#) sampleRate=0)



## 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** [source](#)  
*The source.*
- **BUInt32** [stage](#)  
*The stage (1, 2, 3, ...)*
- **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.*
- [Fir](#) [fir](#)  
*The FIR filters coefficients.*
- **BFloat64** [gain](#)  
*Overall gain at gain frequency. (For information)*
- **BFloat64** [gainFrequency](#)  
*Frequency that gain is valid for. (For information)*
- **BString** [stageType](#)  
*The stage type: A - Analog (rad/sec), B - Analog (Hz), C - Composite, D - Digital.*
- **BFloat64** [decimation](#)  
*Decimation performed post filter.*
- **BString** [symmetry](#)  
*Symmetry for FIR coefficients (A = asymmetric, B = symmetric[odd], C = symmetric[even]) ??*
- **BString** [description](#)  
*Misc description.*
- **BInt32** [measured](#)  
*If response was a measured response.*
- **BFloat64** [sampleRate](#)  
*The stage's sample rate.*

### 7.75.1 Detailed Description

This class defines a seismic [Response](#) characteristic.

For each seismic channel there is a frequency response characteristic. There can be multiple stages in a channels frequency response, this response data describes one of those stages frequencies responses. The stage parameter defines which stage it is for (1, 2, 3, ...) Stage 1 is reserved to store an overall channel response. A response can be in the form of an array of poles and zeros, a FAP array, or a set of FIR coefficients. This object contains members for other response characteristics as defined in various seismic response databases.



## 7.75.2 Constructor & Destructor Documentation

### 7.75.2.1 Response()

```

Bds::Response::Response (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString network = BString(),
    BString station = BString(),
    BString channel = BString(),
    BString source = BString(),
    BUInt32 stage = 0,
    BString name = BString(),
    BString type = BString(),
    PoleZero poleZeros = PoleZero(),
    BArray< Fap > faps = BArray<Fap>(),
    Fir fir = Fir(),
    BFloat64 gain = 0,
    BFloat64 gainFrequency = 0,
    BString stageType = BString(),
    BFloat64 decimation = 0,
    BString symmetry = BString(),
    BString description = BString(),
    BInt32 measured = 0,
    BFloat64 sampleRate = 0 )

```

## 7.75.3 Member Data Documentation

### 7.75.3.1 channel

**BString** Bds::Response::channel

The channel.

### 7.75.3.2 decimation

**BFloat64** Bds::Response::decimation

Decimation performed post filter.



#### 7.75.3.3 description

**BString** Bds::Response::description

Misc description.

#### 7.75.3.4 endTime

**BTimeStamp** Bds::Response::endTime

The End Time the channel was available.

#### 7.75.3.5 faps

**BArray<Fap >** Bds::Response::faps

The Frequency/Amplitude/Phase table.

#### 7.75.3.6 fir

**Fir** Bds::Response::fir

The FIR filters coefficients.

#### 7.75.3.7 gain

**BFloat64** Bds::Response::gain

Overall gain at gain frequency. (For information)

#### 7.75.3.8 gainFrequency

**BFloat64** Bds::Response::gainFrequency

Frequency that gain is valid for. (For information)



### 7.75.3.9 id

**BUInt32** Bds::Response::id

The ID.

### 7.75.3.10 measured

**BInt32** Bds::Response::measured

If response was a measured response.

### 7.75.3.11 name

**BString** Bds::Response::name

The response name. (Anti-Aliasing filter, [Digitiser](#), post filter etc)

### 7.75.3.12 network

**BString** Bds::Response::network

The Network/Organisation Name.

### 7.75.3.13 poleZeros

[PoleZero](#) Bds::Response::poleZeros

[PoleZero](#), AmplitudePhase or FIR Coefficient data.

### 7.75.3.14 sampleRate

**BFloat64** Bds::Response::sampleRate

The stage's sample rate.



#### 7.75.3.15 source

**BString** Bds::Response::source

The source.

#### 7.75.3.16 stage

**BUInt32** Bds::Response::stage

The stage (1, 2, 3, ...)

#### 7.75.3.17 stageType

**BString** Bds::Response::stageType

The stage type: A - Analog (rad/sec), B - Analog (Hz), C - Composite, D - Digital.

#### 7.75.3.18 startTime

**BTimeStamp** Bds::Response::startTime

The Start Time.

#### 7.75.3.19 station

**BString** Bds::Response::station

The station.

#### 7.75.3.20 symmetry

**BString** Bds::Response::symmetry

Symmetry for FIR coefficients (A = asymmetric, B = symmetric[odd], C = symmetric[even]) ??



## 7.75.3.21 type

**BString** Bds::Response::type

The type of response ([PoleZero](#),AmplitudePhase or FIR Coefficients)

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

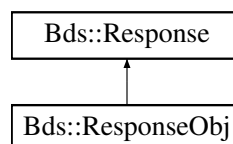
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.76 Bds::ResponseObj Class Reference

[Response](#) object adding string conversion.

```
#include <BdsLib.h>
```

Inheritance diagram for Bds::ResponseObj:



### Public Member Functions

- [ResponseObj](#) (const [Response](#) &response)
- [~ResponseObj](#) ()
- **BString** [getString](#) ()
- void [setString](#) ( **BString** str)

### Additional Inherited Members

#### 7.76.1 Detailed Description

[Response](#) object adding string conversion.

#### 7.76.2 Constructor & Destructor Documentation

##### 7.76.2.1 ResponseObj()

```
Bds::ResponseObj::ResponseObj (
    const Response & response )
```



### 7.76.2.2 ~ResponseObj()

```
Bds::ResponseObj::~~ResponseObj ( )
```

## 7.76.3 Member Function Documentation

### 7.76.3.1 getString()

```
BString Bds::ResponseObj::getString ( )
```

### 7.76.3.2 setString()

```
void Bds::ResponseObj::setString (
    BString str )
```

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

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

## 7.77 Bds::Selection Class Reference

This class defines a generic metadata or seismic data selection.

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

### Public Member Functions

- [Selection](#) ( [BUInt32](#) id=0, [ListRange](#) range=[ListRange](#)(), [BTimeStamp](#) startTime= [BTimeStamp](#)(), [BTimeStamp](#) endTime= [BTimeStamp](#)(), [BList](#)< [SelectionChannel](#) > channels= [BList](#)< [SelectionChannel](#) >(), [BUInt32](#) channelId=0, [BUInt32](#) digitiserId=0, [BUInt32](#) sensorId=0, [BUInt32](#) sensorOldId=0, [BInt32](#) completeSegments=0, [BString](#) calibrationName= [BString](#)() )



## Public Attributes

- **BUInt32** [id](#)  
*The ID of the record to return.*
- **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.*
- **BInt32** [completeSegments](#)  
*Do not clip the segment times to match the required time period.*
- **BString** [calibrationName](#)  
*[Calibration](#) name to use.*

### 7.77.1 Detailed Description

This class defines a generic metadata or seismic data selection.

This defines a set of selection criteria when selecting items from the BDS metadata or seismic data sets. The fields, when set, limit the items returned by the settings provided. In effect it has an "AND" type of function with the parameters provided. There are some specific parameters like: channelId, digitiserId, sensorId etc which are used with specific data selection functions.

### 7.77.2 Constructor & Destructor Documentation

#### 7.77.2.1 Selection()

```
Bds::Selection::Selection (
    BUInt32 id = 0,
    ListRange range = ListRange(),
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BList< SelectionChannel > channels = BList<SelectionChannel >(),
    BUInt32 channelId = 0,
    BUInt32 digitiserId = 0,
    BUInt32 sensorId = 0,
    BUInt32 sensorOldId = 0,
    BInt32 completeSegments = 0,
    BString calibrationName = BString() )
```



### 7.77.3 Member Data Documentation

#### 7.77.3.1 calibrationName

**BString** Bds::Selection::calibrationName

Calibration name to use.

#### 7.77.3.2 channelId

**BUInt32** Bds::Selection::channelId

The Channel id.

#### 7.77.3.3 channels

**BList**<SelectionChannel > Bds::Selection::channels

The data channels to select.

#### 7.77.3.4 completeSegments

**BInt32** Bds::Selection::completeSegments

Do not clip the segment times to match the required time period.

#### 7.77.3.5 digitiserId

**BUInt32** Bds::Selection::digitiserId

The Digitiser id.



#### 7.77.3.6 endTime

**BTimeStamp** Bds::Selection::endTime

The End Time.

#### 7.77.3.7 id

**BUInt32** Bds::Selection::id

The ID of the record to return.

#### 7.77.3.8 range

[ListRange](#) Bds::Selection::range

The range of data to return.

#### 7.77.3.9 sensorId

**BUInt32** Bds::Selection::sensorId

The [Sensor](#) id.

#### 7.77.3.10 sensorOldId

**BUInt32** Bds::Selection::sensorOldId

The [Sensor](#) old id.

#### 7.77.3.11 startTime

**BTimeStamp** Bds::Selection::startTime

The Start Time.

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

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



## 7.78 Bds::SelectionChannel Class Reference

This class defines a channel for selection.

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

### Public Member Functions

- [SelectionChannel](#) ( **BString** *network*= **BString**(), **BString** *station*= **BString**(), **BString** *channel*= **BString**(), **BString** *source*= **BString**() )

### Public Attributes

- **BString** *network*
- **BString** *station*
- **BString** *channel*
- **BString** *source*

### 7.78.1 Detailed Description

This class defines a channel for selection.

It contains the network:station:channel:source names.

### 7.78.2 Constructor & Destructor Documentation

#### 7.78.2.1 SelectionChannel()

```
Bds::SelectionChannel::SelectionChannel (
    BString network = BString(),
    BString station = BString(),
    BString channel = BString(),
    BString source = BString() )
```

### 7.78.3 Member Data Documentation

#### 7.78.3.1 channel

```
BString Bds::SelectionChannel::channel
```



## 7.78.3.2 network

**BString** Bds::SelectionChannel::network

## 7.78.3.3 source

**BString** Bds::SelectionChannel::source

## 7.78.3.4 station

**BString** Bds::SelectionChannel::station

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

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

## 7.79 Bds::SelectionInfo Class Reference

This class defines the set of metadata or seismic data selected when getSelectionInfo() is use.

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

## Public Member Functions

- **SelectionInfo** ( **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), **BList**< **BString** > networks= **BList**< **BString** >(), **BList**< **BString** > arrays= **BList**< **BString** >(), **BList**< **BString** > stations= **BList**< **BString** >(), **BList**< **BString** > arraysAndStations= **BList**< **BString** >(), **BList**< **BString** > channels= **BList**< **BString** >(), **BList**< **BString** > sources= **BList**< **BString** >(), **BUInt32** numDataChannels=0)

## Public Attributes

- **BTimeStamp** startTime  
*The Start Time.*
- **BTimeStamp** endTime  
*The End Time.*
- **BList**< **BString** > networks  
*The list of [Network](#) Names.*
- **BList**< **BString** > arrays  
*The list of Array names.*
- **BList**< **BString** > stations  
*The list of [Station](#) names.*
- **BList**< **BString** > arraysAndStations  
*The list of Array and [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 matching the criteria.*



### 7.79.1 Detailed Description

This class defines the set of metadata or seismic data selected when `getSelectionInfo()` is use.

This provides information on everything selected by a [Selection](#) object from the BDS metadata or seismic data sets.

### 7.79.2 Constructor & Destructor Documentation

#### 7.79.2.1 SelectionInfo()

```
Bds::SelectionInfo::SelectionInfo (
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BList< BString > networks = BList< BString >(),
    BList< BString > arrays = BList< BString >(),
    BList< BString > stations = BList< BString >(),
    BList< BString > arraysAndStations = BList< BString >(),
    BList< BString > channels = BList< BString >(),
    BList< BString > sources = BList< BString >(),
    BUInt32 numDataChannels = 0 )
```

### 7.79.3 Member Data Documentation

#### 7.79.3.1 arrays

```
BList< BString > Bds::SelectionInfo::arrays
```

The list of Array names.

#### 7.79.3.2 arraysAndStations

```
BList< BString > Bds::SelectionInfo::arraysAndStations
```

The list of Array and [Station](#) names.

#### 7.79.3.3 channels

```
BList< BString > Bds::SelectionInfo::channels
```

The list of Channels.



#### 7.79.3.4 endTime

**BTimeStamp** Bds::SelectionInfo::endTime

The End Time.

#### 7.79.3.5 networks

**BList**< **BString** > Bds::SelectionInfo::networks

The list of [Network](#) Names.

#### 7.79.3.6 numDataChannels

**BUInt32** Bds::SelectionInfo::numDataChannels

The number of sets of data in the system matching the criteria.

#### 7.79.3.7 sources

**BList**< **BString** > Bds::SelectionInfo::sources

The list of Data Sources.

#### 7.79.3.8 startTime

**BTimeStamp** Bds::SelectionInfo::startTime

The Start Time.

#### 7.79.3.9 stations

**BList**< **BString** > Bds::SelectionInfo::stations

The list of [Station](#) names.

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

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

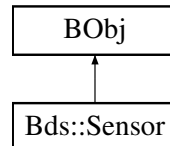


## 7.80 Bds::Sensor Class Reference

This class defines a seismic [Sensor](#).

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

Inheritance diagram for Bds::Sensor:



### Public Member Functions

- [Sensor](#) ( [BUInt32](#) [id](#)=0, [BTimeStamp](#) [startTime](#)= [BTimeStamp](#)(), [BTimeStamp](#) [endTime](#)= [BTime](#)↵  
[Stamp](#)(), [BString](#) [name](#)= [BString](#)(), [BString](#) [type](#)= [BString](#)(), [BString](#) [serialNumber](#)= [BString](#)(), [BUInt32](#)  
[numberChannels](#)=0, [BString](#) [gainUnits](#)= [BString](#)(), [BFloat64](#) [gain](#)=0, [BUInt32](#) [oldId](#)=0, [BInt32](#) [shared](#)=0)
- [BString](#) [getType](#) ()
- [BError](#) [setMembers](#) ( [BDictString](#) &members)
- [BError](#) [setMember](#) ( [BString](#) [name](#), [BString](#) [value](#))
- [BError](#) [getMembers](#) ( [BDictString](#) &members)
- [BError](#) [getMember](#) ( [BString](#) [name](#), [BString](#) &value)

### 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.*
- [BInt32](#) [shared](#)  
*This sensor is shared.*



### 7.80.1 Detailed Description

This class defines a seismic [Sensor](#).

This just stores information on the seismic instrument's sensor. Its contents is generally for information only.

### 7.80.2 Constructor & Destructor Documentation

#### 7.80.2.1 Sensor()

```
Bds::Sensor::Sensor (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString name = BString(),
    BString type = BString(),
    BString serialNumber = BString(),
    BUInt32 numberChannels = 0,
    BString gainUnits = BString(),
    BFloat64 gain = 0,
    BUInt32 oldId = 0,
    BInt32 shared = 0 )
```

### 7.80.3 Member Function Documentation

#### 7.80.3.1 getMember()

```
BError Bds::Sensor::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.80.3.2 getMembers()

```
BError Bds::Sensor::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.



### 7.80.3.3 getType()

```
BString Bds::Sensor::getType ( )
```

### 7.80.3.4 setMember()

```
BError Bds::Sensor::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

### 7.80.3.5 setMembers()

```
BError Bds::Sensor::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

## 7.80.4 Member Data Documentation

### 7.80.4.1 endTime

```
BTimeStamp Bds::Sensor::endTime
```

The End Time.

### 7.80.4.2 gain

```
BFloat64 Bds::Sensor::gain
```

The overall gain of the sensor at the manufacturers calibration frequency. (For information only)

### 7.80.4.3 gainUnits

```
BString Bds::Sensor::gainUnits
```

The gain units.



#### 7.80.4.4 id

**BUInt32** Bds::Sensor::id

The ID.

#### 7.80.4.5 name

**BString** Bds::Sensor::name

The Sensors name.

#### 7.80.4.6 numberChannels

**BUInt32** Bds::Sensor::numberChannels

The number of supported channels.

#### 7.80.4.7 oldId

**BUInt32** Bds::Sensor::oldId

The Id from the old Autodrm database.

#### 7.80.4.8 serialNumber

**BString** Bds::Sensor::serialNumber

The sensor's serial number. Only used when there is a unique physical sensor.

#### 7.80.4.9 shared

**BInt32** Bds::Sensor::shared

This sensor is shared.



#### 7.80.4.10 `startTime`

**BTimeStamp** `Bds::Sensor::startTime`

The Start Time.

#### 7.80.4.11 `type`

**BString** `Bds::Sensor::type`

The type of sensor. (Seismometer, Hydrophone etc)

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

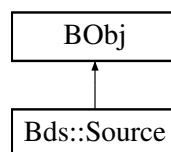
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.81 `Bds::Source` Class Reference

This class defines a seismic data [Source](#).

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

Inheritance diagram for `Bds::Source`:



### Public Member Functions

- [Source](#) ( **BUInt32** `id`=0, **BString** `source`= **BString**(), **BString** `sourceMeta`= **BString**(), **BString** `alias`= **BString**(), **BString** `description`= **BString**() )
- **BString** [getType](#) ()
- **BError** [setMembers](#) ( **BDictString** &members)
- **BError** [setMember](#) ( **BString** name, **BString** value)
- **BError** [getMembers](#) ( **BDictString** &members)
- **BError** [getMember](#) ( **BString** name, **BString** &value)



## Public Attributes

- **BUInt32** [id](#)  
*Unique ID when stored in a database or for other uses.*
- **BString** [source](#)  
*The sensor data's source name.*
- **BString** [sourceMeta](#)  
*The associated metadata's source name.*
- **BString** [alias](#)  
*The short alias for data files.*
- **BString** [description](#)  
*The description.*

### 7.81.1 Detailed Description

This class defines a seismic data [Source](#).

A Seismic data source allows different sources of data to be described and allows different Metadata sets to be used with the different data sources. It might be that there were two different digitisers in use or one data set was received real-time though a particular data processing chain while the other was via CD medium with a different processing chain.

### 7.81.2 Constructor & Destructor Documentation

#### 7.81.2.1 Source()

```
Bds::Source::Source (
    BUInt32 id = 0,
    BString source = BString(),
    BString sourceMeta = BString(),
    BString alias = BString(),
    BString description = BString() )
```

### 7.81.3 Member Function Documentation

#### 7.81.3.1 getMember()

```
BError Bds::Source::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.



#### 7.81.3.2 getMembers()

```
BError Bds::Source::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

#### 7.81.3.3 getType()

```
BString Bds::Source::getType ( )
```

#### 7.81.3.4 setMember()

```
BError Bds::Source::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

#### 7.81.3.5 setMembers()

```
BError Bds::Source::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.81.4 Member Data Documentation

#### 7.81.4.1 alias

```
BString Bds::Source::alias
```

The short alias for data files.



## 7.81.4.2 description

```
BString Bds::Source::description
```

The description.

## 7.81.4.3 id

```
BUInt32 Bds::Source::id
```

Unique ID when stored in a database or for other uses.

## 7.81.4.4 source

```
BString Bds::Source::source
```

The sensor data's source name.

## 7.81.4.5 sourceMeta

```
BString Bds::Source::sourceMeta
```

The associated metadata's source name.

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

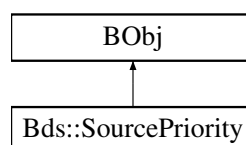
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.82 Bds::SourcePriority Class Reference

This class defines a [Source](#) Priority entry.

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

Inheritance diagram for Bds::SourcePriority:





## Public Member Functions

- [SourcePriority](#) ( [BUInt32](#) id=0, [BTimeStamp](#) startTime= [BTimeStamp](#)(), [BTimeStamp](#) endTime= [BTimeStamp](#)(), [BString](#) source= [BString](#)(), [BUInt32](#) priority=0)
- [BString](#) [getType](#) ()
- [BError](#) [setMembers](#) ( [BDictString](#) &members)
- [BError](#) [setMember](#) ( [BString](#) name, [BString](#) value)
- [BError](#) [getMembers](#) ( [BDictString](#) &members)
- [BError](#) [getMember](#) ( [BString](#) name, [BString](#) &value)

## Public Attributes

- [BUInt32](#) [id](#)  
*Unique ID when stored in a database or for other uses.*
- [BTimeStamp](#) [startTime](#)  
*The Start Time.*
- [BTimeStamp](#) [endTime](#)  
*The End Time the channel was available.*
- [BString](#) [source](#)  
*The source name.*
- [BUInt32](#) [priority](#)  
*The priority order, highest first.*

### 7.82.1 Detailed Description

This class defines a [Source](#) Priority entry.

This allows the default source for data to be selected based on a priority level. It allows a particular source to be used if no other is available and then prioritised through all the different sources available.

### 7.82.2 Constructor & Destructor Documentation

#### 7.82.2.1 [SourcePriority](#)()

```
Bds::SourcePriority::SourcePriority (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString source = BString(),
    BUInt32 priority = 0 )
```

#### 7.82.3 Member Function Documentation



### 7.82.3.1 getMember()

```
BError Bds::SourcePriority::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

### 7.82.3.2 getMembers()

```
BError Bds::SourcePriority::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.82.3.3 getType()

```
BString Bds::SourcePriority::getType ( )
```

### 7.82.3.4 setMember()

```
BError Bds::SourcePriority::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

### 7.82.3.5 setMembers()

```
BError Bds::SourcePriority::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

## 7.82.4 Member Data Documentation



#### 7.82.4.1 endTime

**BTimeStamp** Bds::SourcePriority::endTime

The End Time the channel was available.

#### 7.82.4.2 id

**BUInt32** Bds::SourcePriority::id

Unique ID when stored in a database or for other uses.

#### 7.82.4.3 priority

**BUInt32** Bds::SourcePriority::priority

The priority order, highest first.

#### 7.82.4.4 source

**BString** Bds::SourcePriority::source

The source name.

#### 7.82.4.5 startTime

**BTimeStamp** Bds::SourcePriority::startTime

The Start Time.

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

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

## 7.83 Bds::Station Class Reference

This class defines a seismic station.

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



## Public Member Functions

- [Station](#) ( [BUInt32](#) *id*=0, [BString](#) *name*= [BString](#)(), [BString](#) *alias*= [BString](#)(), [BString](#) *type*= [BString](#)(), [BString](#) *description*= [BString](#)(), [BList](#)< [ArrayChannel](#) > *channels*= [BList](#)< [ArrayChannel](#) >())

## Public Attributes

- [BUInt32](#) *id*  
*Unique ID when stored in a database or for other uses.*
- [BString](#) *name*  
*The name.*
- [BString](#) *alias*  
*Alias name to be returned to the user.*
- [BString](#) *type*  
*The [Station](#) type. Set to "array" or "station".*
- [BString](#) *description*  
*Description.*
- [BList](#)< [ArrayChannel](#) > *channels*  
*List of channels if an Array.*

### 7.83.1 Detailed Description

This class defines a seismic station.

A [Station](#) could be an individual station or an array of stations. If it is an array of stations the [Station](#) object contains the list of chaannels that make up the array. This class defines a [Station](#)

### 7.83.2 Constructor & Destructor Documentation

#### 7.83.2.1 Station()

```
Bds::Station::Station (
    BUInt32 id = 0,
    BString name = BString(),
    BString alias = BString(),
    BString type = BString(),
    BString description = BString(),
    BList< ArrayChannel > channels = BList<ArrayChannel >() )
```

### 7.83.3 Member Data Documentation



#### 7.83.3.1 alias

**BString** Bds::Station::alias

Alias name to be returned to the user.

#### 7.83.3.2 channels

**BList**<[ArrayChannel](#) > Bds::Station::channels

List of channels if an Array.

#### 7.83.3.3 description

**BString** Bds::Station::description

Description.

#### 7.83.3.4 id

**BUInt32** Bds::Station::id

Unique ID when stored in a database or for other uses.

#### 7.83.3.5 name

**BString** Bds::Station::name

The name.

#### 7.83.3.6 type

**BString** Bds::Station::type

The [Station](#) type. Set to "array" or "station".

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

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

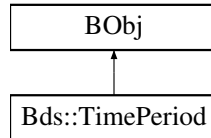


## 7.84 Bds::TimePeriod Class Reference

This class defines a [TimePeriod](#).

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

Inheritance diagram for Bds::TimePeriod:



### Public Member Functions

- [TimePeriod](#) ( **BTimeStamp** [startTime](#)= **BTimeStamp**(), **BTimeStamp** [endTime](#)= **BTimeStamp**())
- **BString** [getType](#) ()
- **BError** [setMembers](#) ( **BDictString** &members)
- **BError** [setMember](#) ( **BString** name, **BString** value)
- **BError** [getMembers](#) ( **BDictString** &members)
- **BError** [getMember](#) ( **BString** name, **BString** &value)

### Public Attributes

- **BTimeStamp** [startTime](#)  
*The Start time to the nearest us.*
- **BTimeStamp** [endTime](#)  
*The End time to the nearest us.*

#### 7.84.1 Detailed Description

This class defines a [TimePeriod](#).

It has [startTime](#) and [endTime](#) fields. [Note](#) the [endTime](#) is not included in the period.

#### 7.84.2 Constructor & Destructor Documentation

##### 7.84.2.1 TimePeriod()

```

Bds::TimePeriod::TimePeriod (
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp() )
  
```



## 7.84.3 Member Function Documentation

### 7.84.3.1 getMember()

```
BError Bds::TimePeriod::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

### 7.84.3.2 getMembers()

```
BError Bds::TimePeriod::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.84.3.3 getType()

```
BString Bds::TimePeriod::getType ( )
```

### 7.84.3.4 setMember()

```
BError Bds::TimePeriod::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

### 7.84.3.5 setMembers()

```
BError Bds::TimePeriod::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.



## 7.84.4 Member Data Documentation

### 7.84.4.1 endTime

**BTimeStamp** Bds::TimePeriod::endTime

The End time to the nearest us.

### 7.84.4.2 startTime

**BTimeStamp** Bds::TimePeriod::startTime

The Start time to the nearest us.

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

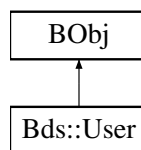
- [BdsD.h](#)
- [BdsD.cc](#)

## 7.85 Bds::User Class Reference

This holds information on a user.

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

Inheritance diagram for Bds::User:



### Public Member Functions

- **User** ( BUInt32 id=0, BString user= BString(), BString password= BString(), BString name= BString(), BString email= BString(), BString telephone= BString(), BString address= BString(), BInt32 enabled=0, BList< BString > groups= BList< BString >())
- **BString** getType ()
- **BError** setMembers ( BDictString &members)
- **BError** setMember ( BString name, BString value)
- **BError** getMembers ( BDictString &members)
- **BError** getMember ( BString name, BString &value)



## 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.*

### 7.85.1 Detailed Description

This holds information on a user.

All information on a BDS users is stored along with the security groups they belong to.

### 7.85.2 Constructor & Destructor Documentation

#### 7.85.2.1 User()

```
Bds::User::User (
    BUInt32 id = 0,
    BString user = BString(),
    BString password = BString(),
    BString name = BString(),
    BString email = BString(),
    BString telephone = BString(),
    BString address = BString(),
    BInt32 enabled = 0,
    BList< BString > groups = BList< BString >() )
```

### 7.85.3 Member Function Documentation



### 7.85.3.1 getMember()

```
BError Bds::User::getMember (
    BString name,
    BString & value ) [virtual]
```

Reimplemented from **BObj**.

### 7.85.3.2 getMembers()

```
BError Bds::User::getMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

### 7.85.3.3 getType()

```
BString Bds::User::getType ( )
```

### 7.85.3.4 setMember()

```
BError Bds::User::setMember (
    BString name,
    BString value ) [virtual]
```

Reimplemented from **BObj**.

### 7.85.3.5 setMembers()

```
BError Bds::User::setMembers (
    BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

## 7.85.4 Member Data Documentation



#### 7.85.4.1 address

**BString** Bds::User::address

The Users postal address.

#### 7.85.4.2 email

**BString** Bds::User::email

The users email Address.

#### 7.85.4.3 enabled

**BInt32** Bds::User::enabled

Whether the users account is enabled.

#### 7.85.4.4 groups

**BList**< **BString** > Bds::User::groups

The security groups the user belongs to.

#### 7.85.4.5 id

**BUInt32** Bds::User::id

The unique user ID.

#### 7.85.4.6 name

**BString** Bds::User::name

The Users full name.



#### 7.85.4.7 password

**BString** Bds::User::password

The Users password.

#### 7.85.4.8 telephone

**BString** Bds::User::telephone

The Users telephone number.

#### 7.85.4.9 user

**BString** Bds::User::user

The [User](#) ID.

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

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







## Chapter 8

# File Documentation

### 8.1 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp File Reference

```
#include <BdsCompress.h>
#include <BEndian.h>
```

#### Namespaces

- [Bds](#)

#### Functions

- **BError** [Bds::bdsUnCompressCm8](#) ( **BUInt8** \*buffer, **BUInt** n, **BArray**< **BInt32** > & data)  
*Uncompress CM8 formatted data.*

### 8.2 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.d File Reference

### 8.3 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.h File Reference

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

#### Classes

- class [Bds::CompressSteim1](#)  
*Steim1 un-compress class.*



## Namespaces

- [Bds](#)

## Functions

- **Error** [Bds::bdsUnCompressCm8](#) ( **BUInt8** \*buffer, **BUInt** n, **BArray**< **BInt32** > & data)  
*Uncompress CM8 formatted data.*
- **Error** [Bds::bdsUnCompressSteim1](#) ( **BUInt8** \*buffer, **BUInt** n, **BArray**< **BInt32** > & data)  
*Uncompress STEIM1 formatted data.*

## 8.4 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp File Reference

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

## Namespaces

- [Bds](#)

## 8.5 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.d File Reference

## 8.6 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h File Reference

```
#include <BString.h>
#include <BFile.h>
#include <BTimeStamp.h>
#include <BdsD.h>
#include <BdsDataFile.h>
```

## Classes

- class [Bds::DataCollate](#)  
*Not sure if this is used or what it does.*

## Namespaces

- [Bds](#)

## 8.7 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp File Reference

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



## Namespaces

- [Bds](#)

## 8.8 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.d File Reference

## 8.9 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h File Reference

```
#include <BString.h>
#include <BFile.h>
#include <BdsLib.h>
#include <BdsD.h>
```

## Classes

- class [Bds::DataBlockPos](#)  
*This defines the position of a data block in a file. It is used by the BDS data convertors to order blocks by time.*
- class [Bds::DataFileOptions](#)  
*This defines a list of BDS data convtor options.*
- class [Bds::DataFile](#)  
*This class defines the interface for generic data file access that all of the BDS data conterors share.*

## Namespaces

- [Bds](#)

## 8.10 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp File Reference

```
#include <BdsDataFileAd22.h>
#include <TimeControlPll.h>
#include <BDebug.h>
#include <errno.h>
```

## Namespaces

- [Bds](#)

## Macros

- `#define` [DEBUG\\_VELATRACK](#) 1



## 8.10.1 Macro Definition Documentation

### 8.10.1.1 DEBUG\_VELATRACK

```
#define DEBUG_VELATRACK 1
```

## 8.11 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.d File Reference

## 8.12 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h File Reference

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

### Classes

- class [Bds::DataFileAd22](#)  
*Data file convertor for AD22 format files.*

### Namespaces

- [Bds](#)

## 8.13 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp File Reference

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

### Namespaces

- [Bds](#)

### Functions

- static **BString** [Bds::nullString](#) ( **BString** s)



## 8.14 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.d File Reference

## 8.15 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h File Reference

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

### Classes

- class [Bds::DataFileAscii](#)  
*Data file convertor for ASCII format files.*

### Namespaces

- [Bds](#)

## 8.16 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp File Reference

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

### Namespaces

- [Bds](#)

## 8.17 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.d File Reference

## 8.18 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h File Reference

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

### Classes

- class [Bds::DataFileBdrs](#)  
*Data file convertor for BDRS format files.*

### Namespaces

- [Bds](#)



## 8.19 /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 <BDebug.h>
#include <zlib.h>
#include <canada_compress.h>
```

### Namespaces

- [Bds](#)

### Macros

- #define [LDEBUG](#) 0
- #define [LDEBUG2](#) 0
- #define [LDEBUG3](#) 0
- #define [dlprintf](#)(fmt, a...)
- #define [dl2printf](#)(fmt, a...)
- #define [dl3printf](#)(fmt, a...)
- #define [ALLOW\\_TIMESTAMP\\_JITTER](#) 1
- #define [TIMESTAMP\\_JITTER](#) 100

### Functions

- **BUInt32** [Bds::crc](#) ( **BUInt32** crc, void \* **data**, int numBytes)

### Variables

- const **BString** [Bds::BdsDataFileVersion](#) = "1.2.0"

### 8.19.1 Macro Definition Documentation

#### 8.19.1.1 [ALLOW\\_TIMESTAMP\\_JITTER](#)

```
#define ALLOW_TIMESTAMP_JITTER 1
```



#### 8.19.1.2 dl2printf

```
#define dl2printf(  
    fmt,  
    a... )
```

#### 8.19.1.3 dl3printf

```
#define dl3printf(  
    fmt,  
    a... )
```

#### 8.19.1.4 dlprintf

```
#define dlprintf(  
    fmt,  
    a... )
```

#### 8.19.1.5 LDEBUG

```
#define LDEBUG 0
```

#### 8.19.1.6 LDEBUG2

```
#define LDEBUG2 0
```

#### 8.19.1.7 LDEBUG3

```
#define LDEBUG3 0
```

#### 8.19.1.8 TIMESTAMP\_JITTER

```
#define TIMESTAMP_JITTER 100
```



## 8.20 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.d File Reference

## 8.21 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h File Reference

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

### Classes

- struct [Bds::BdsDataBlockHeader](#)  
*BdsDataFileBds: internal fixed size BDS Data Block header.*
- struct [Bds::BdsDataBlock](#)  
*BdsDataFileBds: internal fixed size BDS Data Block.*
- struct [Bds::BdsDataPacketHeader](#)  
*BdsDataFileBds internal file storage packet header.*
- class [Bds::BdsDataPacket](#)  
*BdsDataFileBds: internal file storage packet.*
- class [Bds::BdsDataBlockPos](#)  
*BdsDataFileBds: internal file storage data block position.*
- class [Bds::BdsDataSegment](#)  
*BdsDataFileBds: internal file storage data segment.*
- class [Bds::BdsDataStreamlet](#)  
*BdsDataFileBds: internal file storage data streamlet.*
- class [Bds::DataFileBds](#)  
*This class implements the BDS Data File/Stream access system.*

### Namespaces

- [Bds](#)

### Enumerations

- enum [Bds::BdsDataType](#) { [Bds::BdsDataTypeBlock](#) = 0x42534442, [Bds::BdsDataTypeInfo](#) = 0x30534442, [Bds::BdsDataTypeData](#) = 0x31534442, [Bds::BdsDataTypeInfoExtra](#) = 0x32534442 }
- BdsDataFileBds: internal file block type field.*

## 8.22 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp File Reference

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



## Namespaces

- [Bds](#)

## Functions

- `template<typename T >`  
`T clip (T in, T low, T high)`

### 8.22.1 Function Documentation

#### 8.22.1.1 clip()

```
template<typename T >
T clip (
    T in,
    T low,
    T high ) [inline]
```

## 8.23 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.d File Reference

## 8.24 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h File Reference

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

## Classes

- class [Bds::DataFileBknas](#)  
*Data file convertor for BKNAS format files.*

## Namespaces

- [Bds](#)

## 8.25 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.cpp File Reference

```
#include <BdsDataFileCd.h>
#include <arpa/inet.h>
#include <errno.h>
#include <canada_compress.h>
#include <BEndian.h>
#include <BDebug.h>
```



## Namespaces

- [Bds](#)

## Macros

- `#define LDEBUG 0`
- `#define dprintf(fmt, a...)`
- `#define INCLUDE_CHANNEL_AUTH 1`
- `#define ALLOW_TIMESTAMP_JITTER 1`
- `#define TIMESTAMP_JITTER 100`
- `#define MULTIPLE_SEGMENT 0`
- `#define SEGMENT_GAP 3600000`
- `#define ntohl(x) __bswap_64(x)`
- `#define htonl(x) ntohl(x)`

## Functions

- `static void Bds::crclnit ()`
- `static uint64_t Bds::crc64 (const void *buffer, const uint32_t len)`
- `BString Bds::getHexString (char * data, int len)`
- `int Bds::duplicateDump (DataBlock &data1, DataBlock &data2, int channel)`

## Variables

- `const int ErrorFormatNoDataFormat = 100`
- `static uint64_t Bds::crcVec [256]`
- `static int Bds::crclnitDone`

## 8.25.1 Macro Definition Documentation

### 8.25.1.1 ALLOW\_TIMESTAMP\_JITTER

```
#define ALLOW_TIMESTAMP_JITTER 1
```

### 8.25.1.2 dprintf

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



### 8.25.1.3 htonl

```
#define htonl(  
    x )  ntohl(x)
```

### 8.25.1.4 INCLUDE\_CHANNEL\_AUTH

```
#define INCLUDE_CHANNEL_AUTH 1
```

### 8.25.1.5 LDEBUG

```
#define LDEBUG 0
```

### 8.25.1.6 MULTIPLE\_SEGMENT

```
#define MULTIPLE_SEGMENT 0
```

### 8.25.1.7 ntohl

```
#define ntohl(  
    x )  __bswap_64(x)
```

### 8.25.1.8 SEGMENT\_GAP

```
#define SEGMENT_GAP 3600000
```

### 8.25.1.9 TIMESTAMP\_JITTER

```
#define TIMESTAMP_JITTER 100
```

## 8.25.2 Variable Documentation



### 8.25.2.1 ErrorFormatNoDataFormat

```
const int ErrorFormatNoDataFormat = 100
```

## 8.26 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.d File Reference

## 8.27 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h File Reference

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

### Classes

- struct [Bds::CdChannel\\_1v0](#)  
*BdsDataFile: Internal CD1.0 channel information.*
- struct [Bds::CdDataFormatFrame\\_1v0](#)  
*BdsDataFile: Internal CD1.0 frame information.*
- class [Bds::CdDataChannel](#)  
*BdsDataFile: Internal CD channel information.*
- class [Bds::CdPacketData](#)  
*BdsDataFile: Internal CD data packet.*
- class [Bds::CdFlag](#)  
*BdsDataFile: Internal CD flag.*
- class [Bds::DataFileCd](#)  
*Data file convertor for CD1.0 and CD1.1 file formats.*

### Namespaces

- [Bds](#)

## 8.28 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp File Reference

```
#include <BdsDataFileCss.h>  
#include <BEndian.h>  
#include <BDebug.h>  
#include <errno.h>
```

### Namespaces

- [Bds](#)



## 8.29 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.d File Reference

## 8.30 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h File Reference

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

### Classes

- class [Bds::DataFileCssData](#)  
*DataFileCss internal CSS data type.*
- class [Bds::DataFileCss](#)  
*Data file convertor for CSS format files.*

### Namespaces

- [Bds](#)

## 8.31 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp File Reference

```
#include <BdsDataFileGcf.h>
#include <arpa/inet.h>
#include <errno.h>
#include <gcf2.h>
#include <BDebug.h>
```

### Namespaces

- [Bds](#)

### Macros

- `#define DEBUG 0`
- `#define TEST\_REORDER 0`

### 8.31.1 Macro Definition Documentation

#### 8.31.1.1 [DEBUG](#)

```
#define DEBUG 0
```



### 8.31.1.2 TEST\_REORDER

```
#define TEST_REORDER 0
```

## 8.32 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.d File Reference

## 8.33 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h File Reference

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

### Classes

- struct [Bds::GcfChannel](#)  
*DataFileGcf internal GCF channel information.*
- class [Bds::DataFileGcf](#)  
*Data file convertor for GCF format files.*

### Namespaces

- [Bds](#)

## 8.34 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp File Reference

```
#include <BdsDataFileIms.h>
#include <BTimeStamp.h>
#include <errno.h>
```

### Namespaces

- [Bds](#)

### Functions

- static **BError** [Bds::fixedString](#) (double v, int fieldWidth, int numDecimal, **BString** &str)
- void [Bds::dataCalculateDifference](#) ( **BInt32** &prevValue, **BArray**< **BInt32** > & data)
- void [Bds::dataCalculateUnDifference](#) ( **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)
- **BError** [Bds::dataDeCompressCm6](#) (int &prevValue1, int &prevValue2, **BString** &d, **BArray**< **BInt32** > & data)
- static void [Bds::dataConvert](#) (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BInt32** > &dataOut)



## Variables

- static char [Bds::cm6Table](#) [64]
- static **BUInt8** [Bds::cm6TableRev](#) [128]

## 8.35 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.d File Reference

## 8.36 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h File Reference

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

## Classes

- class [Bds::DataFileIms](#)  
*Data file convertor for IMS format files.*

## Namespaces

- [Bds](#)

## 8.37 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp File Reference

```
#include <BdsDataFileLac.h>
#include <BEndian.h>
#include <BDebug.h>
#include <errno.h>
```

## Namespaces

- [Bds](#)

## 8.38 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.d File Reference

## 8.39 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h File Reference

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



## Classes

- class [Bds::DataFileLac](#)  
*Data file convertor for LAC format files.*

## Namespaces

- [Bds](#)

## 8.40 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.cpp File Reference

```
#include <BdsDataFileLog.h>
#include <BTimeStamp.h>
#include <errno.h>
```

## Namespaces

- [Bds](#)

## Functions

- static **BString** [Bds::stringFormat](#) ( **BTimeStamp** t)
- static **BString** [Bds::removeCR](#) ( **BString** str)

## 8.41 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.d File Reference

## 8.42 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.h File Reference

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

## Classes

- class [Bds::DataFileLog](#)  
*Data file convertor for LOG format files.*

## Namespaces

- [Bds](#)



## 8.43 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp File Reference

```
#include <BdsDataFileResponse.h>
#include <errno.h>
```

### Namespaces

- [Bds](#)

### Macros

- #define [LDEBUG](#) 0
- #define [dprintf](#)(fmt, a...)

#### 8.43.1 Macro Definition Documentation

##### 8.43.1.1 dprintf

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

##### 8.43.1.2 LDEBUG

```
#define LDEBUG 0
```

## 8.44 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.d File Reference

## 8.45 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h File Reference

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

### Classes

- class [Bds::DataFileResponse](#)

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



## Namespaces

- [Bds](#)

## 8.46 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp File Reference

```
#include <BdsDataFileSac.h>
#include <errno.h>
```

## Namespaces

- [Bds](#)

## 8.47 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.d File Reference

## 8.48 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.h File Reference

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

## Classes

- class [Bds::DataFileSac](#)  
*Data file convertor for SAC format files.*

## Namespaces

- [Bds](#)

## 8.49 /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

- [Bds](#)



## 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

## 8.50 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.d File Reference

## 8.51 /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>
```

## Classes

- class [Bds::DataFileTapeDigitiser](#)  
*This class implements the TapeDigitiser's file output conversion and storing system.*

## Namespaces

- [Bds](#)

## 8.52 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp File Reference

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

## Namespaces

- [Bds](#)



## 8.53 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.d File Reference

## 8.54 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h File Reference

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

### Classes

- class [Bds::DataFileWra](#)  
*Data file convertor for WRA format files.*

### Namespaces

- [Bds](#)

## 8.55 /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

- [Bds](#)

### Functions

- static **BList**< **BString** > [parseStringFixedFields](#) ( **BString** s, int \*fieldWidths)

### 8.55.1 Function Documentation

#### 8.55.1.1 parseStringFixedFields()

```
static BList< BString> parseStringFixedFields (
    BString s,
    int * fieldWidths ) [static]
```



## 8.56 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.d File Reference

## 8.57 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h File Reference

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

### Classes

- class [Bds::DataFileWraAgso](#)  
*Data file convertor for WRA AGSO format files.*

### Namespaces

- [Bds](#)

## 8.58 /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>
#include <BdsDataFileSeed.h>
#include <BdsDataFileSac.h>
#include <BdsDataFileCd.h>
#include <BdsDataFileResponse.h>
#include <BdsDataFileLog.h>
#include <BdsDataFileAd22.h>
#include <BdsDataFileLac.h>
#include <BdsDataFileCss.h>
```

### Namespaces

- [Bds](#)

### Functions

- **BString** [Bds::fixedWidthValue](#) (double v, int width)  
*This returns a double as a fixed width string truncating the data.*



## Variables

- DataFormats [Bds::dataFormats](#)

## 8.59 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.d File Reference

## 8.60 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h File Reference

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

## Classes

- class [Bds::DataFormats](#)  
*This class defines the interface for generic data file access.*

## Namespaces

- [Bds](#)

## Functions

- **BString** [Bds::fixedWidthValue](#) (double v, int width)  
*This returns a double as a fixed width string truncating the data.*

## 8.61 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp File Reference

```
#include <BdsDataFileSeed.h>  
#include <BEndian.h>  
#include <errno.h>  
#include <BDebug.h>
```

## Namespaces

- [Bds](#)

## Macros

- #define [DEBUG](#) 0
- #define [DEBUG\\_BLOCKETTE](#) 0
- #define [DEBUG\\_BLOCKS](#) 0
- #define [FILL\\_BLOCKS](#) 1
- #define [ROUND\\_TIMESTAMPS\\_TO\\_10US](#) 1
- #define [dlprintf](#)(fmt, a...)



## Functions

- static void [Bds::bdsDataFileSeedLogWarning](#) (char \*str)
- static void [Bds::bdsDataFileSeedLogError](#) (char \*str)
- static hptime\_t [Bds::seedTime](#) ( **BTimeStamp** t)
- static **BString** [Bds::seedTimeString](#) ( **BTimeStamp** t)
- static void [Bds::dataConvert](#) (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BInt32** > &dataOut)
- static void [Bds::dataConvert](#) (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BFloat32** > &dataOut)
- static void [Bds::record\\_handler](#) (char \*record, int reclen, void \*info)

## 8.61.1 Macro Definition Documentation

### 8.61.1.1 DEBUG

```
#define DEBUG 0
```

### 8.61.1.2 DEBUG\_BLOCKETTE

```
#define DEBUG_BLOCKETTE 0
```

### 8.61.1.3 DEBUG\_BLOCKS

```
#define DEBUG_BLOCKS 0
```

### 8.61.1.4 dlprintf

```
#define dlprintf(  
    fmt,  
    a... )
```

### 8.61.1.5 FILL\_BLOCKS

```
#define FILL_BLOCKS 1
```



#### 8.61.1.6 ROUND\_TIMESTAMPS\_TO\_10US

```
#define ROUND_TIMESTAMPS_TO_10US 1
```

### 8.62 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.d File Reference

### 8.63 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h File Reference

```
#include <BdsDataFile.h>
#include <BdsSeedTypes.h>
#include <BMutex.h>
#include <libmseed-beam/libmseed.h>
#include <libmseed-beam/unpackdata.h>
```

#### Classes

- class [Bds::DataFileSeed](#)  
*Data file convertor for SEED file formats.*

#### Namespaces

- [Bds](#)

### 8.64 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp File Reference

```
#include <BdsSeedType.h>
```

#### Namespaces

- [Bds](#)

### 8.65 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.d File Reference

### 8.66 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.h File Reference

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



## Classes

- class [Bds::BdsSeedType](#)  
*BdsDataFileSeed internal parent for all SEED types.*

## Namespaces

- [Bds](#)

## 8.67 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.cpp File Reference

```
#include <BdsSeedTypes.h>
```

## Namespaces

- [Bds](#)

## 8.68 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.d File Reference

## 8.69 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.idl File Reference

## 8.70 /src/blacknest/bds/bds/bdsDataLib/canada\_compress.d File Reference

## 8.71 /src/blacknest/bds/bds/bdsDataLib/canada\_compress.h File Reference

```
#include <arpa/inet.h>
```

## Macros

- #define [CANCOMP\\_ERR](#) -1 /\* unrecoverable **error** (malloc fails) \*/
- #define [CANCOMP\\_SUCCESS](#) 0 /\* success \*/
- #define [CANCOMP\\_NOT\\_20](#) 1 /\* **number** of samples not divisible by 20 \*/
- #define [CANCOMP\\_CORRUPT](#) 2 /\* corrupted call \*/
- #define [CANCOMP\\_EXCEED](#)

## Functions

- int [canada\\_uncompress](#) (unsigned char \*b, uint32\_t \*y, int \*n, int m, uint32\_t \*v0)  
*De-compressses Canada format seismic data.*
- int [canada\\_compress](#) (unsigned char \*b, unsigned long \*y, int \*n, int m, unsigned long \*v0)  
*Compressses Canada format seismic data.*



## 8.71.1 Macro Definition Documentation

### 8.71.1.1 CANCOMP\_CORRUPT

```
#define CANCOMP_CORRUPT 2 /* corrupted call */
```

### 8.71.1.2 CANCOMP\_ERR

```
#define CANCOMP_ERR -1 /* unrecoverable error (malloc fails) */
```

### 8.71.1.3 CANCOMP\_EXCEED

```
#define CANCOMP_EXCEED
```

**Value:**

```
3 /* number of bytes available in compressed  
   data exceeded during decompression */
```

### 8.71.1.4 CANCOMP\_NOT\_20

```
#define CANCOMP_NOT_20 1 /* number of samples not divisible by 20 */
```

### 8.71.1.5 CANCOMP\_SUCCESS

```
#define CANCOMP_SUCCESS 0 /* success */
```

## 8.71.2 Function Documentation



#### 8.71.2.1 canada\_compress()

```
int canada_compress (
    unsigned char * b,
    unsigned long * y,
    int * n,
    int m,
    unsigned long * v0 )
```

Compresses Canada format seismic data.

#### 8.71.2.2 canada\_uncompress()

```
int canada_uncompress (
    unsigned char * b,
    uint32_t * y,
    int * n,
    int m,
    uint32_t * v0 )
```

De-compresses Canada format seismic data.

## 8.72 BdsC.cc File Reference

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

### Namespaces

- [Bds](#)

## 8.73 BdsC.d File Reference

## 8.74 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>
```



## Classes

- class [Bds::DataAccess](#)  
*This is the Data Access API interface.*
- class [Bds::DataAddAccess](#)  
*This is the DataAdd Access API interface.*
- class [Bds::AdminAccess](#)  
*This is the [AdminAccess](#) Access API interface.*

## Namespaces

- [Bds](#)

## Variables

- const **BUInt32** [Bds::apiVersion](#) = 0

## 8.75 BdsD.cc File Reference

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

## Namespaces

- [Bds](#)

## 8.76 BdsD.d File Reference

## 8.77 BdsD.h File Reference

BOAP data class definitions for: [Bds](#).

```
#include <Boap.h>
#include <BObj.h>
#include <BDate.h>
#include <BTimeStamp.h>
#include <BComplex.h>
#include <BList.h>
#include <BArray.h>
```



## Classes

- class [Bds::Point](#)  
*This class defines an X,Y location.*
- class [Bds::TimePeriod](#)  
*This class defines a [TimePeriod](#).*
- class [Bds::ListRange](#)  
*This class defines an integer based range.*
- class [Bds::Network](#)  
*This class defines a seismic [Network](#) organisation.*
- class [Bds::Source](#)  
*This class defines a seismic data [Source](#).*
- class [Bds::SourcePriority](#)  
*This class defines a [Source](#) Priority entry.*
- class [Bds::ChannelName](#)  
*This class defines a full channel name.*
- class [Bds::ArrayChannel](#)  
*This class defines an arrays channel.*
- class [Bds::Station](#)  
*This class defines a seismic station.*
- class [Bds::Location](#)  
*This class defines the physical location of a [Station](#).*
- class [Bds::PoleZero](#)  
*This class defines a Pole/Zero [Response](#).*
- class [Bds::Fap](#)  
*This class defines an entry in an Amplitude/Phase [Response](#) table.*
- class [Bds::FirEntry](#)  
*This class defines an entry in a FIR coefficient table.*
- class [Bds::Fir](#)  
*This class defines an FIR response table.*
- class [Bds::Response](#)  
*This class defines a seismic [Response](#) characteristic.*
- class [Bds::Calibration](#)  
*This class defines a calibration setting.*
- class [Bds::Digitiser](#)  
*This class defines a seismic [Digitiser](#).*
- class [Bds::Sensor](#)  
*This class defines a seismic [Sensor](#).*
- class [Bds::ChannellInstrument](#)  
*This class defines a [Channel](#)'s instrument.*
- class [Bds::Channel](#)  
*This class defines a seismic data [Channel](#).*
- class [Bds::SelectionInfo](#)  
*This class defines the set of metadata or seismic data selected when `getSelectionInfo()` is use.*
- class [Bds::SelectionChannel](#)  
*This class defines a channel for selection.*
- class [Bds::Selection](#)  
*This class defines a generic metadata or seismic data selection.*
- class [Bds::ChannellInfo](#)  
*This class provides information on a channel.*
- class [Bds::ChannellInfos](#)



- This class provides metadata information on a set of channels.*

  - class [Bds::DataFileInfo](#)

*This class defines information on a sensor data file.*
- class [Bds::DataChannel](#)

*This class defines information on a single channels set of data stored in a file.*
- class [Bds::DataInfo](#)

*This class defines information on a set of data.*
- class [Bds::DataAvail](#)

*This class provides availability information on a particular period of data.*
- class [Bds::DataAvailChan](#)

*This class defines availability information on a set of data.*
- class [Bds::DataHandle](#)

*This defines a handle to a sensor data stream/file when opened for read or write.*
- class [Bds::DataBlock](#)

*This class provides the actual seismic data values contained within a single data block.*
- class [Bds::User](#)

*This holds information on a user.*
- class [Bds::Group](#)

*This holds information on a user security group.*
- class [Bds::AccessGroup](#)

*This holds information on data access groups.*
- class [Bds::Change](#)

*This holds information on a metadata or sensor data change.*
- class [Bds::ChangeGroup](#)

*This holds information on a set of Changes.*
- class [Bds::Note](#)

*This holds information on a [Note](#) for general information.*
- class [Bds::Log](#)

*This holds information on a [Log](#) entry.*
- class [Bds::LogSelect](#)

*This defines the selection criteria when requesting a set of log entries.*
- class [Bds::CleanOptions](#)

*This defines the set of clean options used in the `clean()` function.*
- class [Bds::DataFormat](#)

*This holds information on a seismic data format.*

## Namespaces

- [Bds](#)

## Enumerations

- enum [Bds::Errors](#) {  
[Bds::ErrorNoMetaData](#) = 64, [Bds::ErrorDataQuality](#) = 65, [Bds::ErrorSlaveMode](#) = 66, [Bds::ErrorTimeStamp](#) = 67,  
[Bds::ErrorValidate](#) = 80, [Bds::ErrorValidateMissingBlocks](#) = 81, [Bds::ErrorValidateTimeBackwards](#) = 82,  
[Bds::ErrorValidateFilenameTime](#) = 83,  
[Bds::ErrorValidateMetaData](#) = 84, [Bds::ErrorValidateFix](#) = 85, [Bds::ErrorValidateDuplicate](#) = 86,  
[Bds::ErrorValidateReorder](#) = 87,  
[Bds::ErrorValidateBdsFudge](#) = 88 }



*The System Error number list in addition to standard system error numbers.*

- enum [Bds::Priority](#) { [Bds::PriorityLow](#), [Bds::PriorityNormal](#), [Bds::PriorityHigh](#) }

*Priority levels.*

- enum [Bds::Mode](#) { [Bds::ModeMaster](#), [Bds::ModeSlave](#) }

*BdsServer mode.*

- enum [Bds::DataFlags](#) {  
[Bds::DataFlagNone](#) = 0x00, [Bds::DataFlagClipDataToTime](#) = 0x01, [Bds::DataFlagClipDataToChannels](#) =  
0x02, [Bds::DataFlagMergeSegments](#) = 0x04,  
[Bds::DataFlagNoMetadata](#) = 0x08 }

*Flags when opening data files.*

- enum [Bds::SelectionGroup](#) { [Bds::SelectionGroupData](#), [Bds::SelectionGroupMetaData](#), [Bds::SelectionGroupDataWithCount](#) }

*The Selection group when making selections.*

- enum [Bds::SampleFormat](#) {  
[Bds::SampleFormatUnknown](#), [Bds::SampleFormatInt16](#), [Bds::SampleFormatInt32](#), [Bds::SampleFormatFloat32](#),  
[Bds::SampleFormatFloat64](#), [Bds::SampleFormatInt24](#) }

*The actual format of a data sample.*

- enum [Bds::AvailType](#) { [Bds::AvailNone](#), [Bds::AvailPartial](#), [Bds::AvailFull](#) }

*A flag defining the data availability state.*

### 8.77.1 Detailed Description

BOAP data class definitions for: [Bds](#).

Date

2020-06-23T15:00:48

The classes in here have been defined by a BOAP \*.bidl file and define classes able to be communicated across a BOAP link

## 8.78 BdsLib.cpp File Reference

```
#include <BdsLib.h>
#include <math.h>
#include <complex>
```

### Namespaces

- [Bds](#)



## Functions

- void [Bds::bdsDumpPoleZeros](#) (PoleZero poleZeros)  
*Debug print out a PoleZeros object.*
- void [Bds::bdsChannelGetTypeAux](#) ( BString name, BString &type, BString &aux)  
*Get the channel type and aux fields from a generic channel name.*
- BString [Bds::bdsChannelGetName](#) ( BString type, BString aux)  
*Create a full channel name from a channels type and aux fields.*
- BError [Bds::bdsDataInfoSetTimeRange](#) (DataInfo &dataInfo)  
*Restricts the time tange of all of the DataInfo's channels to match the DataInfo's startTime/endTime fields.*
- BError [Bds::bdsDataInfoFromInfo](#) ( BDictString info, DataInfo &dataInfo, Bool append)  
*Convert info to DataInfo.*
- BError [Bds::bdsInfoFromDataInfo](#) (const DataInfo &dataInfo, BDictString &info)  
*Converts a DataInfo object into a BDictString list of named strings.*
- BError [Bds::bdsDataInfoMergeFlatten](#) (DataInfo &dataInfo, const DataInfo &dataInfoAdd)  
*Merges a DataInfo into another flattening the segments to 1 for use in dataOpen() calls.*
- BString [Bds::bdsStationAlias](#) (Station station)  
*Returns the station alias if set else its name.*
- void [Bds::bdsDumpSelection](#) (Selection sel)  
*Debug print out a Selection object.*
- void [Bds::bdsDumpDataInfo](#) (DataInfo dataInfo, int includeInfo=0)  
*Debug print out a DataInfo object.*
- void [Bds::bdsDumpChannelInfos](#) (const ChannelInfos &channelInfos)  
*Debug print out a ChannelInfos object.*
- void [Bds::bdsDumpData](#) (const DataBlock &dataBlock, int nSamples=0)  
*Debug print out a DataBlock object.*
- BString [Bds::bdsDataChannelInfo](#) (const DataChannel &dataChannel)  
*Returns a string representation of a DataChannel object.*
- BString [Bds::bdsDataChannelRef](#) (const DataChannel &dataChannel)  
*Returns the string reference name of a DataChannel object.*
- BString [Bds::bdsDataChannelRef](#) (const ChannelInfo &channelInfo)  
*Returns the string reference name of a ChannelInfo object.*
- BError [Bds::bdsDataChannelOverallResponse](#) (const ChannelInfo &channelInfo, Response &response)  
*Returns the overal response from the list of responses in a ChannelInfo.*
- BString [Bds::bdsSelectionChannelInfo](#) (const Selection &selection, BUInt channel)  
*Returns a string describing the name and time period of a selection channel.*
- double [Bds::bdsPoleZeroGain](#) (const PoleZero &poleZero, double frequency)  
*Calculates the overal gain of the given PoleZero transfer function.*
- void [Bds::bdsPoleZeroGainPhase](#) (const PoleZero &poleZero, double frequency, double &gain, double &phase)  
*Calculates the overal gain and phase of the given PoleZero transfer function.*
- void [Bds::bdsPoleZeroToFap](#) (const PoleZero &poleZero, BUInt nPoints, double calibrationFrequency, double sampleFrequency, BArray< Fap > &fap)  
*Convert PoleZero to FAP.*
- static BString [Bds::fileNameTime](#) ( BTimeStamp t)
- BString [Bds::bdsFileNameExpand](#) ( BString fileName, ChannelInfo &channelInfo)  
*Default filename from a ChannelInfo.*
- BString [Bds::bdsFileNameExpand](#) ( BString fileName, ChannelInfos &channelInfos)  
*Default filename from a list of ChannelInfo's.*



## 8.79 BdsLib.d File Reference

## 8.80 BdsLib.dox File Reference

### Namespaces

- [Bds](#)

### Functions

- void [Bds::bdsChannelGetTypeAux](#) ( **BString** name, **BString** &type, **BString** &aux)

*Get the channel type and aux fields from a generic channel name.*

## 8.81 BdsLib.h File Reference

General BdsLib API functions.

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

### Classes

- class [Bds::ResponseObj](#)  
*Response object adding string conversion.*

- class [Bds::DataError](#)  
*This stores a data error. It includes an error number and a string as well as information on what seismic channel it is for.*

### Namespaces

- [Bds](#)



## Functions

- void [Bds::bdsChannelGetTypeAux](#) ( **BString** name, **BString** &type, **BString** &aux)  
*Get the channel type and aux fields from a generic channel name.*
- **BString** [Bds::bdsChannelGetName](#) ( **BString** type, **BString** aux)  
*Create a full channel name from a channels type and aux fields.*
- **BError** [Bds::bdsDataInfoSetTimeRange](#) (DataInfo &dataInfo)  
*Restricts the time tange of all of the [DataInfo](#)'s channels to match the [DataInfo](#)'s startTime/endTime fields.*
- **BError** [Bds::bdsDataInfoFromInfo](#) ( **BDictString** info, DataInfo &dataInfo, **Bool** append)  
*Convert info to [DataInfo](#).*
- **BError** [Bds::bdsInfoFromDataInfo](#) (const DataInfo &dataInfo, **BDictString** &info)  
*Converts a [DataInfo](#) object into a [BDictString](#) list of named strings.*
- **BError** [Bds::bdsDataInfoMergeFlatten](#) (DataInfo &dataInfo, const DataInfo &dataInfoAdd)  
*Merges a [DataInfo](#) into another flattening the segments to 1 for use in [dataOpen\(\)](#) calls.*
- **BString** [Bds::bdsStationAlias](#) (Station station)  
*Returns the station alias if set else its name.*
- **BString** [Bds::bdsDataChannelInfo](#) (const DataChannel &dataChannel)  
*Returns a string representation of a [DataChannel](#) object.*
- **BString** [Bds::bdsDataChannelRef](#) (const DataChannel &dataChannel)  
*Returns the string reference name of a [DataChannel](#) object.*
- **BString** [Bds::bdsDataChannelRef](#) (const ChannelInfo &channelInfo)  
*Returns the string reference name of a [ChannelInfo](#) object.*
- **BError** [Bds::bdsDataChannelOverallResponse](#) (const ChannelInfo &channelInfo, Response &response)  
*Returns the overal response from the list of responses in a [ChannelInfo](#).*
- **BString** [Bds::bdsSelectionChannelInfo](#) (const Selection &selection, **BUInt** channel)  
*Returns a string describing the name and time period of a selection channel.*
- **BString** [Bds::bdsFileNameExpand](#) ( **BString** fileName, ChannelInfo &channelInfo)  
*Default filename from a [ChannelInfo](#).*
- **BString** [Bds::bdsFileNameExpand](#) ( **BString** fileName, ChannelInfos &channelInfos)  
*Default filename from a list of [ChannelInfo](#)'s.*
- double [Bds::bdsPoleZeroGain](#) (const PoleZero &poleZero, double frequency)  
*Calculates the overal gain of the given [PoleZero](#) transfer function.*
- void [Bds::bdsPoleZeroGainPhase](#) (const PoleZero &poleZero, double frequency, double &gain, double &phase)  
*Calculates the overal gain and phase of the given [PoleZero](#) transfer function.*
- void [Bds::bdsPoleZeroToFap](#) (const PoleZero &poleZero, **BUInt** nPoints, double calibrationFrequency, double sampleFrequency, **BArray**< Fap > &fap)  
*Convert [PoleZero](#) to FAP.*
- void [Bds::bdsDumpSelection](#) (Selection sel)  
*Debug print out a [Selection](#) object.*
- void [Bds::bdsDumpDataInfo](#) (DataInfo dataInfo, int includeInfo=0)  
*Debug print out a [DataInfo](#) object.*
- void [Bds::bdsDumpChannelInfos](#) (const ChannelInfos &channelInfos)  
*Debug print out a [ChannelInfos](#) object.*
- void [Bds::bdsDumpData](#) (const DataBlock &dataBlock, int nSamples=0)  
*Debug print out a [DataBlock](#) object.*
- void [Bds::bdsDumpPoleZeros](#) (PoleZero poleZeros)  
*Debug print out a [PoleZeros](#) object.*



## Variables

- const int [Bds::NetworkNameLen](#) = 3  
*Maximum [Network](#) name length.*
- const int [Bds::StationNameLen](#) = 5  
*Maximum [Station](#) name length.*
- const int [Bds::ChannelTypeLen](#) = 3  
*Maximum [Channel](#) type name length.*
- const int [Bds::ChannelAuxLen](#) = 2  
*Maximum [Channel](#) Aux length.*
- const int [Bds::SourceLen](#) = 16  
*Maximum [Source](#) length.*

### 8.81.1 Detailed Description

General BdsLib API functions.

## 8.82 BdsS.cc File Reference

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

## Namespaces

- [Bds](#)

### 8.83 BdsS.d File Reference

### 8.84 BdsT.cc File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <BdsT.h>
#include <Control.h>
```

### 8.85 /src/blacknest/bds/bds/doc/bdsApiOverview.dox File Reference







# Index

/src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp, 317  
/src/blacknest/bds/bds/bdsDataLib/BdsCompress.d, 317  
/src/blacknest/bds/bds/bdsDataLib/BdsCompress.h, 317  
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp, 318  
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.d, 318  
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h, 318  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp, 318  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.d, 319  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h, 319  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp, 319  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.d, 320  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h, 320  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp, 320  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.d, 321  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h, 321  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp, 321  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.d, 321  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h, 321  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp, 322  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.d, 324  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h, 324  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp, 324  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.d, 325  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h, 325  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.cpp, 325  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.d, 328  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h, 328  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp, 328  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.d, 329  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h, 329  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp, 329  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.d, 330  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h, 330  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp, 330  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.d, 331  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h, 331  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp, 331  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.d, 331  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h, 331  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.cpp, 332  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.d, 332  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.h, 332  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp, 333  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.d, 333  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h, 333  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp, 334  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.d, 334  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.h, 334  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp, 334  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.d, 335  
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h, 335



- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp, 335
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.d, 336
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h, 336
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp, 336
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.d, 337
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h, 337
- /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp, 337
- /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.d, 338
- /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h, 338
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp, 338
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.d, 340
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h, 340
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp, 340
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.d, 340
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.h, 340
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.cpp, 341
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.d, 341
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.idl, 341
- /src/blacknest/bds/bds/bdsDataLib/canada\_compress.d, 341
- /src/blacknest/bds/bds/bdsDataLib/canada\_compress.h, 341
- /src/blacknest/bds/bds/doc/bdsApiOverview.dox, 351
- ~BdsDataPacket
  - Bds::BdsDataPacket, 76
- ~DataCollate
  - Bds::DataCollate, 164
- ~DataFile
  - Bds::DataFile, 174
- ~DataFileBds
  - Bds::DataFileBds, 190
- ~DataFileCssData
  - Bds::DataFileCssData, 201
- ~DataFileSeed
  - Bds::DataFileSeed, 227
- ~DataFormats
  - Bds::DataFormats, 240
- ~ResponseObj
  - Bds::ResponseObj, 287
- a
  - Bds::Fir, 252
- AccessGroup
  - Bds::AccessGroup, 42
- accessGroupDelete
  - Bds::AdminAccess, 49
- accessGroupGetList
  - Bds::AdminAccess, 49
- accessGroupUpdate
  - Bds::AdminAccess, 49
- address
  - Bds::User, 313
- addSource
  - Bds::DataCollate, 164
- AdminAccess
  - Bds::AdminAccess, 49
- alias
  - Bds::Source, 302
  - Bds::Station, 307
- ALLOW\_TIMESTAMP\_JITTER
  - BdsDataFileBds.cpp, 322
  - BdsDataFileCd.cpp, 326
- amplitude
  - Bds::Fap, 250
- apiVersion
  - Bds, 37
- appendDouble
  - Bds::BdsSeedType, 83
- appendExp
  - Bds::BdsSeedType, 83
- appendInt
  - Bds::BdsSeedType, 83
- appendString
  - Bds::BdsSeedType, 84
- appendStringVariable
  - Bds::BdsSeedType, 84
- array
  - Bds::DataInfo, 243
- ArrayChannel
  - Bds::ArrayChannel, 70
- arrayOffsetEast
  - Bds::Location, 263
- arrayOffsetNorth
  - Bds::Location, 263
- arrays
  - Bds::SelectionInfo, 294
- arraysAndStations
  - Bds::SelectionInfo, 294
- auth
  - Bds::CdChannel\_1v0, 91
  - Bds::CdPacketData, 97
- authKey
  - Bds::CdPacketData, 97
- authSize
  - Bds::CdPacketData, 97
- AvailFull
  - Bds, 25
- AvailNone
  - Bds, 25
- AvailPartial
  - Bds, 25
- AvailType



- Bds, [25](#)
- availType
  - Bds::DataAvail, [150](#)
- b
  - Bds::Fir, [252](#)
- baseSamplingFrequency
  - Bds::Digitiser, [247](#)
- Bds, [19](#)
  - apiVersion, [37](#)
  - AvailFull, [25](#)
  - AvailNone, [25](#)
  - AvailPartial, [25](#)
  - AvailType, [25](#)
  - bdsChannelGetName, [28](#)
  - bdsChannelGetTypeAux, [29](#)
  - bdsDataChannelInfo, [29](#)
  - bdsDataChannelOverallResponse, [29](#)
  - bdsDataChannelRef, [29](#), [30](#)
  - bdsDataFileSeedLogError, [30](#)
  - bdsDataFileSeedLogWarning, [30](#)
  - BdsDataFileVersion, [37](#)
  - bdsDataInfoFromInfo, [30](#)
  - bdsDataInfoMergeFlatten, [30](#)
  - bdsDataInfoSetTimeRange, [30](#)
  - BdsDataType, [25](#)
  - BdsDataTypeBlock, [25](#)
  - BdsDataTypeData, [26](#)
  - BdsDataTypeInfo, [26](#)
  - BdsDataTypeInfoExtra, [26](#)
  - bdsDumpChannelInfos, [31](#)
  - bdsDumpData, [31](#)
  - bdsDumpDataInfo, [31](#)
  - bdsDumpPoleZeros, [31](#)
  - bdsDumpSelection, [31](#)
  - bdsFileNameExpand, [32](#)
  - bdsInfoFromDataInfo, [32](#)
  - bdsPoleZeroGain, [32](#)
  - bdsPoleZeroGainPhase, [32](#)
  - bdsPoleZeroToFap, [33](#)
  - bdsSelectionChannelInfo, [33](#)
  - bdsStationAlias, [33](#)
  - bdsUnCompressCm8, [33](#)
  - bdsUnCompressSteim1, [33](#)
  - ChannelAuxLen, [38](#)
  - ChannelTypeLen, [38](#)
  - cm6Table, [38](#)
  - cm6TableRev, [38](#)
  - crc, [34](#)
  - crc64, [34](#)
  - crclnit, [34](#)
  - crclnitDone, [38](#)
  - crcVec, [39](#)
  - dataCalculateDifference, [34](#)
  - dataCalculateUnDifference, [34](#)
  - dataChecksum, [34](#)
  - dataCompressCm6, [35](#)
  - dataConvert, [35](#)
  - dataDeCompressCm6, [35](#)
  - DataFlagClipDataToChannels, [26](#)
  - DataFlagClipDataToTime, [26](#)
  - DataFlagMergeSegments, [26](#)
  - DataFlagNoMetadata, [26](#)
  - DataFlagNone, [26](#)
  - DataFlags, [26](#)
  - dataFormats, [39](#)
  - duplicateDump, [35](#)
  - ErrorDataQuality, [26](#)
  - ErrorNoMetaData, [26](#)
  - Errors, [26](#)
  - ErrorSlaveMode, [26](#)
  - ErrorTimeStamp, [26](#)
  - ErrorValidate, [26](#)
  - ErrorValidateBdsFudge, [27](#)
  - ErrorValidateDuplicate, [27](#)
  - ErrorValidateFilenameTime, [26](#)
  - ErrorValidateFix, [27](#)
  - ErrorValidateMetaData, [26](#)
  - ErrorValidateMissingBlocks, [26](#)
  - ErrorValidateReorder, [27](#)
  - ErrorValidateTimeBackwards, [26](#)
  - FileHeaderType, [27](#)
  - FileHeaderType\_Standard, [27](#)
  - FileHeaderType\_TapeDigitiser, [27](#)
  - fileNameTime, [36](#)
  - FileSampleType, [27](#)
  - FileSampleType\_Float32, [27](#)
  - FileSampleType\_Float64, [27](#)
  - FileSampleType\_Int16, [27](#)
  - FileSampleType\_Int32, [27](#)
  - FileSampleType\_Unknown, [27](#)
  - fixedString, [36](#)
  - fixedWidthValue, [36](#)
  - getHexString, [36](#)
  - Mode, [27](#)
  - ModeMaster, [27](#)
  - ModeSlave, [27](#)
  - NetworkNameLen, [39](#)
  - nullString, [36](#)
  - Priority, [27](#)
  - PriorityHigh, [28](#)
  - PriorityLow, [28](#)
  - PriorityNormal, [28](#)
  - record\_handler, [36](#)
  - removeCR, [37](#)
  - SampleFormat, [28](#)
  - SampleFormatFloat32, [28](#)
  - SampleFormatFloat64, [28](#)
  - SampleFormatInt16, [28](#)
  - SampleFormatInt24, [28](#)
  - SampleFormatInt32, [28](#)
  - SampleFormatUnknown, [28](#)
  - Scale, [39](#)
  - seedTime, [37](#)
  - seedTimeString, [37](#)
  - SelectionGroup, [28](#)
  - SelectionGroupData, [28](#)



- SelectionGroupDataWithCount, 28
- SelectionGroupMetaData, 28
- SourceLen, 39
- StationNameLen, 39
- stringFormat, 37
- Bds::AccessGroup, 41
  - AccessGroup, 42
  - endTime, 43
  - getMember, 42
  - getMembers, 42
  - getType, 42
  - group, 43
  - id, 43
  - network, 43
  - setMember, 42
  - setMembers, 43
  - startTime, 44
  - station, 44
- Bds::AdminAccess, 44
  - accessGroupDelete, 49
  - accessGroupGetList, 49
  - accessGroupUpdate, 49
  - AdminAccess, 49
  - calibrationDelete, 50
  - calibrationGetList, 50
  - calibrationUpdate, 50
  - changeDelete, 50
  - changeGetList, 50
  - changeGetListNumber, 51
  - changeGroupDelete, 51
  - changeGroupEnd, 51
  - changeGroupGetList, 51
  - changeGroupStart, 51
  - channelDelete, 52
  - channelGet, 52
  - channelGetList, 52
  - channelInstrumentDelete, 52
  - channelInstrumentGetList, 52
  - channelInstrumentUpdate, 53
  - channelUpdate, 53
  - clean, 53
  - connect, 53
  - dataAvailability, 53
  - databaseBackup, 54
  - databaseRestore, 54
  - dataChannelDelete, 54
  - dataChannelGetList, 54
  - dataChannelUpdate, 54
  - dataClose, 55
  - dataFileDelete, 55
  - dataFileGetList, 55
  - dataFileUpdate, 55
  - dataFormatGetList, 55
  - dataFormattedGetLength, 56
  - dataFormattedRead, 56
  - dataGetBlock, 56
  - dataGetChannelInfo, 56
  - dataGetInfo, 56
  - dataGetNotes, 57
  - dataGetWarnings, 57
  - dataOpen, 57
  - dataPutBlock, 58
  - dataSearch, 58
  - dataSeekBlock, 58
  - dataSetInfo, 58
  - digitiserDelete, 58
  - digitiserGet, 58
  - digitiserGetList, 59
  - digitiserUpdate, 59
  - getSelectionInfo, 59
  - getSelections, 59
  - getVersion, 59
  - groupDelete, 60
  - groupGetList, 60
  - groupUpdate, 60
  - locationDelete, 60
  - locationGetList, 60
  - locationUpdate, 61
  - logAppend, 61
  - logDelete, 61
  - logGetList, 61
  - logUpdate, 61
  - modeSet, 62
  - modeSnapshotPause, 62
  - networkDelete, 62
  - networkGetList, 62
  - networkUpdate, 62
  - noteDelete, 63
  - noteGetList, 63
  - noteReadDocument, 63
  - noteUpdate, 63
  - noteWriteDocument, 63
  - responseDelete, 64
  - responseGetList, 64
  - responseUpdate, 64
  - sensorDelete, 64
  - sensorGet, 64
  - sensorGetList, 65
  - sensorUpdate, 65
  - setUser, 65
  - setUserReal, 65
  - sourceDelete, 65
  - sourceGetList, 66
  - sourcePriorityDelete, 66
  - sourcePriorityGetList, 66
  - sourcePriorityUpdate, 66
  - sourceUpdate, 66
  - sqlQuery, 66
  - stationDelete, 67
  - stationGetList, 67
  - stationUpdate, 67
  - statisticsGet, 67
  - transactionEnd, 67
  - transactionStart, 68
  - userDelete, 68
  - userGet, 68



- userGetFromId, 68
  - userGetGroups, 68
  - userGetList, 68
  - userSet, 69
  - userUpdate, 69
  - validateUser, 69
- Bds::ArrayChannel, 69
  - ArrayChannel, 70
  - channel, 70
  - station, 70
- Bds::BdsDataBlock, 71
  - data, 71
  - header, 71
- Bds::BdsDataBlockHeader, 71
  - length, 72
  - packetOffset, 72
  - type, 72
- Bds::BdsDataBlockPos, 73
  - BdsDataBlockPos, 73
  - channel, 74
  - endTime, 74
  - numChannels, 74
  - numSamples, 74
  - operator<, 73
  - position, 74
  - segment, 74
  - startTime, 74
- Bds::BdsDataPacket, 75
  - ~BdsDataPacket, 76
  - BdsDataPacket, 75
  - clear, 76
  - dump, 76
  - getHeader, 76
  - reset, 76
  - setChecksumAndLength, 76
  - setHeader, 76
  - validateChecksum, 77
- Bds::BdsDataPacketHeader, 77
  - checksum, 77
  - endTime, 78
  - length, 78
  - sequence, 78
  - startTime, 78
  - streamlet, 78
  - type, 78
- Bds::BdsDataSegment, 79
  - BdsDataSegment, 79
  - blocks, 80
  - endTime, 80
  - numBlocks, 80
  - numSamples, 80
  - operator<, 79
  - sampleRate, 80
  - startTime, 80
- Bds::BdsDataStreamlet, 81
  - BdsDataStreamlet, 81
  - blocks, 81
  - channel, 81
  - numChannels, 82
  - packetNumber, 82
  - position, 82
  - segments, 82
- Bds::BdsSeedType, 82
  - appendDouble, 83
  - appendExp, 83
  - appendInt, 83
  - appendString, 84
  - appendStringVariable, 84
  - BdsSeedType, 83
  - getDouble, 84
  - getInt, 84
  - getString, 84
  - getStringVariable, 84
  - getUInt, 85
- Bds::Calibration, 85
  - Calibration, 86
  - calibrationFactor, 88
  - calibrationFrequency, 88
  - calibrationUnits, 88
  - channel, 88
  - depth, 89
  - endTime, 89
  - getMember, 87
  - getMembers, 87
  - getType, 87
  - horizontalAngle, 89
  - id, 89
  - name, 89
  - network, 89
  - samplingFrequency, 90
  - setMember, 87
  - setMembers, 88
  - source, 90
  - startTime, 90
  - station, 90
  - verticalAngle, 90
- Bds::CdChannel\_1v0, 91
  - auth, 91
  - calibrationFactor, 91
  - calibrationPeriod, 91
  - channel, 91
  - compress, 92
  - name, 92
  - spare0, 92
  - spare1, 92
- Bds::CdDataChannel, 92
  - channel, 93
  - data, 93
  - dataSize, 93
  - mode, 93
  - numSamples, 93
  - period, 93
  - startTime, 93
  - station, 94
  - status, 94
- Bds::CdDataFormatFrame\_1v0, 94



- channels, 94
- frameLength, 95
- frameType, 95
- maxFrameLength, 95
- numChannels, 95
- period, 95
- Bds::CdFlag, 95
  - CdFlag, 96
  - dead, 96
  - zeroed, 96
- Bds::CdPacketData, 96
  - auth, 97
  - authKey, 97
  - authSize, 97
  - channels, 97
  - crc, 97
  - creator, 98
  - destination, 98
  - frameType, 98
  - numChannels, 98
  - period, 98
  - sequenceNum, 98
  - series, 98
  - startTime, 98
  - trailerOffset, 99
- Bds::Change, 99
  - Change, 100
  - changeGroupId, 101
  - getMember, 100
  - getMembers, 100
  - getType, 100
  - id, 101
  - rowId, 101
  - setMember, 101
  - setMembers, 101
  - table, 101
  - time, 102
  - type, 102
- Bds::ChangeGroup, 102
  - ChangeGroup, 103
  - description, 104
  - getMember, 103
  - getMembers, 103
  - getType, 104
  - id, 104
  - setMember, 104
  - setMembers, 104
  - time, 105
  - title, 105
  - type, 105
  - user, 105
- Bds::Channel, 106
  - Channel, 107
  - channel, 108
  - channelAux, 108
  - channelType, 108
  - dataType, 108
  - description, 109
  - endTime, 109
  - getMember, 107
  - getMembers, 107
  - getType, 107
  - id, 109
  - network, 109
  - setMember, 108
  - setMembers, 108
  - startTime, 109
  - station, 109
- Bds::ChannelInfo, 110
  - calibration, 111
  - channel, 111
  - ChannelInfo, 111
  - dataType, 111
  - digitiser, 112
  - endTime, 112
  - location, 112
  - responses, 112
  - sensor, 112
  - source, 112
  - startTime, 113
  - station, 113
- Bds::ChannelInfos, 113
  - ChannelInfos, 114
  - channels, 114
- Bds::ChannelInstrument, 114
  - channelId, 116
  - ChannelInstrument, 115
  - digitiserId, 117
  - endTime, 117
  - getMember, 115
  - getMembers, 116
  - getType, 116
  - id, 117
  - sensorId, 117
  - setMember, 116
  - setMembers, 116
  - source, 117
  - startTime, 117
- Bds::ChannelName, 118
  - channel, 119
  - ChannelName, 118
  - network, 119
  - source, 119
  - station, 119
- Bds::CleanOptions, 120
  - changes, 120
  - CleanOptions, 120
  - deletedFiles, 120
  - logs, 121
- Bds::CompressSteim1, 121
  - clear, 122
  - CompressSteim1, 121
  - setByteOrder, 122
  - unCompress, 122
- Bds::DataAccess, 122
  - calibrationGetList, 125



- channelGetList, [125](#)
- channelInstrumentGetList, [125](#)
- clean, [126](#)
- connect, [126](#)
- DataAccess, [125](#)
- dataAvailability, [126](#)
- databaseBackup, [126](#)
- dataChannelGetList, [126](#)
- dataClose, [127](#)
- dataFileGetList, [127](#)
- dataFormatGetList, [127](#)
- dataFormattedGetLength, [127](#)
- dataFormattedRead, [127](#)
- dataGetBlock, [128](#)
- dataGetChannelInfo, [128](#)
- dataGetInfo, [128](#)
- dataGetNotes, [128](#)
- dataGetWarnings, [128](#)
- dataOpen, [129](#)
- dataSearch, [129](#)
- dataSeekBlock, [129](#)
- digitiserGet, [129](#)
- digitiserGetList, [130](#)
- getSelectionInfo, [130](#)
- getSelections, [130](#)
- getVersion, [130](#)
- groupGetList, [130](#)
- locationGetList, [131](#)
- logAppend, [131](#)
- logUpdate, [131](#)
- modeSet, [131](#)
- modeSnapshotPause, [131](#)
- networkGetList, [132](#)
- noteGetList, [132](#)
- noteReadDocument, [132](#)
- noteUpdate, [132](#)
- noteWriteDocument, [132](#)
- responseGetList, [133](#)
- sensorGet, [133](#)
- sensorGetList, [133](#)
- setUser, [133](#)
- setUserReal, [133](#)
- sourceGetList, [134](#)
- sourcePriorityGetList, [134](#)
- stationGetList, [134](#)
- statisticsGet, [134](#)
- userGet, [134](#)
- userGetFromId, [134](#)
- userGetGroups, [135](#)
- userSet, [135](#)
- validateUser, [135](#)
- Bds::DataAddAccess, [135](#)
- calibrationGetList, [138](#)
- channelGetList, [138](#)
- channelInstrumentGetList, [139](#)
- clean, [139](#)
- connect, [139](#)
- DataAddAccess, [139](#)
- dataAvailability, [139](#)
- databaseBackup, [139](#)
- dataChannelGetList, [140](#)
- dataClose, [140](#)
- dataFileGetList, [140](#)
- dataFormatGetList, [140](#)
- dataFormattedGetLength, [140](#)
- dataFormattedRead, [141](#)
- dataGetBlock, [141](#)
- dataGetChannelInfo, [141](#)
- dataGetInfo, [141](#)
- dataGetNotes, [141](#)
- dataGetWarnings, [142](#)
- dataOpen, [142](#)
- dataPutBlock, [142](#)
- dataSearch, [142](#)
- dataSeekBlock, [143](#)
- dataSetInfo, [143](#)
- digitiserGet, [143](#)
- digitiserGetList, [143](#)
- getSelectionInfo, [143](#)
- getSelections, [144](#)
- getVersion, [144](#)
- groupGetList, [144](#)
- locationGetList, [144](#)
- logAppend, [144](#)
- logUpdate, [145](#)
- modeSet, [145](#)
- modeSnapshotPause, [145](#)
- networkGetList, [145](#)
- noteGetList, [145](#)
- noteReadDocument, [146](#)
- noteUpdate, [146](#)
- noteWriteDocument, [146](#)
- responseGetList, [146](#)
- sensorGet, [146](#)
- sensorGetList, [147](#)
- setUser, [147](#)
- setUserReal, [147](#)
- sourceGetList, [147](#)
- sourcePriorityGetList, [147](#)
- stationGetList, [148](#)
- statisticsGet, [148](#)
- userGet, [148](#)
- userGetFromId, [148](#)
- userGetGroups, [148](#)
- userSet, [149](#)
- validateUser, [149](#)
- Bds::DataAvail, [149](#)
- availType, [150](#)
- DataAvail, [150](#)
- endTime, [150](#)
- startTime, [150](#)
- Bds::DataAvailChan, [151](#)
- channel, [152](#)
- DataAvailChan, [151](#)
- endTime, [152](#)
- network, [152](#)



- segments, [152](#)
- source, [152](#)
- startTime, [153](#)
- station, [153](#)
- Bds::DataBlock, [153](#)
  - channelData, [154](#)
  - channelNumber, [154](#)
  - DataBlock, [154](#)
  - endTime, [154](#)
  - info, [155](#)
  - segmentNumber, [155](#)
  - startTime, [155](#)
- Bds::DataBlockPos, [155](#)
  - DataBlockPos, [156](#)
  - endTime, [156](#)
  - numSamples, [156](#)
  - operator<, [156](#)
  - order, [157](#)
  - position, [157](#)
  - ref, [157](#)
  - startTime, [157](#)
- Bds::DataChannel, [157](#)
  - channel, [160](#)
  - DataChannel, [159](#)
  - dataFileChannel, [160](#)
  - dataFileId, [161](#)
  - endTime, [161](#)
  - getMember, [159](#)
  - getMembers, [159](#)
  - getType, [160](#)
  - id, [161](#)
  - importFilename, [161](#)
  - importFormat, [161](#)
  - importStartTime, [161](#)
  - info, [162](#)
  - network, [162](#)
  - numBlocks, [162](#)
  - numSamples, [162](#)
  - sampleFormat, [162](#)
  - sampleRate, [162](#)
  - setMember, [160](#)
  - setMembers, [160](#)
  - source, [163](#)
  - startTime, [163](#)
  - station, [163](#)
- Bds::DataCollate, [163](#)
  - ~DataCollate, [164](#)
  - addSource, [164](#)
  - DataCollate, [164](#)
  - readData, [164](#)
- Bds::DataError, [165](#)
  - DataError, [166](#)
  - getErrorNumber, [166](#)
  - getString, [166](#)
  - getTitle, [167](#)
  - mergeDataInfo, [167](#)
  - num, [167](#)
  - ochannel, [168](#)
  - odescription, [168](#)
  - oendTime, [168](#)
  - oerrorNumber, [169](#)
  - ofilename, [169](#)
  - onetwork, [169](#)
  - operator int, [167](#)
  - osource, [169](#)
  - ostartTime, [169](#)
  - ostation, [169](#)
  - otitle, [170](#)
  - ouser, [170](#)
  - set, [167](#)
  - setString, [167](#)
  - setStringUser, [168](#)
  - str, [168](#)
- Bds::DataFile, [170](#)
  - ~DataFile, [174](#)
  - close, [174](#)
  - dataErrorFixup, [174](#)
  - DataFile, [174](#)
  - DataOrder, [173](#)
  - DataOrderAll, [173](#)
  - DataOrderChannel, [173](#)
  - DataOrderSample, [173](#)
  - DataOrderUnknown, [173](#)
  - duplicateCheck, [175](#)
  - end, [175](#)
  - FeatureCanRead, [173](#)
  - FeatureCanWrite, [173](#)
  - FeatureNone, [173](#)
  - Features, [173](#)
  - fileNameProcess, [175](#)
  - flush, [175](#)
  - getDataOrder, [175](#)
  - getFeatures, [176](#)
  - getFileName, [176](#)
  - getFilePosition, [176](#)
  - getFormat, [176](#)
  - getFormats, [176](#)
  - getInfo, [177](#)
  - getMetaData, [177](#)
  - init, [177](#)
  - ofile, [179](#)
  - ofilename, [179](#)
  - ofilenameTime, [179](#)
  - oformat, [180](#)
  - omode, [180](#)
  - open, [177](#)
  - readData, [177](#)
  - ReadOptionDeleteDuplicates, [173](#)
  - ReadOptionFileNameProcess, [173](#)
  - ReadOptionIgnoreSamplerate, [173](#)
  - ReadOptionInfoExtra, [173](#)
  - ReadOptionNone, [173](#)
  - ReadOptionPrintBlocks, [173](#)
  - ReadOptionReorder, [173](#)
  - ReadOptionsList, [173](#)
  - ReadOptionValidate, [173](#)



- ReadOptionValidateCorruptions, 173
- seekBlock, 178
- setFormat, 178
- setInfo, 178
- start, 178
- timeCompare, 179
- writeData, 179
- WriteOptionNoMetadata, 174
- WriteOptionNone, 174
- WriteOptionSensorData, 174
- WriteOptionsList, 174
- Bds::DataFileAd22, 180
  - DataFileAd22, 181
  - getDataOrder, 181
  - getFeatures, 181
  - getFormats, 181
  - getInfo, 182
  - readData, 182
- Bds::DataFileAscii, 182
  - DataFileAscii, 183
  - end, 183
  - getDataOrder, 184
  - getFeatures, 184
  - getFormats, 184
  - open, 184
  - setFormat, 184
  - setInfo, 185
  - start, 185
  - writeData, 185
- Bds::DataFileBdrs, 186
  - DataFileBdrs, 186
  - getDataOrder, 187
  - getFeatures, 187
  - getFormats, 187
  - getInfo, 187
  - readData, 187
- Bds::DataFileBds, 188
  - ~DataFileBds, 190
  - close, 190
  - DataFileBds, 190
  - DefaultBlockSize, 190
  - flush, 190
  - getDataOrder, 191
  - getDiskBlockSize, 191
  - getFormats, 191
  - getInfo, 191
  - open, 191
  - PackFormat, 190
  - PackFormat\_CM, 190
  - PackFormat\_SM, 190
  - PackFormat\_SM\_CC, 190
  - PackFormat\_Unknown, 190
  - readData, 192
  - seekBlock, 192
  - setDiskBlockSize, 192
  - setFormat, 192
  - setInfo, 193
  - StreamsMax, 189
  - writeData, 193
- Bds::DataFileBknas, 193
  - DataFileBknas, 194
  - getFormats, 194
  - open, 194
  - setInfo, 195
  - writeData, 195
- Bds::DataFileCd, 195
  - DataFileCd, 196
  - getDataOrder, 196
  - getFeatures, 196
  - getFormats, 197
  - getInfo, 197
  - readData, 197
- Bds::DataFileCss, 198
  - DataFileCss, 198
  - getDataOrder, 199
  - getFeatures, 199
  - getFormats, 199
  - getInfo, 199
  - readData, 199
- Bds::DataFileCssData, 200
  - ~DataFileCssData, 201
  - calibrationFactor, 201
  - calibrationFreq, 202
  - chan, 202
  - chanid, 202
  - clip, 202
  - commId, 202
  - DataFileCssData, 201
  - datatype, 202
  - dirName, 202
  - endTime, 202
  - file, 203
  - fileName, 203
  - fileOffset, 203
  - instType, 203
  - jdate, 203
  - loadDate, 203
  - nsamp, 203
  - sampleBigEndian, 203
  - sampleFormat, 204
  - sampleRate, 204
  - sampleSize, 204
  - segtype, 204
  - set, 201
  - sta, 204
  - startTime, 204
  - wfid, 204
- Bds::DataFileGcf, 205
  - DataFileGcf, 206
  - getDataOrder, 206
  - getFeatures, 206
  - getFormats, 206
  - getInfo, 206
  - readData, 206
- Bds::DataFileIms, 207
  - close, 208



- DataFileIms, 208
- end, 208
- getDataOrder, 208
- getFeatures, 209
- getFormats, 209
- getMetaData, 209
- open, 209
- setInfo, 209
- start, 210
- writeData, 210
- Bds::DataFileInfo, 210
  - comment, 213
  - DataFileInfo, 211
  - endTime, 213
  - format, 213
  - getMember, 212
  - getMembers, 212
  - getType, 212
  - id, 213
  - importTime, 213
  - importUserId, 213
  - location, 214
  - setMember, 212
  - setMembers, 212
  - startTime, 214
  - state, 214
  - url, 214
- Bds::DataFileLac, 215
  - DataFileLac, 215
  - getDataOrder, 216
  - getFeatures, 216
  - getFormats, 216
  - getInfo, 216
  - readData, 216
- Bds::DataFileLog, 217
  - DataFileLog, 218
  - end, 218
  - getDataOrder, 218
  - getFeatures, 218
  - getFormats, 219
  - getInfo, 219
  - open, 219
  - readData, 219
  - setFormat, 219
  - setInfo, 220
  - start, 220
  - writeData, 220
- Bds::DataFileOptions, 221
  - DataFileOptions, 221
  - oignoreBlockList, 222
  - ooptionList, 222
  - operator int, 221
  - operator | =, 221
- Bds::DataFileResponse, 222
  - DataFileResponse, 223
  - getFeatures, 223
  - getFormats, 223
  - getMetaData, 223
  - setInfo, 223
- Bds::DataFileSac, 224
  - DataFileSac, 225
  - getFeatures, 225
  - getFormats, 225
  - setInfo, 225
- Bds::DataFileSeed, 226
  - ~DataFileSeed, 227
  - close, 227
  - DataFileSeed, 227
  - end, 227
  - getDataOrder, 227
  - getFeatures, 228
  - getFormats, 228
  - getInfo, 228
  - msrFileWrite, 228
  - omsrErr, 230
  - onoLock, 230
  - readData, 228
  - setFormat, 229
  - setInfo, 229
  - start, 229
  - writeData, 229
- Bds::DataFileTapeDigitiser, 230
  - DataFileTapeDigitiser, 231
  - getFormats, 231
  - getInfo, 231
  - open, 231
  - readData, 232
- Bds::DataFileWra, 232
  - DataFileWra, 233
  - getDataOrder, 233
  - getFeatures, 233
  - getFormats, 233
  - getInfo, 234
  - readData, 234
  - setFormat, 234
- Bds::DataFileWraAgso, 235
  - DataFileWraAgso, 235
  - getDataOrder, 236
  - getFeatures, 236
  - getFormats, 236
  - getInfo, 236
  - readData, 236
- Bds::DataFormat, 237
  - DataFormat, 238
  - dataRead, 238
  - dataWrite, 238
  - description, 238
  - extension, 238
  - metaDataRead, 239
  - metaDataWrite, 239
  - names, 239
- Bds::DataFormats, 239
  - ~DataFormats, 240
  - DataFormats, 240
  - findFormat, 240
  - formatGet, 240



- formatList, [240](#)
- Bds::DataHandle, [241](#)
  - dataFileId, [241](#)
  - DataHandle, [241](#)
  - handle, [242](#)
- Bds::DataInfo, [242](#)
  - array, [243](#)
  - channels, [243](#)
  - DataInfo, [243](#)
  - description, [243](#)
  - endTime, [244](#)
  - info, [244](#)
  - infoExtra, [244](#)
  - startTime, [244](#)
  - synchronous, [244](#)
  - warnings, [244](#)
- Bds::Digitiser, [245](#)
  - baseSamplingFrequency, [247](#)
  - Digitiser, [246](#)
  - endTime, [247](#)
  - gain, [248](#)
  - getMember, [246](#)
  - getMembers, [246](#)
  - getType, [247](#)
  - id, [248](#)
  - initialSamplingFrequency, [248](#)
  - name, [248](#)
  - numberChannels, [248](#)
  - serialNumber, [248](#)
  - setMember, [247](#)
  - setMembers, [247](#)
  - shared, [249](#)
  - startTime, [249](#)
  - type, [249](#)
- Bds::Fap, [249](#)
  - amplitude, [250](#)
  - Fap, [250](#)
  - frequency, [250](#)
  - phase, [250](#)
- Bds::Fir, [251](#)
  - a, [252](#)
  - b, [252](#)
  - Fir, [251](#)
- Bds::FirEntry, [252](#)
  - coefficient, [253](#)
  - error, [253](#)
  - FirEntry, [253](#)
- Bds::GcfChannel, [253](#)
  - channel, [254](#)
  - format, [254](#)
  - sampleRate, [254](#)
  - streamId, [254](#)
  - systemId, [254](#)
  - type, [254](#)
- Bds::Group, [255](#)
  - description, [257](#)
  - getMember, [256](#)
  - getMembers, [256](#)
  - getType, [256](#)
  - Group, [255](#)
  - group, [257](#)
  - id, [257](#)
  - setMember, [256](#)
  - setMembers, [256](#)
- Bds::ListRange, [257](#)
  - getMember, [258](#)
  - getMembers, [258](#)
  - getType, [259](#)
  - ListRange, [258](#)
  - number, [259](#)
  - reverse, [259](#)
  - setMember, [259](#)
  - setMembers, [259](#)
  - start, [260](#)
- Bds::Location, [260](#)
  - arrayOffsetEast, [263](#)
  - arrayOffsetNorth, [263](#)
  - datum, [263](#)
  - elevation, [263](#)
  - endTime, [263](#)
  - getMember, [262](#)
  - getMembers, [262](#)
  - getType, [262](#)
  - id, [263](#)
  - latitude, [264](#)
  - Location, [261](#)
  - longitude, [264](#)
  - network, [264](#)
  - setMember, [262](#)
  - setMembers, [262](#)
  - startTime, [264](#)
  - station, [264](#)
- Bds::Log, [265](#)
  - description, [267](#)
  - getMember, [266](#)
  - getMembers, [266](#)
  - getType, [266](#)
  - id, [267](#)
  - Log, [266](#)
  - priority, [267](#)
  - setMember, [266](#)
  - setMembers, [267](#)
  - subSystem, [267](#)
  - time, [268](#)
  - title, [268](#)
  - type, [268](#)
- Bds::LogSelect, [268](#)
  - LogSelect, [269](#)
  - priority, [269](#)
  - startTime, [269](#)
  - subSystem, [269](#)
  - type, [270](#)
- Bds::Network, [270](#)
  - description, [272](#)
  - getMember, [271](#)
  - getMembers, [271](#)



- getType, 271
  - id, 272
  - Network, 271
  - network, 272
  - setMember, 271
  - setMembers, 272
  - stations, 272
- Bds::Note, 273
  - channel, 276
  - dataFileId, 276
  - description, 276
  - docFormat, 276
  - docUrl, 276
  - endTime, 276
  - errorNumber, 277
  - getMember, 275
  - getMembers, 275
  - getType, 275
  - id, 277
  - importFilename, 277
  - network, 277
  - Note, 274
  - setMember, 275
  - setMembers, 275
  - source, 277
  - startTime, 277
  - station, 278
  - timeAdded, 278
  - title, 278
  - type, 278
  - user, 278
- Bds::Point, 279
  - Point, 279
  - x, 279
  - y, 279
- Bds::PoleZero, 280
  - poles, 281
  - PoleZero, 280
  - zeros, 281
- Bds::Response, 281
  - channel, 283
  - decimation, 283
  - description, 283
  - endTime, 284
  - faps, 284
  - fir, 284
  - gain, 284
  - gainFrequency, 284
  - id, 284
  - measured, 285
  - name, 285
  - network, 285
  - poleZeros, 285
  - Response, 283
  - sampleRate, 285
  - source, 285
  - stage, 286
  - stageType, 286
  - startTime, 286
  - station, 286
  - symmetry, 286
  - type, 286
- Bds::ResponseObj, 287
  - ~ResponseObj, 287
  - getString, 288
  - ResponseObj, 287
  - setString, 288
- Bds::Selection, 288
  - calibrationName, 290
  - channelId, 290
  - channels, 290
  - completeSegments, 290
  - digitiserId, 290
  - endTime, 290
  - id, 291
  - range, 291
  - Selection, 289
  - sensorId, 291
  - sensorOldId, 291
  - startTime, 291
- Bds::SelectionChannel, 292
  - channel, 292
  - network, 292
  - SelectionChannel, 292
  - source, 293
  - station, 293
- Bds::SelectionInfo, 293
  - arrays, 294
  - arraysAndStations, 294
  - channels, 294
  - endTime, 294
  - networks, 295
  - numDataChannels, 295
  - SelectionInfo, 294
  - sources, 295
  - startTime, 295
  - stations, 295
- Bds::Sensor, 296
  - endTime, 298
  - gain, 298
  - gainUnits, 298
  - getMember, 297
  - getMembers, 297
  - getType, 297
  - id, 298
  - name, 299
  - numberChannels, 299
  - oldId, 299
  - Sensor, 297
  - serialNumber, 299
  - setMember, 298
  - setMembers, 298
  - shared, 299
  - startTime, 299
  - type, 300
- Bds::Source, 300



- alias, [302](#)
- description, [302](#)
- getMember, [301](#)
- getMembers, [301](#)
- getType, [302](#)
- id, [303](#)
- setMember, [302](#)
- setMembers, [302](#)
- Source, [301](#)
- source, [303](#)
- sourceMeta, [303](#)
- Bds::SourcePriority, [303](#)
  - endTime, [305](#)
  - getMember, [304](#)
  - getMembers, [305](#)
  - getType, [305](#)
  - id, [306](#)
  - priority, [306](#)
  - setMember, [305](#)
  - setMembers, [305](#)
  - source, [306](#)
  - SourcePriority, [304](#)
  - startTime, [306](#)
- Bds::Station, [306](#)
  - alias, [307](#)
  - channels, [308](#)
  - description, [308](#)
  - id, [308](#)
  - name, [308](#)
  - Station, [307](#)
  - type, [308](#)
- Bds::TimePeriod, [309](#)
  - endTime, [311](#)
  - getMember, [310](#)
  - getMembers, [310](#)
  - getType, [310](#)
  - setMember, [310](#)
  - setMembers, [310](#)
  - startTime, [311](#)
  - TimePeriod, [309](#)
- Bds::User, [311](#)
  - address, [313](#)
  - email, [314](#)
  - enabled, [314](#)
  - getMember, [312](#)
  - getMembers, [313](#)
  - getType, [313](#)
  - groups, [314](#)
  - id, [314](#)
  - name, [314](#)
  - password, [314](#)
  - setMember, [313](#)
  - setMembers, [313](#)
  - telephone, [315](#)
  - User, [312](#)
  - user, [315](#)
- BdsC.cc, [343](#)
- BdsC.d, [343](#)
- BdsC.h, [343](#)
- bdsChannelGetName
  - Bds, [28](#)
- bdsChannelGetTypeAux
  - Bds, [29](#)
- BdsD.cc, [344](#)
- BdsD.d, [344](#)
- BdsD.h, [344](#)
- BdsDataBlockPos
  - Bds::BdsDataBlockPos, [73](#)
- bdsDataChannelInfo
  - Bds, [29](#)
- bdsDataChannelOverallResponse
  - Bds, [29](#)
- bdsDataChannelRef
  - Bds, [29](#), [30](#)
- BdsDataFileAd22.cpp
  - DEBUG\_VELATRACK, [320](#)
- BdsDataFileBds.cpp
  - ALLOW\_TIMESTAMP\_JITTER, [322](#)
  - dl2printf, [322](#)
  - dl3printf, [323](#)
  - dlprintf, [323](#)
  - LDEBUG, [323](#)
  - LDEBUG2, [323](#)
  - LDEBUG3, [323](#)
  - TIMESTAMP\_JITTER, [323](#)
- BdsDataFileBknas.cpp
  - clip, [325](#)
- BdsDataFileCd.cpp
  - ALLOW\_TIMESTAMP\_JITTER, [326](#)
  - dprintf, [326](#)
  - ErrorFormatNoDataFormat, [327](#)
  - htonll, [326](#)
  - INCLUDE\_CHANNEL\_AUTH, [327](#)
  - LDEBUG, [327](#)
  - MULTIPLE\_SEGMENT, [327](#)
  - ntohll, [327](#)
  - SEGMENT\_GAP, [327](#)
  - TIMESTAMP\_JITTER, [327](#)
- BdsDataFileGcf.cpp
  - DEBUG, [329](#)
  - TEST\_REORDER, [329](#)
- BdsDataFileResponse.cpp
  - dprintf, [333](#)
  - LDEBUG, [333](#)
- BdsDataFileSeed.cpp
  - DEBUG, [339](#)
  - DEBUG\_BLOCKETTE, [339](#)
  - DEBUG\_BLOCKS, [339](#)
  - dlprintf, [339](#)
  - FILL\_BLOCKS, [339](#)
  - ROUND\_TIMESTAMPS\_TO\_10US, [339](#)
- bdsDataFileSeedLogError
  - Bds, [30](#)
- bdsDataFileSeedLogWarning
  - Bds, [30](#)
- BdsDataFileVersion



- Bds, [37](#)
- BdsDataFileWraAgso.cpp
  - parseStringFixedFields, [336](#)
- bdsDataInfoFromInfo
  - Bds, [30](#)
- bdsDataInfoMergeFlatten
  - Bds, [30](#)
- bdsDataInfoSetTimeRange
  - Bds, [30](#)
- BdsDataPacket
  - Bds::BdsDataPacket, [75](#)
- BdsDataSegment
  - Bds::BdsDataSegment, [79](#)
- BdsDataStreamlet
  - Bds::BdsDataStreamlet, [81](#)
- BdsDataType
  - Bds, [25](#)
- BdsDataTypeBlock
  - Bds, [25](#)
- BdsDataTypeData
  - Bds, [26](#)
- BdsDataTypeInfo
  - Bds, [26](#)
- BdsDataTypeInfoExtra
  - Bds, [26](#)
- bdsDumpChannelInfos
  - Bds, [31](#)
- bdsDumpData
  - Bds, [31](#)
- bdsDumpDataInfo
  - Bds, [31](#)
- bdsDumpPoleZeros
  - Bds, [31](#)
- bdsDumpSelection
  - Bds, [31](#)
- bdsFileNameExpand
  - Bds, [32](#)
- bdsInfoFromDataInfo
  - Bds, [32](#)
- BdsLib.cpp, [347](#)
- BdsLib.d, [349](#)
- BdsLib.dox, [349](#)
- BdsLib.h, [349](#)
- bdsPoleZeroGain
  - Bds, [32](#)
- bdsPoleZeroGainPhase
  - Bds, [32](#)
- bdsPoleZeroToFap
  - Bds, [33](#)
- BdsS.cc, [351](#)
- BdsS.d, [351](#)
- BdsSeedType
  - Bds::BdsSeedType, [83](#)
- bdsSelectionChannelInfo
  - Bds, [33](#)
- bdsStationAlias
  - Bds, [33](#)
- BdsT.cc, [351](#)
- bdsUnCompressCm8
  - Bds, [33](#)
- bdsUnCompressSteim1
  - Bds, [33](#)
- blocks
  - Bds::BdsDataSegment, [80](#)
  - Bds::BdsDataStreamlet, [81](#)
- Calibration
  - Bds::Calibration, [86](#)
- calibration
  - Bds::ChannelInfo, [111](#)
- calibrationDelete
  - Bds::AdminAccess, [50](#)
- calibrationFactor
  - Bds::Calibration, [88](#)
  - Bds::CdChannel\_1v0, [91](#)
  - Bds::DataFileCssData, [201](#)
- calibrationFreq
  - Bds::DataFileCssData, [202](#)
- calibrationFrequency
  - Bds::Calibration, [88](#)
- calibrationGetList
  - Bds::AdminAccess, [50](#)
  - Bds::DataAccess, [125](#)
  - Bds::DataAddAccess, [138](#)
- calibrationName
  - Bds::Selection, [290](#)
- calibrationPeriod
  - Bds::CdChannel\_1v0, [91](#)
- calibrationUnits
  - Bds::Calibration, [88](#)
- calibrationUpdate
  - Bds::AdminAccess, [50](#)
- canada\_compress
  - canada\_compress.h, [342](#)
- canada\_compress.h
  - canada\_compress, [342](#)
  - canada\_uncompress, [343](#)
  - CANCOMP\_CORRUPT, [342](#)
  - CANCOMP\_ERR, [342](#)
  - CANCOMP\_EXCEED, [342](#)
  - CANCOMP\_NOT\_20, [342](#)
  - CANCOMP\_SUCCESS, [342](#)
- canada\_uncompress
  - canada\_compress.h, [343](#)
- CANCOMP\_CORRUPT
  - canada\_compress.h, [342](#)
- CANCOMP\_ERR
  - canada\_compress.h, [342](#)
- CANCOMP\_EXCEED
  - canada\_compress.h, [342](#)
- CANCOMP\_NOT\_20
  - canada\_compress.h, [342](#)
- CANCOMP\_SUCCESS
  - canada\_compress.h, [342](#)
- CdFlag
  - Bds::CdFlag, [96](#)
- chan



- Bds::DataFileCssData, 202
- Change
  - Bds::Change, 100
- changeDelete
  - Bds::AdminAccess, 50
- changeGetList
  - Bds::AdminAccess, 50
- changeGetListNumber
  - Bds::AdminAccess, 51
- ChangeGroup
  - Bds::ChangeGroup, 103
- changeGroupDelete
  - Bds::AdminAccess, 51
- changeGroupEnd
  - Bds::AdminAccess, 51
- changeGroupGetList
  - Bds::AdminAccess, 51
- changeGroupId
  - Bds::Change, 101
- changeGroupStart
  - Bds::AdminAccess, 51
- changes
  - Bds::CleanOptions, 120
- chanid
  - Bds::DataFileCssData, 202
- Channel
  - Bds::Channel, 107
- channel
  - Bds::ArrayChannel, 70
  - Bds::BdsDataBlockPos, 74
  - Bds::BdsDataStreamlet, 81
  - Bds::Calibration, 88
  - Bds::CdChannel\_1v0, 91
  - Bds::CdDataChannel, 93
  - Bds::Channel, 108
  - Bds::ChannelInfo, 111
  - Bds::ChannelName, 119
  - Bds::DataAvailChan, 152
  - Bds::DataChannel, 160
  - Bds::GcfChannel, 254
  - Bds::Note, 276
  - Bds::Response, 283
  - Bds::SelectionChannel, 292
- channelAux
  - Bds::Channel, 108
- ChannelAuxLen
  - Bds, 38
- channelData
  - Bds::DataBlock, 154
- channelDelete
  - Bds::AdminAccess, 52
- channelGet
  - Bds::AdminAccess, 52
- channelGetList
  - Bds::AdminAccess, 52
  - Bds::DataAccess, 125
  - Bds::DataAddAccess, 138
- channelId
  - Bds::ChannelInstrument, 116
  - Bds::Selection, 290
- ChannelInfo
  - Bds::ChannelInfo, 111
- ChannelInfos
  - Bds::ChannelInfos, 114
- ChannelInstrument
  - Bds::ChannelInstrument, 115
- channelInstrumentDelete
  - Bds::AdminAccess, 52
- channelInstrumentGetList
  - Bds::AdminAccess, 52
  - Bds::DataAccess, 125
  - Bds::DataAddAccess, 139
- channelInstrumentUpdate
  - Bds::AdminAccess, 53
- ChannelName
  - Bds::ChannelName, 118
- channelNumber
  - Bds::DataBlock, 154
- channels
  - Bds::CdDataFormatFrame\_1v0, 94
  - Bds::CdPacketData, 97
  - Bds::ChannelInfos, 114
  - Bds::DataInfo, 243
  - Bds::Selection, 290
  - Bds::SelectionInfo, 294
  - Bds::Station, 308
- channelType
  - Bds::Channel, 108
- ChannelTypeLen
  - Bds, 38
- channelUpdate
  - Bds::AdminAccess, 53
- checksum
  - Bds::BdsDataPacketHeader, 77
- clean
  - Bds::AdminAccess, 53
  - Bds::DataAccess, 126
  - Bds::DataAddAccess, 139
- CleanOptions
  - Bds::CleanOptions, 120
- clear
  - Bds::BdsDataPacket, 76
  - Bds::CompressSteim1, 122
- clip
  - Bds::DataFileCssData, 202
  - BdsDataFileBknas.cpp, 325
- close
  - Bds::DataFile, 174
  - Bds::DataFileBds, 190
  - Bds::DataFileIms, 208
  - Bds::DataFileSeed, 227
- cm6Table
  - Bds, 38
- cm6TableRev
  - Bds, 38
- coefficient



- Bds::FirEntry, 253
- comment
  - Bds::DataFileInfo, 213
- commId
  - Bds::DataFileCssData, 202
- completeSegments
  - Bds::Selection, 290
- compress
  - Bds::CdChannel\_1v0, 92
- CompressSteim1
  - Bds::CompressSteim1, 121
- connect
  - Bds::AdminAccess, 53
  - Bds::DataAccess, 126
  - Bds::DataAddAccess, 139
- crc
  - Bds, 34
  - Bds::CdPacketData, 97
- crc64
  - Bds, 34
- crclnit
  - Bds, 34
- crclnitDone
  - Bds, 38
- crcVec
  - Bds, 39
- creator
  - Bds::CdPacketData, 98
- data
  - Bds::BdsDataBlock, 71
  - Bds::CdDataChannel, 93
- DataAccess
  - Bds::DataAccess, 125
- DataAddAccess
  - Bds::DataAddAccess, 138
- DataAvail
  - Bds::DataAvail, 150
- dataAvailability
  - Bds::AdminAccess, 53
  - Bds::DataAccess, 126
  - Bds::DataAddAccess, 139
- DataAvailChan
  - Bds::DataAvailChan, 151
- databaseBackup
  - Bds::AdminAccess, 54
  - Bds::DataAccess, 126
  - Bds::DataAddAccess, 139
- databaseRestore
  - Bds::AdminAccess, 54
- DataBlock
  - Bds::DataBlock, 154
- DataBlockPos
  - Bds::DataBlockPos, 156
- dataCalculateDifference
  - Bds, 34
- dataCalculateUnDifference
  - Bds, 34
- DataChannel
  - Bds::DataChannel, 159
- dataChannelDelete
  - Bds::AdminAccess, 54
- dataChannelGetList
  - Bds::AdminAccess, 54
  - Bds::DataAccess, 126
  - Bds::DataAddAccess, 140
- dataChannelUpdate
  - Bds::AdminAccess, 54
- dataChecksum
  - Bds, 34
- dataClose
  - Bds::AdminAccess, 55
  - Bds::DataAccess, 127
  - Bds::DataAddAccess, 140
- DataCollate
  - Bds::DataCollate, 164
- dataCompressCm6
  - Bds, 35
- dataConvert
  - Bds, 35
- dataDeCompressCm6
  - Bds, 35
- DataError
  - Bds::DataError, 166
- dataErrorFixup
  - Bds::DataFile, 174
- DataFile
  - Bds::DataFile, 174
- DataFileAd22
  - Bds::DataFileAd22, 181
- DataFileAscii
  - Bds::DataFileAscii, 183
- DataFileBdrs
  - Bds::DataFileBdrs, 186
- DataFileBds
  - Bds::DataFileBds, 190
- DataFileBknas
  - Bds::DataFileBknas, 194
- DataFileCd
  - Bds::DataFileCd, 196
- dataFileChannel
  - Bds::DataChannel, 160
- DataFileCss
  - Bds::DataFileCss, 198
- DataFileCssData
  - Bds::DataFileCssData, 201
- dataFileDelete
  - Bds::AdminAccess, 55
- DataFileGcf
  - Bds::DataFileGcf, 206
- dataFileGetList
  - Bds::AdminAccess, 55
  - Bds::DataAccess, 127
  - Bds::DataAddAccess, 140
- dataFileId
  - Bds::DataChannel, 161
  - Bds::DataHandle, 241



- Bds::Note, [276](#)
- DataFileIms
  - Bds::DataFileIms, [208](#)
- DataFileInfo
  - Bds::DataFileInfo, [211](#)
- DataFileLac
  - Bds::DataFileLac, [215](#)
- DataFileLog
  - Bds::DataFileLog, [218](#)
- DataFileOptions
  - Bds::DataFileOptions, [221](#)
- DataFileResponse
  - Bds::DataFileResponse, [223](#)
- DataFileSac
  - Bds::DataFileSac, [225](#)
- DataFileSeed
  - Bds::DataFileSeed, [227](#)
- DataFileTapeDigitiser
  - Bds::DataFileTapeDigitiser, [231](#)
- dataFileUpdate
  - Bds::AdminAccess, [55](#)
- DataFileWra
  - Bds::DataFileWra, [233](#)
- DataFileWraAgso
  - Bds::DataFileWraAgso, [235](#)
- DataFlagClipDataToChannels
  - Bds, [26](#)
- DataFlagClipDataToTime
  - Bds, [26](#)
- DataFlagMergeSegments
  - Bds, [26](#)
- DataFlagNoMetadata
  - Bds, [26](#)
- DataFlagNone
  - Bds, [26](#)
- DataFlags
  - Bds, [26](#)
- DataFormat
  - Bds::DataFormat, [238](#)
- dataFormatGetList
  - Bds::AdminAccess, [55](#)
  - Bds::DataAccess, [127](#)
  - Bds::DataAddAccess, [140](#)
- DataFormats
  - Bds::DataFormats, [240](#)
- dataFormats
  - Bds, [39](#)
- dataFormattedGetLength
  - Bds::AdminAccess, [56](#)
  - Bds::DataAccess, [127](#)
  - Bds::DataAddAccess, [140](#)
- dataFormattedRead
  - Bds::AdminAccess, [56](#)
  - Bds::DataAccess, [127](#)
  - Bds::DataAddAccess, [141](#)
- dataGetBlock
  - Bds::AdminAccess, [56](#)
  - Bds::DataAccess, [128](#)
- Bds::DataAddAccess, [141](#)
- dataGetChannelInfo
  - Bds::AdminAccess, [56](#)
  - Bds::DataAccess, [128](#)
  - Bds::DataAddAccess, [141](#)
- dataGetInfo
  - Bds::AdminAccess, [56](#)
  - Bds::DataAccess, [128](#)
  - Bds::DataAddAccess, [141](#)
- dataGetNotes
  - Bds::AdminAccess, [57](#)
  - Bds::DataAccess, [128](#)
  - Bds::DataAddAccess, [141](#)
- dataGetWarnings
  - Bds::AdminAccess, [57](#)
  - Bds::DataAccess, [128](#)
  - Bds::DataAddAccess, [142](#)
- DataHandle
  - Bds::DataHandle, [241](#)
- DataInfo
  - Bds::DataInfo, [243](#)
- dataOpen
  - Bds::AdminAccess, [57](#)
  - Bds::DataAccess, [129](#)
  - Bds::DataAddAccess, [142](#)
- DataOrder
  - Bds::DataFile, [173](#)
- DataOrderAll
  - Bds::DataFile, [173](#)
- DataOrderChannel
  - Bds::DataFile, [173](#)
- DataOrderSample
  - Bds::DataFile, [173](#)
- DataOrderUnknown
  - Bds::DataFile, [173](#)
- dataPutBlock
  - Bds::AdminAccess, [58](#)
  - Bds::DataAddAccess, [142](#)
- dataRead
  - Bds::DataFormat, [238](#)
- dataSearch
  - Bds::AdminAccess, [58](#)
  - Bds::DataAccess, [129](#)
  - Bds::DataAddAccess, [142](#)
- dataSeekBlock
  - Bds::AdminAccess, [58](#)
  - Bds::DataAccess, [129](#)
  - Bds::DataAddAccess, [143](#)
- dataSetInfo
  - Bds::AdminAccess, [58](#)
  - Bds::DataAddAccess, [143](#)
- dataSize
  - Bds::CdDataChannel, [93](#)
- dataType
  - Bds::Channel, [108](#)
  - Bds::ChannelInfo, [111](#)
- datatype
  - Bds::DataFileCssData, [202](#)



- dataWrite
  - Bds::DataFormat, [238](#)
- datum
  - Bds::Location, [263](#)
- dead
  - Bds::CdFlag, [96](#)
- DEBUG
  - BdsDataFileGcf.cpp, [329](#)
  - BdsDataFileSeed.cpp, [339](#)
- DEBUG\_BLOCKETTE
  - BdsDataFileSeed.cpp, [339](#)
- DEBUG\_BLOCKS
  - BdsDataFileSeed.cpp, [339](#)
- DEBUG\_VELATRACK
  - BdsDataFileAd22.cpp, [320](#)
- decimation
  - Bds::Response, [283](#)
- DefaultBlockSize
  - Bds::DataFileBds, [190](#)
- deletedFiles
  - Bds::CleanOptions, [120](#)
- depth
  - Bds::Calibration, [89](#)
- description
  - Bds::ChangeGroup, [104](#)
  - Bds::Channel, [109](#)
  - Bds::DataFormat, [238](#)
  - Bds::DataInfo, [243](#)
  - Bds::Group, [257](#)
  - Bds::Log, [267](#)
  - Bds::Network, [272](#)
  - Bds::Note, [276](#)
  - Bds::Response, [283](#)
  - Bds::Source, [302](#)
  - Bds::Station, [308](#)
- destination
  - Bds::CdPacketData, [98](#)
- Digitiser
  - Bds::Digitiser, [246](#)
- digitiser
  - Bds::ChannelInfo, [112](#)
- digitiserDelete
  - Bds::AdminAccess, [58](#)
- digitiserGet
  - Bds::AdminAccess, [58](#)
  - Bds::DataAccess, [129](#)
  - Bds::DataAddAccess, [143](#)
- digitiserGetList
  - Bds::AdminAccess, [59](#)
  - Bds::DataAccess, [130](#)
  - Bds::DataAddAccess, [143](#)
- digitiserId
  - Bds::ChannelInstrument, [117](#)
  - Bds::Selection, [290](#)
- digitiserUpdate
  - Bds::AdminAccess, [59](#)
- dirName
  - Bds::DataFileCssData, [202](#)
- dl2printf
  - BdsDataFileBds.cpp, [322](#)
- dl3printf
  - BdsDataFileBds.cpp, [323](#)
- dlprintf
  - BdsDataFileBds.cpp, [323](#)
  - BdsDataFileSeed.cpp, [339](#)
- docFormat
  - Bds::Note, [276](#)
- docUrl
  - Bds::Note, [276](#)
- dprintf
  - BdsDataFileCd.cpp, [326](#)
  - BdsDataFileResponse.cpp, [333](#)
- dump
  - Bds::BdsDataPacket, [76](#)
- duplicateCheck
  - Bds::DataFile, [175](#)
- duplicateDump
  - Bds, [35](#)
- elevation
  - Bds::Location, [263](#)
- email
  - Bds::User, [314](#)
- enabled
  - Bds::User, [314](#)
- end
  - Bds::DataFile, [175](#)
  - Bds::DataFileAscii, [183](#)
  - Bds::DataFileIms, [208](#)
  - Bds::DataFileLog, [218](#)
  - Bds::DataFileSeed, [227](#)
- endTime
  - Bds::AccessGroup, [43](#)
  - Bds::BdsDataBlockPos, [74](#)
  - Bds::BdsDataPacketHeader, [78](#)
  - Bds::BdsDataSegment, [80](#)
  - Bds::Calibration, [89](#)
  - Bds::Channel, [109](#)
  - Bds::ChannelInfo, [112](#)
  - Bds::ChannelInstrument, [117](#)
  - Bds::DataAvail, [150](#)
  - Bds::DataAvailChan, [152](#)
  - Bds::DataBlock, [154](#)
  - Bds::DataBlockPos, [156](#)
  - Bds::DataChannel, [161](#)
  - Bds::DataFileCssData, [202](#)
  - Bds::DataFileInfo, [213](#)
  - Bds::DataInfo, [244](#)
  - Bds::Digitiser, [247](#)
  - Bds::Location, [263](#)
  - Bds::Note, [276](#)
  - Bds::Response, [284](#)
  - Bds::Selection, [290](#)
  - Bds::SelectionInfo, [294](#)
  - Bds::Sensor, [298](#)
  - Bds::SourcePriority, [305](#)
  - Bds::TimePeriod, [311](#)



- error
  - Bds::FirEntry, [253](#)
- ErrorDataQuality
  - Bds, [26](#)
- ErrorFormatNoDataFormat
  - BdsDataFileCd.cpp, [327](#)
- ErrorNoMetaData
  - Bds, [26](#)
- errorNumber
  - Bds::Note, [277](#)
- Errors
  - Bds, [26](#)
- ErrorSlaveMode
  - Bds, [26](#)
- ErrorTimeStamp
  - Bds, [26](#)
- ErrorValidate
  - Bds, [26](#)
- ErrorValidateBdsFudge
  - Bds, [27](#)
- ErrorValidateDuplicate
  - Bds, [27](#)
- ErrorValidateFilenameTime
  - Bds, [26](#)
- ErrorValidateFix
  - Bds, [27](#)
- ErrorValidateMetaData
  - Bds, [26](#)
- ErrorValidateMissingBlocks
  - Bds, [26](#)
- ErrorValidateReorder
  - Bds, [27](#)
- ErrorValidateTimeBackwards
  - Bds, [26](#)
- extension
  - Bds::DataFormat, [238](#)
- Fap
  - Bds::Fap, [250](#)
- faps
  - Bds::Response, [284](#)
- FeatureCanRead
  - Bds::DataFile, [173](#)
- FeatureCanWrite
  - Bds::DataFile, [173](#)
- FeatureNone
  - Bds::DataFile, [173](#)
- Features
  - Bds::DataFile, [173](#)
- file
  - Bds::DataFileCssData, [203](#)
- FileHeaderType
  - Bds, [27](#)
- FileHeaderType\_Standard
  - Bds, [27](#)
- FileHeaderType\_TapeDigitiser
  - Bds, [27](#)
- fileName
  - Bds::DataFileCssData, [203](#)
- fileNameProcess
  - Bds::DataFile, [175](#)
- fileNameTime
  - Bds, [36](#)
- fileOffset
  - Bds::DataFileCssData, [203](#)
- FileSampleType
  - Bds, [27](#)
- FileSampleType\_Float32
  - Bds, [27](#)
- FileSampleType\_Float64
  - Bds, [27](#)
- FileSampleType\_Int16
  - Bds, [27](#)
- FileSampleType\_Int32
  - Bds, [27](#)
- FileSampleType\_Unknown
  - Bds, [27](#)
- FILL\_BLOCKS
  - BdsDataFileSeed.cpp, [339](#)
- findFormat
  - Bds::DataFormats, [240](#)
- Fir
  - Bds::Fir, [251](#)
- fir
  - Bds::Response, [284](#)
- FirEntry
  - Bds::FirEntry, [253](#)
- fixedString
  - Bds, [36](#)
- fixedWidthValue
  - Bds, [36](#)
- flush
  - Bds::DataFile, [175](#)
  - Bds::DataFileBds, [190](#)
- format
  - Bds::DataFileInfo, [213](#)
  - Bds::GcfChannel, [254](#)
- formatGet
  - Bds::DataFormats, [240](#)
- formatList
  - Bds::DataFormats, [240](#)
- frameLength
  - Bds::CdDataFormatFrame\_1v0, [95](#)
- frameType
  - Bds::CdDataFormatFrame\_1v0, [95](#)
  - Bds::CdPacketData, [98](#)
- frequency
  - Bds::Fap, [250](#)
- gain
  - Bds::Digitiser, [248](#)
  - Bds::Response, [284](#)
  - Bds::Sensor, [298](#)
- gainFrequency
  - Bds::Response, [284](#)
- gainUnits
  - Bds::Sensor, [298](#)
- getDataOrder



- Bds::DataFile, 175
- Bds::DataFileAd22, 181
- Bds::DataFileAscii, 184
- Bds::DataFileBdrs, 187
- Bds::DataFileBds, 191
- Bds::DataFileCd, 196
- Bds::DataFileCss, 199
- Bds::DataFileGcf, 206
- Bds::DataFileImms, 208
- Bds::DataFileLac, 216
- Bds::DataFileLog, 218
- Bds::DataFileSeed, 227
- Bds::DataFileWra, 233
- Bds::DataFileWraAgso, 236
- getDiskBlockSize
  - Bds::DataFileBds, 191
- getDouble
  - Bds::BdsSeedType, 84
- getErrorNumber
  - Bds::DataError, 166
- getFeatures
  - Bds::DataFile, 176
  - Bds::DataFileAd22, 181
  - Bds::DataFileAscii, 184
  - Bds::DataFileBdrs, 187
  - Bds::DataFileCd, 196
  - Bds::DataFileCss, 199
  - Bds::DataFileGcf, 206
  - Bds::DataFileImms, 209
  - Bds::DataFileLac, 216
  - Bds::DataFileLog, 218
  - Bds::DataFileResponse, 223
  - Bds::DataFileSac, 225
  - Bds::DataFileSeed, 228
  - Bds::DataFileWra, 233
  - Bds::DataFileWraAgso, 236
- getFileName
  - Bds::DataFile, 176
- getPosition
  - Bds::DataFile, 176
- getFormat
  - Bds::DataFile, 176
- getFormats
  - Bds::DataFile, 176
  - Bds::DataFileAd22, 181
  - Bds::DataFileAscii, 184
  - Bds::DataFileBdrs, 187
  - Bds::DataFileBds, 191
  - Bds::DataFileBknas, 194
  - Bds::DataFileCd, 197
  - Bds::DataFileCss, 199
  - Bds::DataFileGcf, 206
  - Bds::DataFileImms, 209
  - Bds::DataFileLac, 216
  - Bds::DataFileLog, 219
  - Bds::DataFileResponse, 223
  - Bds::DataFileSac, 225
  - Bds::DataFileSeed, 228
  - Bds::DataFileTapeDigitiser, 231
  - Bds::DataFileWra, 233
  - Bds::DataFileWraAgso, 236
- getHeader
  - Bds::BdsDataPacket, 76
- getHexString
  - Bds, 36
- getInfo
  - Bds::DataFile, 177
  - Bds::DataFileAd22, 182
  - Bds::DataFileBdrs, 187
  - Bds::DataFileBds, 191
  - Bds::DataFileCd, 197
  - Bds::DataFileCss, 199
  - Bds::DataFileGcf, 206
  - Bds::DataFileLac, 216
  - Bds::DataFileLog, 219
  - Bds::DataFileSeed, 228
  - Bds::DataFileTapeDigitiser, 231
  - Bds::DataFileWra, 234
  - Bds::DataFileWraAgso, 236
- getInt
  - Bds::BdsSeedType, 84
- getMember
  - Bds::AccessGroup, 42
  - Bds::Calibration, 87
  - Bds::Change, 100
  - Bds::ChangeGroup, 103
  - Bds::Channel, 107
  - Bds::ChannelInstrument, 115
  - Bds::DataChannel, 159
  - Bds::DataFileInfo, 212
  - Bds::Digitiser, 246
  - Bds::Group, 256
  - Bds::ListRange, 258
  - Bds::Location, 262
  - Bds::Log, 266
  - Bds::Network, 271
  - Bds::Note, 275
  - Bds::Sensor, 297
  - Bds::Source, 301
  - Bds::SourcePriority, 304
  - Bds::TimePeriod, 310
  - Bds::User, 312
- getMembers
  - Bds::AccessGroup, 42
  - Bds::Calibration, 87
  - Bds::Change, 100
  - Bds::ChangeGroup, 103
  - Bds::Channel, 107
  - Bds::ChannelInstrument, 116
  - Bds::DataChannel, 159
  - Bds::DataFileInfo, 212
  - Bds::Digitiser, 246
  - Bds::Group, 256
  - Bds::ListRange, 258
  - Bds::Location, 262
  - Bds::Log, 266



- Bds::Network, 271
- Bds::Note, 275
- Bds::Sensor, 297
- Bds::Source, 301
- Bds::SourcePriority, 305
- Bds::TimePeriod, 310
- Bds::User, 313
- getMetaData
  - Bds::DataFile, 177
  - Bds::DataFileIms, 209
  - Bds::DataFileResponse, 223
- getSelectionInfo
  - Bds::AdminAccess, 59
  - Bds::DataAccess, 130
  - Bds::DataAddAccess, 143
- getSelections
  - Bds::AdminAccess, 59
  - Bds::DataAccess, 130
  - Bds::DataAddAccess, 144
- getString
  - Bds::BdsSeedType, 84
  - Bds::DataError, 166
  - Bds::ResponseObj, 288
- getStringVariable
  - Bds::BdsSeedType, 84
- getTitle
  - Bds::DataError, 167
- getType
  - Bds::AccessGroup, 42
  - Bds::Calibration, 87
  - Bds::Change, 100
  - Bds::ChangeGroup, 104
  - Bds::Channel, 107
  - Bds::ChannelInstrument, 116
  - Bds::DataChannel, 160
  - Bds::DataFileInfo, 212
  - Bds::Digitiser, 247
  - Bds::Group, 256
  - Bds::ListRange, 259
  - Bds::Location, 262
  - Bds::Log, 266
  - Bds::Network, 271
  - Bds::Note, 275
  - Bds::Sensor, 297
  - Bds::Source, 302
  - Bds::SourcePriority, 305
  - Bds::TimePeriod, 310
  - Bds::User, 313
- getUInt
  - Bds::BdsSeedType, 85
- getVersion
  - Bds::AdminAccess, 59
  - Bds::DataAccess, 130
  - Bds::DataAddAccess, 144
- Group
  - Bds::Group, 255
- group
  - Bds::AccessGroup, 43
  - Bds::Group, 257
- groupDelete
  - Bds::AdminAccess, 60
- groupGetList
  - Bds::AdminAccess, 60
  - Bds::DataAccess, 130
  - Bds::DataAddAccess, 144
- groups
  - Bds::User, 314
- groupUpdate
  - Bds::AdminAccess, 60
- handle
  - Bds::DataHandle, 242
- header
  - Bds::BdsDataBlock, 71
- horizontalAngle
  - Bds::Calibration, 89
- htonll
  - BdsDataFileCd.cpp, 326
- id
  - Bds::AccessGroup, 43
  - Bds::Calibration, 89
  - Bds::Change, 101
  - Bds::ChangeGroup, 104
  - Bds::Channel, 109
  - Bds::ChannelInstrument, 117
  - Bds::DataChannel, 161
  - Bds::DataFileInfo, 213
  - Bds::Digitiser, 248
  - Bds::Group, 257
  - Bds::Location, 263
  - Bds::Log, 267
  - Bds::Network, 272
  - Bds::Note, 277
  - Bds::Response, 284
  - Bds::Selection, 291
  - Bds::Sensor, 298
  - Bds::Source, 303
  - Bds::SourcePriority, 306
  - Bds::Station, 308
  - Bds::User, 314
- importFilename
  - Bds::DataChannel, 161
  - Bds::Note, 277
- importFormat
  - Bds::DataChannel, 161
- importStartTime
  - Bds::DataChannel, 161
- importTime
  - Bds::DataFileInfo, 213
- importUserId
  - Bds::DataFileInfo, 213
- INCLUDE\_CHANNEL\_AUTH
  - BdsDataFileCd.cpp, 327
- info
  - Bds::DataBlock, 155
  - Bds::DataChannel, 162



- Bds::DataInfo, 244
- infoExtra
  - Bds::DataInfo, 244
- init
  - Bds::DataFile, 177
- initialSamplingFrequency
  - Bds::Digitiser, 248
- instType
  - Bds::DataFileCssData, 203
- jdate
  - Bds::DataFileCssData, 203
- latitude
  - Bds::Location, 264
- LDEBUG
  - BdsDataFileBds.cpp, 323
  - BdsDataFileCd.cpp, 327
  - BdsDataFileResponse.cpp, 333
- LDEBUG2
  - BdsDataFileBds.cpp, 323
- LDEBUG3
  - BdsDataFileBds.cpp, 323
- length
  - Bds::BdsDataBlockHeader, 72
  - Bds::BdsDataPacketHeader, 78
- ListRange
  - Bds::ListRange, 258
- loadDate
  - Bds::DataFileCssData, 203
- Location
  - Bds::Location, 261
- location
  - Bds::ChannelInfo, 112
  - Bds::DataFileInfo, 214
- locationDelete
  - Bds::AdminAccess, 60
- locationGetList
  - Bds::AdminAccess, 60
  - Bds::DataAccess, 131
  - Bds::DataAddAccess, 144
- locationUpdate
  - Bds::AdminAccess, 61
- Log
  - Bds::Log, 266
- logAppend
  - Bds::AdminAccess, 61
  - Bds::DataAccess, 131
  - Bds::DataAddAccess, 144
- logDelete
  - Bds::AdminAccess, 61
- logGetList
  - Bds::AdminAccess, 61
- logs
  - Bds::CleanOptions, 121
- LogSelect
  - Bds::LogSelect, 269
- logUpdate
  - Bds::AdminAccess, 61
- Bds::DataAccess, 131
- Bds::DataAddAccess, 145
- longitude
  - Bds::Location, 264
- maxFrameLength
  - Bds::CdDataFormatFrame\_1v0, 95
- measured
  - Bds::Response, 285
- mergeDataInfo
  - Bds::DataError, 167
- metaDataRead
  - Bds::DataFormat, 239
- metaDataWrite
  - Bds::DataFormat, 239
- Mode
  - Bds, 27
- mode
  - Bds::CdDataChannel, 93
- ModeMaster
  - Bds, 27
- modeSet
  - Bds::AdminAccess, 62
  - Bds::DataAccess, 131
  - Bds::DataAddAccess, 145
- ModeSlave
  - Bds, 27
- modeSnapshotPause
  - Bds::AdminAccess, 62
  - Bds::DataAccess, 131
  - Bds::DataAddAccess, 145
- msrFileWrite
  - Bds::DataFileSeed, 228
- MULTIPLE\_SEGMENT
  - BdsDataFileCd.cpp, 327
- name
  - Bds::Calibration, 89
  - Bds::CdChannel\_1v0, 92
  - Bds::Digitiser, 248
  - Bds::Response, 285
  - Bds::Sensor, 299
  - Bds::Station, 308
  - Bds::User, 314
- names
  - Bds::DataFormat, 239
- Network
  - Bds::Network, 271
- network
  - Bds::AccessGroup, 43
  - Bds::Calibration, 89
  - Bds::Channel, 109
  - Bds::ChannelName, 119
  - Bds::DataAvailChan, 152
  - Bds::DataChannel, 162
  - Bds::Location, 264
  - Bds::Network, 272
  - Bds::Note, 277
  - Bds::Response, 285



- Bds::SelectionChannel, 292
- networkDelete
  - Bds::AdminAccess, 62
- networkGetList
  - Bds::AdminAccess, 62
  - Bds::DataAccess, 132
  - Bds::DataAddAccess, 145
- NetworkNameLen
  - Bds, 39
- networks
  - Bds::SelectionInfo, 295
- networkUpdate
  - Bds::AdminAccess, 62
- Note
  - Bds::Note, 274
- noteDelete
  - Bds::AdminAccess, 63
- noteGetList
  - Bds::AdminAccess, 63
  - Bds::DataAccess, 132
  - Bds::DataAddAccess, 145
- noteReadDocument
  - Bds::AdminAccess, 63
  - Bds::DataAccess, 132
  - Bds::DataAddAccess, 146
- noteUpdate
  - Bds::AdminAccess, 63
  - Bds::DataAccess, 132
  - Bds::DataAddAccess, 146
- noteWriteDocument
  - Bds::AdminAccess, 63
  - Bds::DataAccess, 132
  - Bds::DataAddAccess, 146
- nsamp
  - Bds::DataFileCssData, 203
- ntohl
  - BdsDataFileCd.cpp, 327
- nullString
  - Bds, 36
- num
  - Bds::DataError, 167
- number
  - Bds::ListRange, 259
- numberChannels
  - Bds::Digitiser, 248
  - Bds::Sensor, 299
- numBlocks
  - Bds::BdsDataSegment, 80
  - Bds::DataChannel, 162
- numChannels
  - Bds::BdsDataBlockPos, 74
  - Bds::BdsDataStreamlet, 82
  - Bds::CdDataFormatFrame\_1v0, 95
  - Bds::CdPacketData, 98
- numDataChannels
  - Bds::SelectionInfo, 295
- numSamples
  - Bds::BdsDataBlockPos, 74
- Bds::BdsDataSegment, 80
- Bds::CdDataChannel, 93
- Bds::DataBlockPos, 156
- Bds::DataChannel, 162
- ochannel
  - Bds::DataError, 168
- odescription
  - Bds::DataError, 168
- oendTime
  - Bds::DataError, 168
- oerrorNumber
  - Bds::DataError, 169
- ofile
  - Bds::DataFile, 179
- ofilename
  - Bds::DataFile, 179
- ofilename
  - Bds::DataError, 169
- ofilenameTime
  - Bds::DataFile, 179
- oformat
  - Bds::DataFile, 180
- oignoreBlockList
  - Bds::DataFileOptions, 222
- oldId
  - Bds::Sensor, 299
- omode
  - Bds::DataFile, 180
- omsrErr
  - Bds::DataFileSeed, 230
- onetwork
  - Bds::DataError, 169
- onoLock
  - Bds::DataFileSeed, 230
- ooptionList
  - Bds::DataFileOptions, 222
- open
  - Bds::DataFile, 177
  - Bds::DataFileAscii, 184
  - Bds::DataFileBds, 191
  - Bds::DataFileBknas, 194
  - Bds::DataFileImms, 209
  - Bds::DataFileLog, 219
  - Bds::DataFileTapeDigitiser, 231
- operator int
  - Bds::DataError, 167
  - Bds::DataFileOptions, 221
- operator<
  - Bds::BdsDataBlockPos, 73
  - Bds::BdsDataSegment, 79
  - Bds::DataBlockPos, 156
- operator |=
  - Bds::DataFileOptions, 221
- order
  - Bds::DataBlockPos, 157
- osource
  - Bds::DataError, 169
- ostartTime



- Bds::DataError, 169
- ostation
  - Bds::DataError, 169
- otitle
  - Bds::DataError, 170
- ouser
  - Bds::DataError, 170
- packetNumber
  - Bds::BdsDataStreamlet, 82
- packetOffset
  - Bds::BdsDataBlockHeader, 72
- PackFormat
  - Bds::DataFileBds, 190
- PackFormat\_CM
  - Bds::DataFileBds, 190
- PackFormat\_SM
  - Bds::DataFileBds, 190
- PackFormat\_SM\_CC
  - Bds::DataFileBds, 190
- PackFormat\_Unknown
  - Bds::DataFileBds, 190
- parseStringFixedFields
  - BdsDataFileWraAgso.cpp, 336
- password
  - Bds::User, 314
- period
  - Bds::CdDataChannel, 93
  - Bds::CdDataFormatFrame\_1v0, 95
  - Bds::CdPacketData, 98
- phase
  - Bds::Fap, 250
- Point
  - Bds::Point, 279
- poles
  - Bds::PoleZero, 281
- PoleZero
  - Bds::PoleZero, 280
- poleZeros
  - Bds::Response, 285
- position
  - Bds::BdsDataBlockPos, 74
  - Bds::BdsDataStreamlet, 82
  - Bds::DataBlockPos, 157
- Priority
  - Bds, 27
- priority
  - Bds::Log, 267
  - Bds::LogSelect, 269
  - Bds::SourcePriority, 306
- PriorityHigh
  - Bds, 28
- PriorityLow
  - Bds, 28
- PriorityNormal
  - Bds, 28
- range
  - Bds::Selection, 291
- readData
  - Bds::DataCollate, 164
  - Bds::DataFile, 177
  - Bds::DataFileAd22, 182
  - Bds::DataFileBdrs, 187
  - Bds::DataFileBds, 192
  - Bds::DataFileCd, 197
  - Bds::DataFileCss, 199
  - Bds::DataFileGcf, 206
  - Bds::DataFileLac, 216
  - Bds::DataFileLog, 219
  - Bds::DataFileSeed, 228
  - Bds::DataFileTapeDigitiser, 232
  - Bds::DataFileWra, 234
  - Bds::DataFileWraAgso, 236
- ReadOptionDeleteDuplicates
  - Bds::DataFile, 173
- ReadOptionFileNameProcess
  - Bds::DataFile, 173
- ReadOptionIgnoreSamplerate
  - Bds::DataFile, 173
- ReadOptionInfoExtra
  - Bds::DataFile, 173
- ReadOptionNone
  - Bds::DataFile, 173
- ReadOptionPrintBlocks
  - Bds::DataFile, 173
- ReadOptionReorder
  - Bds::DataFile, 173
- ReadOptionsList
  - Bds::DataFile, 173
- ReadOptionValidate
  - Bds::DataFile, 173
- ReadOptionValidateCorruptions
  - Bds::DataFile, 173
- record\_handler
  - Bds, 36
- ref
  - Bds::DataBlockPos, 157
- removeCR
  - Bds, 37
- reset
  - Bds::BdsDataPacket, 76
- Response
  - Bds::Response, 283
- responseDelete
  - Bds::AdminAccess, 64
- responseGetList
  - Bds::AdminAccess, 64
  - Bds::DataAccess, 133
  - Bds::DataAddAccess, 146
- ResponseObj
  - Bds::ResponseObj, 287
- responses
  - Bds::ChannelInfo, 112
- responseUpdate
  - Bds::AdminAccess, 64
- reverse



- Bds::ListRange, [259](#)
- ROUND\_TIMESTAMPS\_TO\_10US
  - BdsDataFileSeed.cpp, [339](#)
- rowId
  - Bds::Change, [101](#)
- sampleBigEndian
  - Bds::DataFileCssData, [203](#)
- SampleFormat
  - Bds, [28](#)
- sampleFormat
  - Bds::DataChannel, [162](#)
  - Bds::DataFileCssData, [204](#)
- SampleFormatFloat32
  - Bds, [28](#)
- SampleFormatFloat64
  - Bds, [28](#)
- SampleFormatInt16
  - Bds, [28](#)
- SampleFormatInt24
  - Bds, [28](#)
- SampleFormatInt32
  - Bds, [28](#)
- SampleFormatUnknown
  - Bds, [28](#)
- sampleRate
  - Bds::BdsDataSegment, [80](#)
  - Bds::DataChannel, [162](#)
  - Bds::DataFileCssData, [204](#)
  - Bds::GcfChannel, [254](#)
  - Bds::Response, [285](#)
- sampleSize
  - Bds::DataFileCssData, [204](#)
- samplingFrequency
  - Bds::Calibration, [90](#)
- Scale
  - Bds, [39](#)
- seedTime
  - Bds, [37](#)
- seedTimeString
  - Bds, [37](#)
- seekBlock
  - Bds::DataFile, [178](#)
  - Bds::DataFileBds, [192](#)
- segment
  - Bds::BdsDataBlockPos, [74](#)
- SEGMENT\_GAP
  - BdsDataFileCd.cpp, [327](#)
- segmentNumber
  - Bds::DataBlock, [155](#)
- segments
  - Bds::BdsDataStreamlet, [82](#)
  - Bds::DataAvailChan, [152](#)
- segtype
  - Bds::DataFileCssData, [204](#)
- Selection
  - Bds::Selection, [289](#)
- SelectionChannel
  - Bds::SelectionChannel, [292](#)
- SelectionGroup
  - Bds, [28](#)
- SelectionGroupData
  - Bds, [28](#)
- SelectionGroupDataWithCount
  - Bds, [28](#)
- SelectionGroupMetaData
  - Bds, [28](#)
- SelectionInfo
  - Bds::SelectionInfo, [294](#)
- Sensor
  - Bds::Sensor, [297](#)
- sensor
  - Bds::ChannelInfo, [112](#)
- sensorDelete
  - Bds::AdminAccess, [64](#)
- sensorGet
  - Bds::AdminAccess, [64](#)
  - Bds::DataAccess, [133](#)
  - Bds::DataAddAccess, [146](#)
- sensorGetList
  - Bds::AdminAccess, [65](#)
  - Bds::DataAccess, [133](#)
  - Bds::DataAddAccess, [147](#)
- sensorId
  - Bds::ChannelInstrument, [117](#)
  - Bds::Selection, [291](#)
- sensorOldId
  - Bds::Selection, [291](#)
- sensorUpdate
  - Bds::AdminAccess, [65](#)
- sequence
  - Bds::BdsDataPacketHeader, [78](#)
- sequenceNum
  - Bds::CdPacketData, [98](#)
- serialNumber
  - Bds::Digitiser, [248](#)
  - Bds::Sensor, [299](#)
- series
  - Bds::CdPacketData, [98](#)
- set
  - Bds::DataError, [167](#)
  - Bds::DataFileCssData, [201](#)
- setByteOrder
  - Bds::CompressSteim1, [122](#)
- setChecksumAndLength
  - Bds::BdsDataPacket, [76](#)
- setDiskBlockSize
  - Bds::DataFileBds, [192](#)
- setFormat
  - Bds::DataFile, [178](#)
  - Bds::DataFileAscii, [184](#)
  - Bds::DataFileBds, [192](#)
  - Bds::DataFileLog, [219](#)
  - Bds::DataFileSeed, [229](#)
  - Bds::DataFileWra, [234](#)
- setHeader
  - Bds::BdsDataPacket, [76](#)



- setInfo
  - Bds::DataFile, 178
  - Bds::DataFileAscii, 185
  - Bds::DataFileBds, 193
  - Bds::DataFileBknas, 195
  - Bds::DataFileIms, 209
  - Bds::DataFileLog, 220
  - Bds::DataFileResponse, 223
  - Bds::DataFileSac, 225
  - Bds::DataFileSeed, 229
- setMember
  - Bds::AccessGroup, 42
  - Bds::Calibration, 87
  - Bds::Change, 101
  - Bds::ChangeGroup, 104
  - Bds::Channel, 108
  - Bds::ChannelInstrument, 116
  - Bds::DataChannel, 160
  - Bds::DataFileInfo, 212
  - Bds::Digitiser, 247
  - Bds::Group, 256
  - Bds::ListRange, 259
  - Bds::Location, 262
  - Bds::Log, 266
  - Bds::Network, 271
  - Bds::Note, 275
  - Bds::Sensor, 298
  - Bds::Source, 302
  - Bds::SourcePriority, 305
  - Bds::TimePeriod, 310
  - Bds::User, 313
- setMembers
  - Bds::AccessGroup, 43
  - Bds::Calibration, 88
  - Bds::Change, 101
  - Bds::ChangeGroup, 104
  - Bds::Channel, 108
  - Bds::ChannelInstrument, 116
  - Bds::DataChannel, 160
  - Bds::DataFileInfo, 212
  - Bds::Digitiser, 247
  - Bds::Group, 256
  - Bds::ListRange, 259
  - Bds::Location, 262
  - Bds::Log, 267
  - Bds::Network, 272
  - Bds::Note, 275
  - Bds::Sensor, 298
  - Bds::Source, 302
  - Bds::SourcePriority, 305
  - Bds::TimePeriod, 310
  - Bds::User, 313
- setString
  - Bds::DataError, 167
  - Bds::ResponseObj, 288
- setStringUser
  - Bds::DataError, 168
- setUser
  - Bds::AdminAccess, 65
  - Bds::DataAccess, 133
  - Bds::DataAddAccess, 147
- setUserReal
  - Bds::AdminAccess, 65
  - Bds::DataAccess, 133
  - Bds::DataAddAccess, 147
- shared
  - Bds::Digitiser, 249
  - Bds::Sensor, 299
- Source
  - Bds::Source, 301
- source
  - Bds::Calibration, 90
  - Bds::ChannelInfo, 112
  - Bds::ChannelInstrument, 117
  - Bds::ChannelName, 119
  - Bds::DataAvailChan, 152
  - Bds::DataChannel, 163
  - Bds::Note, 277
  - Bds::Response, 285
  - Bds::SelectionChannel, 293
  - Bds::Source, 303
  - Bds::SourcePriority, 306
- sourceDelete
  - Bds::AdminAccess, 65
- sourceGetList
  - Bds::AdminAccess, 66
  - Bds::DataAccess, 134
  - Bds::DataAddAccess, 147
- SourceLen
  - Bds, 39
- sourceMeta
  - Bds::Source, 303
- SourcePriority
  - Bds::SourcePriority, 304
- sourcePriorityDelete
  - Bds::AdminAccess, 66
- sourcePriorityGetList
  - Bds::AdminAccess, 66
  - Bds::DataAccess, 134
  - Bds::DataAddAccess, 147
- sourcePriorityUpdate
  - Bds::AdminAccess, 66
- sources
  - Bds::SelectionInfo, 295
- sourceUpdate
  - Bds::AdminAccess, 66
- spare0
  - Bds::CdChannel\_1v0, 92
- spare1
  - Bds::CdChannel\_1v0, 92
- sqlQuery
  - Bds::AdminAccess, 66
- sta
  - Bds::DataFileCssData, 204
- stage
  - Bds::Response, 286



- stageType
  - Bds::Response, [286](#)
- start
  - Bds::DataFile, [178](#)
  - Bds::DataFileAscii, [185](#)
  - Bds::DataFileIms, [210](#)
  - Bds::DataFileLog, [220](#)
  - Bds::DataFileSeed, [229](#)
  - Bds::ListRange, [260](#)
- startTime
  - Bds::AccessGroup, [44](#)
  - Bds::BdsDataBlockPos, [74](#)
  - Bds::BdsDataPacketHeader, [78](#)
  - Bds::BdsDataSegment, [80](#)
  - Bds::Calibration, [90](#)
  - Bds::CdDataChannel, [93](#)
  - Bds::CdPacketData, [98](#)
  - Bds::Channel, [109](#)
  - Bds::ChannelInfo, [113](#)
  - Bds::ChannelInstrument, [117](#)
  - Bds::DataAvail, [150](#)
  - Bds::DataAvailChan, [153](#)
  - Bds::DataBlock, [155](#)
  - Bds::DataBlockPos, [157](#)
  - Bds::DataChannel, [163](#)
  - Bds::DataFileCssData, [204](#)
  - Bds::DataFileInfo, [214](#)
  - Bds::DataInfo, [244](#)
  - Bds::Digitiser, [249](#)
  - Bds::Location, [264](#)
  - Bds::LogSelect, [269](#)
  - Bds::Note, [277](#)
  - Bds::Response, [286](#)
  - Bds::Selection, [291](#)
  - Bds::SelectionInfo, [295](#)
  - Bds::Sensor, [299](#)
  - Bds::SourcePriority, [306](#)
  - Bds::TimePeriod, [311](#)
- state
  - Bds::DataFileInfo, [214](#)
- Station
  - Bds::Station, [307](#)
- station
  - Bds::AccessGroup, [44](#)
  - Bds::ArrayChannel, [70](#)
  - Bds::Calibration, [90](#)
  - Bds::CdDataChannel, [94](#)
  - Bds::Channel, [109](#)
  - Bds::ChannelInfo, [113](#)
  - Bds::ChannelName, [119](#)
  - Bds::DataAvailChan, [153](#)
  - Bds::DataChannel, [163](#)
  - Bds::Location, [264](#)
  - Bds::Note, [278](#)
  - Bds::Response, [286](#)
  - Bds::SelectionChannel, [293](#)
- stationDelete
  - Bds::AdminAccess, [67](#)
- stationGetList
  - Bds::AdminAccess, [67](#)
  - Bds::DataAccess, [134](#)
  - Bds::DataAddAccess, [148](#)
- StationNameLen
  - Bds, [39](#)
- stations
  - Bds::Network, [272](#)
  - Bds::SelectionInfo, [295](#)
- stationUpdate
  - Bds::AdminAccess, [67](#)
- statisticsGet
  - Bds::AdminAccess, [67](#)
  - Bds::DataAccess, [134](#)
  - Bds::DataAddAccess, [148](#)
- status
  - Bds::CdDataChannel, [94](#)
- str
  - Bds::DataError, [168](#)
- streamId
  - Bds::GcfChannel, [254](#)
- streamlet
  - Bds::BdsDataPacketHeader, [78](#)
- StreamsMax
  - Bds::DataFileBds, [189](#)
- stringFormat
  - Bds, [37](#)
- subSystem
  - Bds::Log, [267](#)
  - Bds::LogSelect, [269](#)
- symmetry
  - Bds::Response, [286](#)
- synchronous
  - Bds::DataInfo, [244](#)
- systemId
  - Bds::GcfChannel, [254](#)
- table
  - Bds::Change, [101](#)
- telephone
  - Bds::User, [315](#)
- TEST\_REORDER
  - BdsDataFileGcf.cpp, [329](#)
- time
  - Bds::Change, [102](#)
  - Bds::ChangeGroup, [105](#)
  - Bds::Log, [268](#)
- timeAdded
  - Bds::Note, [278](#)
- timeCompare
  - Bds::DataFile, [179](#)
- TimePeriod
  - Bds::TimePeriod, [309](#)
- TIMESTAMP\_JITTER
  - BdsDataFileBds.cpp, [323](#)
  - BdsDataFileCd.cpp, [327](#)
- title
  - Bds::ChangeGroup, [105](#)
  - Bds::Log, [268](#)



- Bds::Note, [278](#)
- trailerOffset
  - Bds::CdPacketData, [99](#)
- transactionEnd
  - Bds::AdminAccess, [67](#)
- transactionStart
  - Bds::AdminAccess, [68](#)
- type
  - Bds::BdsDataBlockHeader, [72](#)
  - Bds::BdsDataPacketHeader, [78](#)
  - Bds::Change, [102](#)
  - Bds::ChangeGroup, [105](#)
  - Bds::Digitiser, [249](#)
  - Bds::GcfChannel, [254](#)
  - Bds::Log, [268](#)
  - Bds::LogSelect, [270](#)
  - Bds::Note, [278](#)
  - Bds::Response, [286](#)
  - Bds::Sensor, [300](#)
  - Bds::Station, [308](#)
- unCompress
  - Bds::CompressSteim1, [122](#)
- url
  - Bds::DataFileInfo, [214](#)
- User
  - Bds::User, [312](#)
- user
  - Bds::ChangeGroup, [105](#)
  - Bds::Note, [278](#)
  - Bds::User, [315](#)
- userDelete
  - Bds::AdminAccess, [68](#)
- userGet
  - Bds::AdminAccess, [68](#)
  - Bds::DataAccess, [134](#)
  - Bds::DataAddAccess, [148](#)
- userGetFromId
  - Bds::AdminAccess, [68](#)
  - Bds::DataAccess, [134](#)
  - Bds::DataAddAccess, [148](#)
- userGetGroups
  - Bds::AdminAccess, [68](#)
  - Bds::DataAccess, [135](#)
  - Bds::DataAddAccess, [148](#)
- userGetList
  - Bds::AdminAccess, [68](#)
- userSet
  - Bds::AdminAccess, [69](#)
  - Bds::DataAccess, [135](#)
  - Bds::DataAddAccess, [149](#)
- userUpdate
  - Bds::AdminAccess, [69](#)
- validateChecksum
  - Bds::BdsDataPacket, [77](#)
- validateUser
  - Bds::AdminAccess, [69](#)
  - Bds::DataAccess, [135](#)
- Bds::DataAddAccess, [149](#)
- verticalAngle
  - Bds::Calibration, [90](#)
- warnings
  - Bds::DataInfo, [244](#)
- wfid
  - Bds::DataFileCssData, [204](#)
- writeData
  - Bds::DataFile, [179](#)
  - Bds::DataFileAscii, [185](#)
  - Bds::DataFileBds, [193](#)
  - Bds::DataFileBknas, [195](#)
  - Bds::DataFileIms, [210](#)
  - Bds::DataFileLog, [220](#)
  - Bds::DataFileSeed, [229](#)
- WriteOptionNoMetadata
  - Bds::DataFile, [174](#)
- WriteOptionNone
  - Bds::DataFile, [174](#)
- WriteOptionSensorData
  - Bds::DataFile, [174](#)
- WriteOptionsList
  - Bds::DataFile, [174](#)
- x
  - Bds::Point, [279](#)
- y
  - Bds::Point, [279](#)
- zeroed
  - Bds::CdFlag, [96](#)
- zeros
  - Bds::PoleZero, [281](#)