

BdsApi

3.0.12

Generated by Doxygen 1.9.1

1 Main Page	1
1.1 Introduction	1
1.2 Overview	2
1.3 C++ Examples	2
1.4 Python Examples	5
2 Namespace Index	9
2.1 Namespace List	9
3 Hierarchical Index	11
3.1 Class Hierarchy	11
4 Class Index	15
4.1 Class List	15
5 File Index	19
5.1 File List	19
6 Namespace Documentation	21
6.1 Bds Namespace Reference	21
6.1.1 Typedef Documentation	28
6.1.1.1 DataFormats	28
6.1.2 Enumeration Type Documentation	28
6.1.2.1 AvailType	28
6.1.2.2 BdsDataType	29
6.1.2.3 DataFlags	29
6.1.2.4 DataFormatSet	30
6.1.2.5 Errors	30
6.1.2.6 FileHeaderType	30
6.1.2.7 FileSampleType	31
6.1.2.8 LocationSelect	31
6.1.2.9 Mode	31
6.1.2.10 Priority	31
6.1.2.11 SampleFormat	33
6.1.2.12 SelectionGroup	33
6.1.3 Function Documentation	33
6.1.3.1 bdsChannelGetName()	33
6.1.3.2 bdsChannelGetTypeAux()	34
6.1.3.3 bdsDataChannelInfo()	34
6.1.3.4 bdsDataChannelOverallResponse()	34
6.1.3.5 bdsDataChannelRef() [1/2]	34
6.1.3.6 bdsDataChannelRef() [2/2]	35
6.1.3.7 bdsDataFileSeedLogError()	35
6.1.3.8 bdsDataFileSeedLogWarning()	35

6.1.3.9 bdsDataInfoFlatten()	35
6.1.3.10 bdsDataInfoFromInfo()	35
6.1.3.11 bdsDataInfoMergeFlatten()	35
6.1.3.12 bdsDataInfoSetTimeRange()	36
6.1.3.13 bdsDataTypes()	36
6.1.3.14 bdsDumpChannelInfos()	36
6.1.3.15 bdsDumpData()	36
6.1.3.16 bdsDumpDataInfo()	36
6.1.3.17 bdsDumpLocation()	37
6.1.3.18 bdsDumpPoleZeros()	37
6.1.3.19 bdsDumpSelection()	37
6.1.3.20 bdsDumpSelectionInfo()	37
6.1.3.21 bdsFileNameExpand() [1/3]	37
6.1.3.22 bdsFileNameExpand() [2/3]	38
6.1.3.23 bdsFileNameExpand() [3/3]	38
6.1.3.24 bdsInfoFromDataInfo()	38
6.1.3.25 bdsLibInit() [1/3]	38
6.1.3.26 bdsLibInit() [2/3]	38
6.1.3.27 bdsLibInit() [3/3]	39
6.1.3.28 bdsMetadataExportFix()	39
6.1.3.29 bdsMetadataImportFix()	39
6.1.3.30 bdsPoleZeroGain()	39
6.1.3.31 bdsPoleZeroGainPhase()	39
6.1.3.32 bdsPoleZeroToFap()	40
6.1.3.33 bdsSelectionChannelInfo()	40
6.1.3.34 bdsSpecialChannelIgnore()	40
6.1.3.35 bdsSpecialChannels()	40
6.1.3.36 bdsSpecialChannelsSet()	40
6.1.3.37 bdsStationAlias()	41
6.1.3.38 bdsUnCompressCm8()	41
6.1.3.39 bdsUnCompressSteim1()	41
6.1.3.40 bdsUnitCase()	41
6.1.3.41 bdsUnits()	41
6.1.3.42 bdsUnitsConvert()	42
6.1.3.43 crc()	42
6.1.3.44 crc64()	42
6.1.3.45 crcInit()	42
6.1.3.46 dataCalculateDifference()	42
6.1.3.47 dataCalculateUnDifference()	42
6.1.3.48 dataChecksum()	43
6.1.3.49 dataCompressCm6()	43
6.1.3.50 dataConvert() [1/3]	43

6.1.3.51 dataConvert() [2/3]	43
6.1.3.52 dataConvert() [3/3]	43
6.1.3.53 dataDeCompressCm6()	43
6.1.3.54 duplicateDump()	44
6.1.3.55 fileNameTime()	44
6.1.3.56 fixedString()	44
6.1.3.57 fixedWidthValue()	44
6.1.3.58 fromSeedTimeString()	44
6.1.3.59 getHexString()	44
6.1.3.60 nullString()	45
6.1.3.61 record_handler()	45
6.1.3.62 removeCR()	45
6.1.3.63 responseSort()	45
6.1.3.64 roundDigits()	45
6.1.3.65 seedChannelDataType()	45
6.1.3.66 seedChannelInstrumentCode()	46
6.1.3.67 seedTime()	46
6.1.3.68 seedTimeString()	46
6.1.3.69 stringFormat()	46
6.1.3.70 unitsCode()	46
6.1.4 Variable Documentation	46
6.1.4.1 apiVersion	46
6.1.4.2 BdsDataFileVersion	47
6.1.4.3 bdsSpecialChannelsList	47
6.1.4.4 ChannelAuxLen	47
6.1.4.5 ChannelTypeLen	47
6.1.4.6 cm6Table	47
6.1.4.7 cm6TableRev	48
6.1.4.8 crcInitDone	48
6.1.4.9 crcVec	48
6.1.4.10 dataFormatAll	48
6.1.4.11 NetworkNameLen	48
6.1.4.12 node_types	49
6.1.4.13 Scale	49
6.1.4.14 seedIcodeToDataTypes	49
6.1.4.15 SourceLen	49
6.1.4.16 StationNameLen	49
7 Class Documentation	51
7.1 Bds::AccessGroup Class Reference	51
7.1.1 Detailed Description	52
7.1.2 Constructor & Destructor Documentation	52

7.1.2.1 AccessGroup()	52
7.1.3 Member Function Documentation	52
7.1.3.1 getMember()	52
7.1.3.2 getMembers()	52
7.1.3.3 getType()	52
7.1.3.4 setMember()	53
7.1.3.5 setMembers()	53
7.1.4 Member Data Documentation	53
7.1.4.1 endTime	53
7.1.4.2 group	53
7.1.4.3 id	53
7.1.4.4 network	54
7.1.4.5 startTime	54
7.1.4.6 station	54
7.2 Bds::AdminAccess Class Reference	54
7.2.1 Detailed Description	59
7.2.2 Constructor & Destructor Documentation	59
7.2.2.1 AdminAccess()	60
7.2.3 Member Function Documentation	60
7.2.3.1 accessGroupDelete()	60
7.2.3.2 accessGroupGetList()	60
7.2.3.3 accessGroupUpdate()	60
7.2.3.4 calibrationDelete()	60
7.2.3.5 calibrationGetList()	61
7.2.3.6 calibrationUpdate()	61
7.2.3.7 changeDelete()	61
7.2.3.8 changeGetList()	61
7.2.3.9 changeGetListNumber()	61
7.2.3.10 changeGroupDelete()	62
7.2.3.11 changeGroupEnd()	62
7.2.3.12 changeGroupGetList()	62
7.2.3.13 changeGroupStart()	62
7.2.3.14 channelDelete()	62
7.2.3.15 channelGet()	63
7.2.3.16 channelGetList()	63
7.2.3.17 channelInstrumentDelete()	63
7.2.3.18 channelInstrumentGetList()	63
7.2.3.19 channelInstrumentUpdate()	63
7.2.3.20 channelUpdate()	64
7.2.3.21 clean()	64
7.2.3.22 connect()	64
7.2.3.23 dataAvailability()	64

7.2.3.24 databaseBackup()	64
7.2.3.25 databaseRestore()	65
7.2.3.26 dataChannelDelete()	65
7.2.3.27 dataChannelGetList()	65
7.2.3.28 dataChannelUpdate()	65
7.2.3.29 dataClose()	65
7.2.3.30 dataFileDelete()	66
7.2.3.31 dataFileGetList()	66
7.2.3.32 dataFileUpdate()	66
7.2.3.33 dataFormatGetList()	66
7.2.3.34 dataFormattedGetLength()	66
7.2.3.35 dataFormattedRead()	67
7.2.3.36 dataGetBlock()	67
7.2.3.37 dataGetChannelInfo()	67
7.2.3.38 dataGetInfo()	67
7.2.3.39 dataGetNotes()	67
7.2.3.40 dataGetWarnings()	68
7.2.3.41 dataOpen()	68
7.2.3.42 dataPutBlock()	68
7.2.3.43 dataRealtimeConfig()	68
7.2.3.44 dataRealtimeGet()	69
7.2.3.45 dataSearch()	69
7.2.3.46 dataSeekBlock()	69
7.2.3.47 dataSetInfo()	69
7.2.3.48 digitiserDelete()	69
7.2.3.49 digitiserGet()	70
7.2.3.50 digitiserGetList()	70
7.2.3.51 digitiserUpdate()	70
7.2.3.52 eventDelete()	70
7.2.3.53 eventGetList()	70
7.2.3.54 eventUpdate()	71
7.2.3.55 extraCall()	71
7.2.3.56 getSelectionInfo()	71
7.2.3.57 getSelections()	71
7.2.3.58 getVersion()	71
7.2.3.59 groupDelete()	72
7.2.3.60 groupGetList()	72
7.2.3.61 groupUpdate()	72
7.2.3.62 locationDelete()	72
7.2.3.63 locationGetList()	72
7.2.3.64 locationUpdate()	73
7.2.3.65 logAppend()	73

7.2.3.66 logDelete()	73
7.2.3.67 logGetList()	73
7.2.3.68 logUpdate()	73
7.2.3.69 metadataGetChannelInfo()	74
7.2.3.70 metadataGetFormatted()	74
7.2.3.71 modeSet()	74
7.2.3.72 modeSnapshotPause()	74
7.2.3.73 networkDelete()	74
7.2.3.74 networkGetList()	75
7.2.3.75 networkUpdate()	75
7.2.3.76 noteDelete()	75
7.2.3.77 noteGetList()	75
7.2.3.78 noteReadDocument()	75
7.2.3.79 noteUpdate()	76
7.2.3.80 noteWriteDocument()	76
7.2.3.81 responseDelete()	76
7.2.3.82 responseGetList()	76
7.2.3.83 responseUpdate()	76
7.2.3.84 sensorDelete()	77
7.2.3.85 sensorGet()	77
7.2.3.86 sensorGetList()	77
7.2.3.87 sensorUpdate()	77
7.2.3.88 serverConfigurationGet()	77
7.2.3.89 setUser()	78
7.2.3.90 setUserReal()	78
7.2.3.91 sourceDelete()	78
7.2.3.92 sourceGetList()	78
7.2.3.93 sourcePriorityDelete()	78
7.2.3.94 sourcePriorityGetList()	78
7.2.3.95 sourcePriorityUpdate()	79
7.2.3.96 sourceUpdate()	79
7.2.3.97 specialChannelDelete()	79
7.2.3.98 specialChannelGetList()	79
7.2.3.99 specialChannelUpdate()	79
7.2.3.100 sqlQuery()	80
7.2.3.101 stationDelete()	80
7.2.3.102 stationGetList()	80
7.2.3.103 stationUpdate()	80
7.2.3.104 statisticsGet()	80
7.2.3.105 transactionEnd()	81
7.2.3.106 transactionStart()	81
7.2.3.107 userDelete()	81

7.2.3.108 userGet()	81
7.2.3.109 userGetFromId()	81
7.2.3.110 userGetGroups()	81
7.2.3.111 userGetList()	82
7.2.3.112 userGetOptions()	82
7.2.3.113 userSet()	82
7.2.3.114 userSetOptions()	82
7.2.3.115 userUpdate()	82
7.2.3.116 validateUser()	83
7.3 Bds::ArrayChannel Class Reference	83
7.3.1 Detailed Description	83
7.3.2 Constructor & Destructor Documentation	83
7.3.2.1 ArrayChannel()	84
7.3.3 Member Data Documentation	84
7.3.3.1 arrayOffsetEast	84
7.3.3.2 arrayOffsetNorth	84
7.3.3.3 channel	84
7.3.3.4 network	84
7.3.3.5 station	85
7.4 Bds::BdsDataBlock Struct Reference	85
7.4.1 Detailed Description	85
7.4.2 Member Data Documentation	85
7.4.2.1 data	85
7.4.2.2 header	85
7.5 Bds::BdsDataBlockHeader Struct Reference	86
7.5.1 Detailed Description	86
7.5.2 Member Data Documentation	86
7.5.2.1 length	86
7.5.2.2 packetOffset	86
7.5.2.3 type	86
7.6 Bds::BdsDataBlockPos Class Reference	87
7.6.1 Detailed Description	87
7.6.2 Constructor & Destructor Documentation	87
7.6.2.1 BdsDataBlockPos()	87
7.6.3 Member Function Documentation	87
7.6.3.1 operator<()	88
7.6.4 Member Data Documentation	88
7.6.4.1 channel	88
7.6.4.2 endTime	88
7.6.4.3 numChannels	88
7.6.4.4 numSamples	88
7.6.4.5 position	88

7.6.4.6 segment	88
7.6.4.7 startTime	89
7.7 Bds::BdsDataPacket Class Reference	89
7.7.1 Detailed Description	89
7.7.2 Constructor & Destructor Documentation	89
7.7.2.1 BdsDataPacket()	90
7.7.2.2 ~BdsDataPacket()	90
7.7.3 Member Function Documentation	90
7.7.3.1 clear()	90
7.7.3.2 dump()	90
7.7.3.3 getHeader()	90
7.7.3.4 reset()	90
7.7.3.5 setChecksumAndLength()	90
7.7.3.6 setHeader()	91
7.7.3.7 validateChecksum()	91
7.8 Bds::BdsDataPacketHeader Struct Reference	91
7.8.1 Detailed Description	91
7.8.2 Member Data Documentation	91
7.8.2.1 checksum	92
7.8.2.2 endTime	92
7.8.2.3 length	92
7.8.2.4 sequence	92
7.8.2.5 startTime	92
7.8.2.6 streamlet	92
7.8.2.7 type	93
7.9 Bds::BdsDataSegment Class Reference	93
7.9.1 Detailed Description	93
7.9.2 Constructor & Destructor Documentation	93
7.9.2.1 BdsDataSegment()	93
7.9.3 Member Function Documentation	93
7.9.3.1 operator<()	94
7.9.4 Member Data Documentation	94
7.9.4.1 blocks	94
7.9.4.2 endTime	94
7.9.4.3 numBlocks	94
7.9.4.4 numSamples	94
7.9.4.5 sampleRate	94
7.9.4.6 startTime	94
7.10 Bds::BdsDataStreamlet Class Reference	95
7.10.1 Detailed Description	95
7.10.2 Constructor & Destructor Documentation	95
7.10.2.1 BdsDataStreamlet()	95

7.10.3 Member Data Documentation	95
7.10.3.1 blocks	95
7.10.3.2 channel	96
7.10.3.3 numChannels	96
7.10.3.4 packetNumber	96
7.10.3.5 position	96
7.10.3.6 segments	96
7.11 Bds::BdsSeedType Class Reference	96
7.11.1 Detailed Description	97
7.11.2 Constructor & Destructor Documentation	97
7.11.2.1 BdsSeedType()	97
7.11.3 Member Function Documentation	97
7.11.3.1 appendDouble()	97
7.11.3.2 appendExp()	97
7.11.3.3 appendInt()	98
7.11.3.4 appendString()	98
7.11.3.5 appendStringVariable()	98
7.11.3.6 getDouble()	98
7.11.3.7 getInt()	98
7.11.3.8 getString()	98
7.11.3.9 getStringVariable()	99
7.11.3.10 getUInt()	99
7.12 Bds::Calibration Class Reference	99
7.12.1 Detailed Description	100
7.12.2 Constructor & Destructor Documentation	101
7.12.2.1 Calibration()	101
7.12.3 Member Function Documentation	101
7.12.3.1 getMember()	101
7.12.3.2 getMembers()	101
7.12.3.3 getType()	102
7.12.3.4 setMember()	102
7.12.3.5 setMembers()	102
7.12.4 Member Data Documentation	102
7.12.4.1 calibrationFactor	102
7.12.4.2 calibrationFrequency	102
7.12.4.3 calibrationUnits	103
7.12.4.4 calibrationUnitsDesc	103
7.12.4.5 channel	103
7.12.4.6 depth	103
7.12.4.7 endTime	103
7.12.4.8 horizontalAngle	103
7.12.4.9 id	104

7.12.4.10 name	104
7.12.4.11 network	104
7.12.4.12 rawCalibrationFactor	104
7.12.4.13 rawCalibrationFrequency	104
7.12.4.14 rawCalibrationUnits	104
7.12.4.15 samplingFrequency	105
7.12.4.16 source	105
7.12.4.17 startTime	105
7.12.4.18 station	105
7.12.4.19 verticalAngle	105
7.12.4.20 waterLevel	105
7.13 Bds::CdChannel_1v0 Struct Reference	106
7.13.1 Detailed Description	106
7.13.2 Member Data Documentation	106
7.13.2.1 auth	106
7.13.2.2 calibrationFactor	106
7.13.2.3 calibrationPeriod	106
7.13.2.4 channel	107
7.13.2.5 channelName	107
7.13.2.6 compress	107
7.13.2.7 name	107
7.13.2.8 spare0	107
7.13.2.9 spare1	107
7.13.2.10 stationName	107
7.14 Bds::CdDataChannel Class Reference	108
7.14.1 Detailed Description	108
7.14.2 Member Data Documentation	108
7.14.2.1 channel	108
7.14.2.2 data	108
7.14.2.3 dataSize	108
7.14.2.4 mode	109
7.14.2.5 numSamples	109
7.14.2.6 period	109
7.14.2.7 startTime	109
7.14.2.8 station	109
7.14.2.9 status	109
7.15 Bds::CdDataFormatFrame_1v0 Struct Reference	109
7.15.1 Detailed Description	110
7.15.2 Member Data Documentation	110
7.15.2.1 channels	110
7.15.2.2 frameLength	110
7.15.2.3 frameType	110

7.15.2.4 maxFrameLength	110
7.15.2.5 numChannels	110
7.15.2.6 period	111
7.16 Bds::CdFlag Class Reference	111
7.16.1 Detailed Description	111
7.16.2 Constructor & Destructor Documentation	111
7.16.2.1 CdFlag()	111
7.16.3 Member Data Documentation	111
7.16.3.1 dead	111
7.16.3.2 zeroed	112
7.17 Bds::CdPacketData Class Reference	112
7.17.1 Detailed Description	112
7.17.2 Member Data Documentation	112
7.17.2.1 auth	112
7.17.2.2 authKey	113
7.17.2.3 authSize	113
7.17.2.4 channels	113
7.17.2.5 crc	113
7.17.2.6 creator	113
7.17.2.7 destination	113
7.17.2.8 frameType	113
7.17.2.9 numChannels	113
7.17.2.10 period	114
7.17.2.11 sequenceNum	114
7.17.2.12 series	114
7.17.2.13 startTime	114
7.17.2.14 trailerOffset	114
7.18 Bds::Change Class Reference	114
7.18.1 Detailed Description	115
7.18.2 Constructor & Destructor Documentation	115
7.18.2.1 Change()	115
7.18.3 Member Function Documentation	116
7.18.3.1 getMember()	116
7.18.3.2 getMembers()	116
7.18.3.3 getType()	116
7.18.3.4 setMember()	116
7.18.3.5 setMembers()	116
7.18.4 Member Data Documentation	117
7.18.4.1 changeGroupId	117
7.18.4.2 id	117
7.18.4.3 rowId	117
7.18.4.4 table	117

7.18.4.5 time	117
7.18.4.6 type	118
7.19 Bds::ChangeGroup Class Reference	118
7.19.1 Detailed Description	119
7.19.2 Constructor & Destructor Documentation	119
7.19.2.1 ChangeGroup()	119
7.19.3 Member Function Documentation	119
7.19.3.1 getMember()	119
7.19.3.2 getMembers()	119
7.19.3.3 getType()	119
7.19.3.4 setMember()	120
7.19.3.5 setMembers()	120
7.19.4 Member Data Documentation	120
7.19.4.1 description	120
7.19.4.2 id	120
7.19.4.3 time	120
7.19.4.4 title	121
7.19.4.5 type	121
7.19.4.6 user	121
7.20 Bds::Channel Class Reference	121
7.20.1 Detailed Description	122
7.20.2 Constructor & Destructor Documentation	122
7.20.2.1 Channel()	122
7.20.3 Member Function Documentation	123
7.20.3.1 getMember()	123
7.20.3.2 getMembers()	123
7.20.3.3 getType()	123
7.20.3.4 setMember()	123
7.20.3.5 setMembers()	123
7.20.4 Member Data Documentation	124
7.20.4.1 channel	124
7.20.4.2 channelAux	124
7.20.4.3 channelType	124
7.20.4.4 dataType	124
7.20.4.5 description	124
7.20.4.6 endTime	125
7.20.4.7 id	125
7.20.4.8 network	125
7.20.4.9 startTime	125
7.20.4.10 station	125
7.21 Bds::ChannelInfo Class Reference	125
7.21.1 Detailed Description	126

7.21.2 Constructor & Destructor Documentation	126
7.21.2.1 ChannelInfo()	127
7.21.3 Member Data Documentation	127
7.21.3.1 calibration	127
7.21.3.2 channel	127
7.21.3.3 channelLocation	127
7.21.3.4 digitiser	127
7.21.3.5 endTime	128
7.21.3.6 responses	128
7.21.3.7 sensor	128
7.21.3.8 source	128
7.21.3.9 startTime	128
7.21.3.10 station	128
7.21.3.11 stationLocation	129
7.22 Bds::ChannelInfos Class Reference	129
7.22.1 Detailed Description	129
7.22.2 Constructor & Destructor Documentation	129
7.22.2.1 ChannelInfos()	129
7.22.3 Member Data Documentation	130
7.22.3.1 array	130
7.22.3.2 channels	130
7.23 Bds::ChannelInstrument Class Reference	130
7.23.1 Detailed Description	131
7.23.2 Constructor & Destructor Documentation	131
7.23.2.1 ChannelInstrument()	131
7.23.3 Member Function Documentation	131
7.23.3.1 getMember()	132
7.23.3.2 getMembers()	132
7.23.3.3 getType()	132
7.23.3.4 setMember()	132
7.23.3.5 setMembers()	132
7.23.4 Member Data Documentation	132
7.23.4.1 channelId	133
7.23.4.2 digitiserId	133
7.23.4.3 endTime	133
7.23.4.4 id	133
7.23.4.5 sensorId	133
7.23.4.6 source	133
7.23.4.7 startTime	134
7.24 Bds::ChannelName Class Reference	134
7.24.1 Detailed Description	134
7.24.2 Constructor & Destructor Documentation	134

7.24.2.1 ChannelName()	135
7.24.3 Member Data Documentation	135
7.24.3.1 channel	135
7.24.3.2 network	135
7.24.3.3 source	135
7.24.3.4 station	135
7.25 Bds::CleanOptions Class Reference	136
7.25.1 Detailed Description	136
7.25.2 Constructor & Destructor Documentation	136
7.25.2.1 CleanOptions()	136
7.25.3 Member Data Documentation	136
7.25.3.1 changes	136
7.25.3.2 deletedFiles	137
7.25.3.3 logs	137
7.26 Bds::CompressSteim1 Class Reference	137
7.26.1 Detailed Description	137
7.26.2 Constructor & Destructor Documentation	137
7.26.2.1 CompressSteim1()	137
7.26.3 Member Function Documentation	138
7.26.3.1 clear()	138
7.26.3.2 setByteOrder()	138
7.26.3.3 unCompress()	138
7.27 Bds::DataAccess Class Reference	138
7.27.1 Detailed Description	141
7.27.2 Constructor & Destructor Documentation	141
7.27.2.1 DataAccess()	141
7.27.3 Member Function Documentation	142
7.27.3.1 calibrationGetList()	142
7.27.3.2 channelGetList()	142
7.27.3.3 channelInstrumentGetList()	142
7.27.3.4 clean()	142
7.27.3.5 connect()	142
7.27.3.6 dataAvailability()	143
7.27.3.7 databaseBackup()	143
7.27.3.8 dataChannelGetList()	143
7.27.3.9 dataClose()	143
7.27.3.10 dataFileGetList()	143
7.27.3.11 dataFormatGetList()	144
7.27.3.12 dataFormattedGetLength()	144
7.27.3.13 dataFormattedRead()	144
7.27.3.14 dataGetBlock()	144
7.27.3.15 dataGetChannelInfo()	144

7.27.3.16 dataGetInfo()	145
7.27.3.17 dataGetNotes()	145
7.27.3.18 dataGetWarnings()	145
7.27.3.19 dataOpen()	145
7.27.3.20 dataRealtimeConfig()	146
7.27.3.21 dataRealtimeGet()	146
7.27.3.22 dataSearch()	146
7.27.3.23 dataSeekBlock()	146
7.27.3.24 digitiserGet()	147
7.27.3.25 digitiserGetList()	147
7.27.3.26 eventGetList()	147
7.27.3.27 getSelectionInfo()	147
7.27.3.28 getSelections()	147
7.27.3.29 getVersion()	148
7.27.3.30 groupGetList()	148
7.27.3.31 locationGetList()	148
7.27.3.32 logAppend()	148
7.27.3.33 logUpdate()	148
7.27.3.34 metadataGetChannelInfo()	149
7.27.3.35 metadataGetFormatted()	149
7.27.3.36 modeSet()	149
7.27.3.37 modeSnapshotPause()	149
7.27.3.38 networkGetList()	149
7.27.3.39 noteGetList()	150
7.27.3.40 noteReadDocument()	150
7.27.3.41 noteUpdate()	150
7.27.3.42 noteWriteDocument()	150
7.27.3.43 responseGetList()	150
7.27.3.44 sensorGet()	151
7.27.3.45 sensorGetList()	151
7.27.3.46 serverConfigurationGet()	151
7.27.3.47 setUser()	151
7.27.3.48 setUserReal()	151
7.27.3.49 sourceGetList()	152
7.27.3.50 sourcePriorityGetList()	152
7.27.3.51 specialChannelGetList()	152
7.27.3.52 stationGetList()	152
7.27.3.53 statisticsGet()	152
7.27.3.54 userGet()	153
7.27.3.55 userGetFromId()	153
7.27.3.56 userGetGroups()	153
7.27.3.57 userGetOptions()	153

7.27.3.58 userSet()	153
7.27.3.59 userSetOptions()	154
7.27.3.60 validateUser()	154
7.28 Bds::DataAddAccess Class Reference	154
7.28.1 Detailed Description	157
7.28.2 Constructor & Destructor Documentation	157
7.28.2.1 DataAddAccess()	158
7.28.3 Member Function Documentation	158
7.28.3.1 calibrationGetList()	158
7.28.3.2 channelGetList()	158
7.28.3.3 channelInstrumentGetList()	158
7.28.3.4 clean()	158
7.28.3.5 connect()	159
7.28.3.6 dataAvailability()	159
7.28.3.7 databaseBackup()	159
7.28.3.8 dataChannelGetList()	159
7.28.3.9 dataClose()	159
7.28.3.10 dataFileGetList()	160
7.28.3.11 dataFormatGetList()	160
7.28.3.12 dataFormattedGetLength()	160
7.28.3.13 dataFormattedRead()	160
7.28.3.14 dataGetBlock()	160
7.28.3.15 dataGetChannelInfo()	161
7.28.3.16 dataGetInfo()	161
7.28.3.17 dataGetNotes()	161
7.28.3.18 dataGetWarnings()	161
7.28.3.19 dataOpen()	161
7.28.3.20 dataPutBlock()	162
7.28.3.21 dataRealtimeConfig()	162
7.28.3.22 dataRealtimeGet()	162
7.28.3.23 dataSearch()	162
7.28.3.24 dataSeekBlock()	163
7.28.3.25 dataSetInfo()	163
7.28.3.26 digitiserGet()	163
7.28.3.27 digitiserGetList()	163
7.28.3.28 eventDelete()	163
7.28.3.29 eventGetList()	164
7.28.3.30 eventUpdate()	164
7.28.3.31 getSelectionInfo()	164
7.28.3.32 getSelections()	164
7.28.3.33 getVersion()	164
7.28.3.34 groupGetList()	165

7.28.3.35 locationGetList()	165
7.28.3.36 logAppend()	165
7.28.3.37 logUpdate()	165
7.28.3.38 metadataGetChannelInfo()	165
7.28.3.39 metadataGetFormatted()	166
7.28.3.40 modeSet()	166
7.28.3.41 modeSnapshotPause()	166
7.28.3.42 networkGetList()	166
7.28.3.43 noteGetList()	166
7.28.3.44 noteReadDocument()	167
7.28.3.45 noteUpdate()	167
7.28.3.46 noteWriteDocument()	167
7.28.3.47 responseGetList()	167
7.28.3.48 sensorGet()	167
7.28.3.49 sensorGetList()	168
7.28.3.50 serverConfigurationGet()	168
7.28.3.51 setUser()	168
7.28.3.52 setUserReal()	168
7.28.3.53 sourceGetList()	168
7.28.3.54 sourcePriorityGetList()	169
7.28.3.55 specialChannelGetList()	169
7.28.3.56 stationGetList()	169
7.28.3.57 statisticsGet()	169
7.28.3.58 userGet()	169
7.28.3.59 userGetFromId()	170
7.28.3.60 userGetGroups()	170
7.28.3.61 userGetOptions()	170
7.28.3.62 userSet()	170
7.28.3.63 userSetOptions()	170
7.28.3.64 validateUser()	171
7.29 Bds::DataAvail Class Reference	171
7.29.1 Detailed Description	171
7.29.2 Constructor & Destructor Documentation	171
7.29.2.1 DataAvail()	171
7.29.3 Member Data Documentation	172
7.29.3.1 availType	172
7.29.3.2 endTime	172
7.29.3.3 startTime	172
7.30 Bds::DataAvailChan Class Reference	172
7.30.1 Detailed Description	173
7.30.2 Constructor & Destructor Documentation	173
7.30.2.1 DataAvailChan()	173

7.30.3 Member Data Documentation	173
7.30.3.1 channel	174
7.30.3.2 endTime	174
7.30.3.3 network	174
7.30.3.4 segments	174
7.30.3.5 source	174
7.30.3.6 startTime	174
7.30.3.7 station	175
7.31 Bds::DataBlock Class Reference	175
7.31.1 Detailed Description	175
7.31.2 Constructor & Destructor Documentation	176
7.31.2.1 DataBlock()	176
7.31.3 Member Data Documentation	176
7.31.3.1 channelData	176
7.31.3.2 channelNumber	176
7.31.3.3 endTime	176
7.31.3.4 info	176
7.31.3.5 segmentNumber	177
7.31.3.6 startTime	177
7.32 Bds::DataBlockChannel Class Reference	177
7.32.1 Detailed Description	177
7.32.2 Constructor & Destructor Documentation	178
7.32.2.1 DataBlockChannel()	178
7.32.3 Member Data Documentation	178
7.32.3.1 channel	178
7.32.3.2 network	178
7.32.3.3 source	178
7.32.3.4 station	178
7.33 Bds::DataBlockPos Class Reference	179
7.33.1 Detailed Description	179
7.33.2 Constructor & Destructor Documentation	179
7.33.2.1 DataBlockPos()	179
7.33.3 Member Function Documentation	179
7.33.3.1 operator<()	180
7.33.4 Member Data Documentation	180
7.33.4.1 endTime	180
7.33.4.2 numSamples	180
7.33.4.3 order	180
7.33.4.4 position	180
7.33.4.5 ref	180
7.33.4.6 startTime	180
7.34 Bds::DataChannel Class Reference	181

7.34.1 Detailed Description	182
7.34.2 Constructor & Destructor Documentation	182
7.34.2.1 DataChannel()	182
7.34.3 Member Function Documentation	183
7.34.3.1 getMember()	183
7.34.3.2 getMembers()	183
7.34.3.3 getType()	183
7.34.3.4 setMember()	183
7.34.3.5 setMembers()	183
7.34.4 Member Data Documentation	184
7.34.4.1 channel	184
7.34.4.2 dataFileChannel	184
7.34.4.3 dataFileId	184
7.34.4.4 endTime	184
7.34.4.5 id	184
7.34.4.6 importFilename	185
7.34.4.7 importFormat	185
7.34.4.8 importStartTime	185
7.34.4.9 info	185
7.34.4.10 network	185
7.34.4.11 numBlocks	185
7.34.4.12 numSamples	186
7.34.4.13 sampleFormat	186
7.34.4.14 sampleRate	186
7.34.4.15 source	186
7.34.4.16 startTime	186
7.34.4.17 station	186
7.35 Bds::DataCollate Class Reference	187
7.35.1 Detailed Description	187
7.35.2 Constructor & Destructor Documentation	187
7.35.2.1 DataCollate()	187
7.35.2.2 ~DataCollate()	187
7.35.3 Member Function Documentation	187
7.35.3.1 addSource()	187
7.35.3.2 readData()	188
7.36 Bds::DataError Class Reference	188
7.36.1 Detailed Description	189
7.36.2 Constructor & Destructor Documentation	189
7.36.2.1 DataError() [1/2]	189
7.36.2.2 DataError() [2/2]	189
7.36.3 Member Function Documentation	190
7.36.3.1 getErrorNumber()	190

7.36.3.2 getString()	190
7.36.3.3 getTitle()	190
7.36.3.4 mergeDataInfo()	190
7.36.3.5 num()	190
7.36.3.6 operator int()	191
7.36.3.7 set()	191
7.36.3.8 setString()	191
7.36.3.9 setStringUser()	191
7.36.3.10 str()	191
7.36.4 Member Data Documentation	192
7.36.4.1 ochannel	192
7.36.4.2 odescription	192
7.36.4.3 oendTime	192
7.36.4.4 oerrorNumber	192
7.36.4.5 ofilename	192
7.36.4.6 onetwork	193
7.36.4.7 osource	193
7.36.4.8 ostartTime	193
7.36.4.9 ostation	193
7.36.4.10 otitle	193
7.36.4.11 ouser	193
7.37 Bds::DataFile Class Reference	194
7.37.1 Detailed Description	196
7.37.2 Member Enumeration Documentation	196
7.37.2.1 DataOrder	196
7.37.2.2 Features	196
7.37.2.3 ReadOptionsList	197
7.37.2.4 WriteOptionsList	197
7.37.3 Constructor & Destructor Documentation	197
7.37.3.1 DataFile()	197
7.37.3.2 ~DataFile()	197
7.37.4 Member Function Documentation	198
7.37.4.1 close()	198
7.37.4.2 dataErrorFixup()	198
7.37.4.3 duplicateCheck()	198
7.37.4.4 end()	198
7.37.4.5 fileNameProcess()	198
7.37.4.6 flush()	199
7.37.4.7 getDataOrder()	199
7.37.4.8 getFeatures()	199
7.37.4.9 getFileName()	199
7.37.4.10 getFilePosition()	199

7.37.4.11	getFixesInfo()	200
7.37.4.12	getFormat()	200
7.37.4.13	getFormats()	200
7.37.4.14	getInfo()	200
7.37.4.15	getMetaData()	200
7.37.4.16	init()	201
7.37.4.17	open()	201
7.37.4.18	readData()	201
7.37.4.19	seekBlock()	201
7.37.4.20	setFormat()	202
7.37.4.21	setInfo()	202
7.37.4.22	start()	202
7.37.4.23	timeCompare()	202
7.37.4.24	writeData()	203
7.37.5	Member Data Documentation	203
7.37.5.1	ofile	203
7.37.5.2	ofilename	203
7.37.5.3	ofilenameTime	203
7.37.5.4	offormat	203
7.37.5.5	omode	203
7.38	Bds::DataFileAd22 Class Reference	204
7.38.1	Detailed Description	204
7.38.2	Constructor & Destructor Documentation	204
7.38.2.1	DataFileAd22()	204
7.38.3	Member Function Documentation	205
7.38.3.1	getDataOrder()	205
7.38.3.2	getFeatures()	205
7.38.3.3	getFixesInfo()	205
7.38.3.4	getFormats()	205
7.38.3.5	getInfo()	205
7.38.3.6	readData()	206
7.39	Bds::DataFileAscii Class Reference	206
7.39.1	Detailed Description	207
7.39.2	Constructor & Destructor Documentation	207
7.39.2.1	DataFileAscii()	207
7.39.3	Member Function Documentation	207
7.39.3.1	end()	207
7.39.3.2	getDataOrder()	207
7.39.3.3	getFeatures()	208
7.39.3.4	getFormats()	208
7.39.3.5	open()	208
7.39.3.6	setFormat()	208

7.39.3.7 setInfo()	208
7.39.3.8 start()	209
7.39.3.9 writeData()	209
7.40 Bds::DataFileBdrs Class Reference	209
7.40.1 Detailed Description	210
7.40.2 Constructor & Destructor Documentation	210
7.40.2.1 DataFileBdrs()	210
7.40.3 Member Function Documentation	210
7.40.3.1 getDataOrder()	210
7.40.3.2 getFeatures()	210
7.40.3.3 getFixesInfo()	211
7.40.3.4 getFormats()	211
7.40.3.5 getInfo()	211
7.40.3.6 readData()	211
7.41 Bds::DataFileBds Class Reference	212
7.41.1 Detailed Description	213
7.41.2 Member Enumeration Documentation	213
7.41.2.1 anonymous enum	213
7.41.2.2 anonymous enum	213
7.41.2.3 PackFormat	214
7.41.3 Constructor & Destructor Documentation	214
7.41.3.1 DataFileBds()	214
7.41.3.2 ~DataFileBds()	214
7.41.4 Member Function Documentation	214
7.41.4.1 close()	214
7.41.4.2 flush()	215
7.41.4.3 getDataOrder()	215
7.41.4.4 getDiskBlockSize()	215
7.41.4.5 getFormats()	215
7.41.4.6 getInfo()	215
7.41.4.7 open()	216
7.41.4.8 packetRead()	216
7.41.4.9 packetWrite()	216
7.41.4.10 readData()	216
7.41.4.11 seekBlock()	217
7.41.4.12 setDiskBlockSize()	217
7.41.4.13 setFormat()	217
7.41.4.14 setInfo()	217
7.41.4.15 setReadPositionToStart()	218
7.41.4.16 setWritePositionForAppend()	218
7.41.4.17 streamletToChannel()	218
7.41.4.18 writeData()	218

7.42 Bds::DataFileBknas Class Reference	218
7.42.1 Detailed Description	219
7.42.2 Constructor & Destructor Documentation	219
7.42.2.1 DataFileBknas()	219
7.42.3 Member Function Documentation	219
7.42.3.1 getFormats()	219
7.42.3.2 open()	220
7.42.3.3 setInfo()	220
7.42.3.4 writeData()	220
7.43 Bds::DataFileCd Class Reference	220
7.43.1 Detailed Description	221
7.43.2 Constructor & Destructor Documentation	221
7.43.2.1 DataFileCd()	221
7.43.3 Member Function Documentation	221
7.43.3.1 getDataOrder()	221
7.43.3.2 getFeatures()	222
7.43.3.3 getFixesInfo()	222
7.43.3.4 getFormats()	222
7.43.3.5 getInfo()	222
7.43.3.6 readData()	222
7.44 Bds::DataFileCss Class Reference	223
7.44.1 Detailed Description	223
7.44.2 Constructor & Destructor Documentation	223
7.44.2.1 DataFileCss()	223
7.44.3 Member Function Documentation	224
7.44.3.1 getDataOrder()	224
7.44.3.2 getFeatures()	224
7.44.3.3 getFormats()	224
7.44.3.4 getInfo()	224
7.44.3.5 readData()	225
7.45 Bds::DataFileCssData Class Reference	225
7.45.1 Detailed Description	226
7.45.2 Constructor & Destructor Documentation	226
7.45.2.1 DataFileCssData()	226
7.45.2.2 ~DataFileCssData()	226
7.45.3 Member Function Documentation	226
7.45.3.1 set()	226
7.45.4 Member Data Documentation	226
7.45.4.1 calibrationFactor	227
7.45.4.2 calibrationFreq	227
7.45.4.3 chan	227
7.45.4.4 chanid	227

7.45.4.5 clip	227
7.45.4.6 commId	227
7.45.4.7 datatype	227
7.45.4.8 dirName	227
7.45.4.9 endTime	228
7.45.4.10 file	228
7.45.4.11 fileName	228
7.45.4.12 fileOffset	228
7.45.4.13 instType	228
7.45.4.14 jdate	228
7.45.4.15 loadDate	228
7.45.4.16 nsamp	228
7.45.4.17 sampleBigEndian	229
7.45.4.18 sampleFormat	229
7.45.4.19 sampleRate	229
7.45.4.20 sampleSize	229
7.45.4.21 segtype	229
7.45.4.22 sta	229
7.45.4.23 startTime	229
7.45.4.24 wfid	230
7.46 Bds::DataFileGcf Class Reference	230
7.46.1 Detailed Description	230
7.46.2 Constructor & Destructor Documentation	231
7.46.2.1 DataFileGcf()	231
7.46.3 Member Function Documentation	231
7.46.3.1 getDataOrder()	231
7.46.3.2 getFeatures()	231
7.46.3.3 getFixesInfo()	231
7.46.3.4 getFormats()	231
7.46.3.5 getInfo()	232
7.46.3.6 readData()	232
7.47 Bds::DataFileIdc Class Reference	232
7.47.1 Detailed Description	233
7.47.2 Constructor & Destructor Documentation	233
7.47.2.1 DataFileIdc()	233
7.47.3 Member Function Documentation	233
7.47.3.1 getFeatures()	233
7.47.3.2 getFormats()	233
7.47.3.3 getMetaData()	233
7.47.3.4 setInfo()	234
7.48 Bds::DataFileIms Class Reference	234
7.48.1 Detailed Description	235

7.48.2 Constructor & Destructor Documentation	235
7.48.2.1 DataFileImgs()	235
7.48.3 Member Function Documentation	235
7.48.3.1 close()	235
7.48.3.2 end()	235
7.48.3.3 getDataOrder()	236
7.48.3.4 getFeatures()	236
7.48.3.5 getFormats()	236
7.48.3.6 getMetaData()	236
7.48.3.7 open()	236
7.48.3.8 setInfo()	237
7.48.3.9 start()	237
7.48.3.10 writeData()	237
7.49 Bds::DataFileInfo Class Reference	237
7.49.1 Detailed Description	238
7.49.2 Constructor & Destructor Documentation	238
7.49.2.1 DataFileInfo()	239
7.49.3 Member Function Documentation	239
7.49.3.1 getMember()	239
7.49.3.2 getMembers()	239
7.49.3.3 getType()	239
7.49.3.4 setMember()	239
7.49.3.5 setMembers()	240
7.49.4 Member Data Documentation	240
7.49.4.1 comment	240
7.49.4.2 endTime	240
7.49.4.3 format	240
7.49.4.4 id	240
7.49.4.5 importTime	240
7.49.4.6 importUserId	241
7.49.4.7 location	241
7.49.4.8 startTime	241
7.49.4.9 state	241
7.49.4.10 stream	241
7.49.4.11 url	241
7.50 Bds::DataFileLac Class Reference	242
7.50.1 Detailed Description	242
7.50.2 Constructor & Destructor Documentation	242
7.50.2.1 DataFileLac()	242
7.50.3 Member Function Documentation	243
7.50.3.1 getDataOrder()	243
7.50.3.2 getFeatures()	243

7.50.3.3	getFixesInfo()	243
7.50.3.4	getFormats()	243
7.50.3.5	getInfo()	243
7.50.3.6	readData()	244
7.51	Bds::DataFileLog Class Reference	244
7.51.1	Detailed Description	245
7.51.2	Constructor & Destructor Documentation	245
7.51.2.1	DataFileLog()	245
7.51.3	Member Function Documentation	245
7.51.3.1	end()	245
7.51.3.2	getDataOrder()	245
7.51.3.3	getFeatures()	246
7.51.3.4	getFormats()	246
7.51.3.5	getInfo()	246
7.51.3.6	open()	246
7.51.3.7	readData()	246
7.51.3.8	setFormat()	247
7.51.3.9	setInfo()	247
7.51.3.10	start()	247
7.51.3.11	writeData()	247
7.52	Bds::DataFileOptions Class Reference	248
7.52.1	Detailed Description	248
7.52.2	Constructor & Destructor Documentation	248
7.52.2.1	DataFileOptions()	248
7.52.3	Member Function Documentation	248
7.52.3.1	operator int()	248
7.52.3.2	operator" =()	248
7.52.4	Member Data Documentation	249
7.52.4.1	oignoreBlockList	249
7.52.4.2	ooptionList	249
7.53	Bds::DataFileResponse Class Reference	249
7.53.1	Detailed Description	250
7.53.2	Constructor & Destructor Documentation	250
7.53.2.1	DataFileResponse()	250
7.53.3	Member Function Documentation	250
7.53.3.1	getFeatures()	250
7.53.3.2	getFormats()	250
7.53.3.3	getMetaData()	250
7.53.3.4	setInfo()	251
7.54	Bds::DataFileSac Class Reference	251
7.54.1	Detailed Description	251
7.54.2	Constructor & Destructor Documentation	252

7.54.2.1 DataFileSac()	252
7.54.3 Member Function Documentation	252
7.54.3.1 getFeatures()	252
7.54.3.2 getFormats()	252
7.54.3.3 getMetaData()	252
7.54.3.4 setInfo()	253
7.55 Bds::DataFileSeed Class Reference	253
7.55.1 Detailed Description	254
7.55.2 Constructor & Destructor Documentation	254
7.55.2.1 DataFileSeed()	254
7.55.2.2 ~DataFileSeed()	254
7.55.3 Member Function Documentation	254
7.55.3.1 close()	255
7.55.3.2 end()	255
7.55.3.3 getDataOrder()	255
7.55.3.4 getFeatures()	255
7.55.3.5 getFixesInfo()	255
7.55.3.6 getFormats()	256
7.55.3.7 getInfo()	256
7.55.3.8 getMetaData()	256
7.55.3.9 msrFileWrite()	256
7.55.3.10 readData()	256
7.55.3.11 setFormat()	257
7.55.3.12 setInfo()	257
7.55.3.13 start()	257
7.55.3.14 writeData()	257
7.55.4 Member Data Documentation	257
7.55.4.1 omsrErr	258
7.55.4.2 onoLock	258
7.56 Bds::DataFileStationXml Class Reference	258
7.56.1 Detailed Description	259
7.56.2 Constructor & Destructor Documentation	259
7.56.2.1 DataFileStationXml()	259
7.56.3 Member Function Documentation	259
7.56.3.1 getFeatures()	259
7.56.3.2 getFormats()	259
7.56.3.3 getMetaData()	259
7.56.3.4 setInfo()	260
7.57 Bds::DataFileTapeDigitiser Class Reference	260
7.57.1 Detailed Description	261
7.57.2 Constructor & Destructor Documentation	261
7.57.2.1 DataFileTapeDigitiser()	261

7.57.3 Member Function Documentation	261
7.57.3.1 getFormats()	261
7.57.3.2 getInfo()	261
7.57.3.3 open()	261
7.57.3.4 readData()	262
7.58 Bds::DataFileWra Class Reference	262
7.58.1 Detailed Description	263
7.58.2 Constructor & Destructor Documentation	263
7.58.2.1 DataFileWra()	263
7.58.3 Member Function Documentation	263
7.58.3.1 getDataOrder()	263
7.58.3.2 getFeatures()	263
7.58.3.3 getFixesInfo()	263
7.58.3.4 getFormats()	264
7.58.3.5 getInfo()	264
7.58.3.6 readData()	264
7.58.3.7 setFormat()	264
7.59 Bds::DataFileWraAgso Class Reference	265
7.59.1 Detailed Description	265
7.59.2 Constructor & Destructor Documentation	265
7.59.2.1 DataFileWraAgso()	265
7.59.3 Member Function Documentation	266
7.59.3.1 getDataOrder()	266
7.59.3.2 getFeatures()	266
7.59.3.3 getFormats()	266
7.59.3.4 getInfo()	266
7.59.3.5 readData()	267
7.60 Bds::DataFormat Class Reference	267
7.60.1 Detailed Description	268
7.60.2 Constructor & Destructor Documentation	268
7.60.2.1 DataFormat()	268
7.60.3 Member Data Documentation	268
7.60.3.1 dataRead	268
7.60.3.2 dataWrite	268
7.60.3.3 description	268
7.60.3.4 extension	269
7.60.3.5 metadataRead	269
7.60.3.6 metadataWrite	269
7.60.3.7 names	269
7.61 Bds::DataFormatAll Class Reference	269
7.61.1 Detailed Description	270
7.61.2 Constructor & Destructor Documentation	270

7.61.2.1 DataFormatAll()	270
7.61.2.2 ~DataFormatAll()	270
7.61.3 Member Function Documentation	270
7.61.3.1 findFormat()	270
7.61.3.2 formatGet()	271
7.61.3.3 formatGetExtension()	271
7.61.3.4 formatList()	271
7.62 Bds::DataHandle Class Reference	271
7.62.1 Detailed Description	272
7.62.2 Constructor & Destructor Documentation	272
7.62.2.1 DataHandle()	272
7.62.3 Member Data Documentation	272
7.62.3.1 dataFileId	272
7.62.3.2 handle	272
7.63 Bds::DataInfo Class Reference	272
7.63.1 Detailed Description	273
7.63.2 Constructor & Destructor Documentation	273
7.63.2.1 DataInfo()	274
7.63.3 Member Data Documentation	274
7.63.3.1 array	274
7.63.3.2 channels	274
7.63.3.3 description	274
7.63.3.4 endTime	274
7.63.3.5 info	275
7.63.3.6 infoExtra	275
7.63.3.7 startTime	275
7.63.3.8 synchronous	275
7.63.3.9 warnings	275
7.64 Bds::Digitiser Class Reference	276
7.64.1 Detailed Description	277
7.64.2 Constructor & Destructor Documentation	277
7.64.2.1 Digitiser()	277
7.64.3 Member Function Documentation	277
7.64.3.1 getMember()	277
7.64.3.2 getMembers()	277
7.64.3.3 getType()	278
7.64.3.4 setMember()	278
7.64.3.5 setMembers()	278
7.64.4 Member Data Documentation	278
7.64.4.1 baseSamplingFrequency	278
7.64.4.2 endTime	278
7.64.4.3 gain	278

7.64.4.4 id	279
7.64.4.5 initialSamplingFrequency	279
7.64.4.6 name	279
7.64.4.7 numberChannels	279
7.64.4.8 serialNumber	279
7.64.4.9 shared	279
7.64.4.10 startTime	280
7.64.4.11 type	280
7.65 Bds::Event Class Reference	280
7.65.1 Detailed Description	281
7.65.2 Constructor & Destructor Documentation	281
7.65.2.1 Event()	281
7.65.3 Member Data Documentation	282
7.65.3.1 dataChannels	282
7.65.3.2 description	282
7.65.3.3 elevation	282
7.65.3.4 endTime	282
7.65.3.5 eventTime	283
7.65.3.6 extra	283
7.65.3.7 id	283
7.65.3.8 latitude	283
7.65.3.9 longitude	283
7.65.3.10 magnitude	283
7.65.3.11 magnitudeUnits	284
7.65.3.12 network	284
7.65.3.13 notes	284
7.65.3.14 source	284
7.65.3.15 startTime	284
7.65.3.16 title	284
7.65.3.17 type	285
7.65.3.18 userId	285
7.65.3.19 waterDepth	285
7.66 Bds::Fap Class Reference	285
7.66.1 Detailed Description	286
7.66.2 Constructor & Destructor Documentation	286
7.66.2.1 Fap()	286
7.66.3 Member Data Documentation	286
7.66.3.1 amplitude	286
7.66.3.2 frequency	286
7.66.3.3 phase	286
7.67 Bds::Fir Class Reference	287
7.67.1 Detailed Description	287

7.67.2 Constructor & Destructor Documentation	287
7.67.2.1 Fir()	287
7.67.3 Member Data Documentation	287
7.67.3.1 a	287
7.67.3.2 b	288
7.68 Bds::FirEntry Class Reference	288
7.68.1 Detailed Description	288
7.68.2 Constructor & Destructor Documentation	288
7.68.2.1 FirEntry()	288
7.68.3 Member Data Documentation	288
7.68.3.1 coefficient	289
7.68.3.2 error	289
7.69 Bds::GcfChannel Struct Reference	289
7.69.1 Detailed Description	289
7.69.2 Member Data Documentation	289
7.69.2.1 channel	289
7.69.2.2 format	290
7.69.2.3 sampleRate	290
7.69.2.4 streamId	290
7.69.2.5 systemId	290
7.69.2.6 type	290
7.70 Bds::Group Class Reference	290
7.70.1 Detailed Description	291
7.70.2 Constructor & Destructor Documentation	291
7.70.2.1 Group()	291
7.70.3 Member Function Documentation	291
7.70.3.1 getMember()	291
7.70.3.2 getMembers()	292
7.70.3.3 getType()	292
7.70.3.4 setMember()	292
7.70.3.5 setMembers()	292
7.70.4 Member Data Documentation	292
7.70.4.1 description	292
7.70.4.2 group	293
7.70.4.3 id	293
7.71 Bds::ListRange Class Reference	293
7.71.1 Detailed Description	294
7.71.2 Constructor & Destructor Documentation	294
7.71.2.1 ListRange()	294
7.71.3 Member Function Documentation	294
7.71.3.1 getMember()	294
7.71.3.2 getMembers()	294

7.71.3.3	getType()	294
7.71.3.4	setMember()	295
7.71.3.5	setMembers()	295
7.71.4	Member Data Documentation	295
7.71.4.1	number	295
7.71.4.2	reverse	295
7.71.4.3	start	295
7.72	Bds::Location Class Reference	296
7.72.1	Detailed Description	297
7.72.2	Constructor & Destructor Documentation	297
7.72.2.1	Location()	297
7.72.3	Member Function Documentation	297
7.72.3.1	getMember()	297
7.72.3.2	getMembers()	297
7.72.3.3	getType()	298
7.72.3.4	setMember()	298
7.72.3.5	setMembers()	298
7.72.4	Member Data Documentation	298
7.72.4.1	arrayOffsetEast	298
7.72.4.2	arrayOffsetNorth	298
7.72.4.3	channel	298
7.72.4.4	datum	299
7.72.4.5	elevation	299
7.72.4.6	endTime	299
7.72.4.7	id	299
7.72.4.8	latitude	299
7.72.4.9	longitude	299
7.72.4.10	network	300
7.72.4.11	startTime	300
7.72.4.12	station	300
7.73	Bds::Log Class Reference	300
7.73.1	Detailed Description	301
7.73.2	Constructor & Destructor Documentation	301
7.73.2.1	Log()	301
7.73.3	Member Function Documentation	301
7.73.3.1	getMember()	301
7.73.3.2	getMembers()	302
7.73.3.3	getType()	302
7.73.3.4	setMember()	302
7.73.3.5	setMembers()	302
7.73.4	Member Data Documentation	302
7.73.4.1	description	302

7.73.4.2 id	303
7.73.4.3 priority	303
7.73.4.4 subSystem	303
7.73.4.5 time	303
7.73.4.6 title	303
7.73.4.7 type	303
7.74 Bds::LogSelect Class Reference	304
7.74.1 Detailed Description	304
7.74.2 Constructor & Destructor Documentation	304
7.74.2.1 LogSelect()	304
7.74.3 Member Data Documentation	304
7.74.3.1 priority	305
7.74.3.2 startTime	305
7.74.3.3 subSystem	305
7.74.3.4 type	305
7.75 Bds::Network Class Reference	305
7.75.1 Detailed Description	306
7.75.2 Constructor & Destructor Documentation	306
7.75.2.1 Network()	306
7.75.3 Member Function Documentation	306
7.75.3.1 getMember()	307
7.75.3.2 getMembers()	307
7.75.3.3 getType()	307
7.75.3.4 setMember()	307
7.75.3.5 setMembers()	307
7.75.4 Member Data Documentation	307
7.75.4.1 description	308
7.75.4.2 id	308
7.75.4.3 network	308
7.75.4.4 stations	308
7.76 Bds::Note Class Reference	308
7.76.1 Detailed Description	310
7.76.2 Constructor & Destructor Documentation	310
7.76.2.1 Note()	310
7.76.3 Member Function Documentation	310
7.76.3.1 getMember()	310
7.76.3.2 getMembers()	311
7.76.3.3 getType()	311
7.76.3.4 setMember()	311
7.76.3.5 setMembers()	311
7.76.4 Member Data Documentation	311
7.76.4.1 channel	311

7.76.4.2 dataFileId	312
7.76.4.3 description	312
7.76.4.4 docFormat	312
7.76.4.5 docUrl	312
7.76.4.6 endTime	312
7.76.4.7 errorNumber	312
7.76.4.8 eventId	313
7.76.4.9 id	313
7.76.4.10 importFilename	313
7.76.4.11 network	313
7.76.4.12 source	313
7.76.4.13 startTime	313
7.76.4.14 station	314
7.76.4.15 timeAdded	314
7.76.4.16 title	314
7.76.4.17 type	314
7.76.4.18 user	314
7.77 Bds::Point Class Reference	314
7.77.1 Detailed Description	315
7.77.2 Constructor & Destructor Documentation	315
7.77.2.1 Point()	315
7.77.3 Member Data Documentation	315
7.77.3.1 x	315
7.77.3.2 y	315
7.78 Bds::PoleZero Class Reference	316
7.78.1 Detailed Description	316
7.78.2 Constructor & Destructor Documentation	316
7.78.2.1 PoleZero()	316
7.78.3 Member Data Documentation	316
7.78.3.1 poles	316
7.78.3.2 zeros	317
7.79 Bds::Polynomial Class Reference	317
7.79.1 Detailed Description	318
7.79.2 Constructor & Destructor Documentation	318
7.79.2.1 Polynomial()	318
7.79.3 Member Data Documentation	318
7.79.3.1 approximationLowerBound	318
7.79.3.2 approximationType	318
7.79.3.3 approximationUpperBound	318
7.79.3.4 coefficients	319
7.79.3.5 frequencyLowerBound	319
7.79.3.6 frequencyUpperBound	319

7.79.3.7 maximumError	319
7.79.3.8 transferType	319
7.79.3.9 validFrequencyUnits	319
7.80 Bds::PolynomialEntry Class Reference	320
7.80.1 Detailed Description	320
7.80.2 Constructor & Destructor Documentation	320
7.80.2.1 PolynomialEntry()	320
7.80.3 Member Data Documentation	320
7.80.3.1 coefficient	321
7.80.3.2 measurementMethod	321
7.80.3.3 minusError	321
7.80.3.4 plusError	321
7.81 Bds::Response Class Reference	321
7.81.1 Detailed Description	323
7.81.2 Constructor & Destructor Documentation	323
7.81.2.1 Response()	323
7.81.3 Member Data Documentation	324
7.81.3.1 channel	324
7.81.3.2 decimation	324
7.81.3.3 decimationCorr	324
7.81.3.4 decimationDelay	325
7.81.3.5 decimationOffset	325
7.81.3.6 description	325
7.81.3.7 endTime	325
7.81.3.8 faps	325
7.81.3.9 fir	325
7.81.3.10 gain	326
7.81.3.11 gainFrequency	326
7.81.3.12 id	326
7.81.3.13 inputUnits	326
7.81.3.14 inputUnitsDesc	326
7.81.3.15 measured	326
7.81.3.16 name	327
7.81.3.17 network	327
7.81.3.18 outputUnits	327
7.81.3.19 outputUnitsDesc	327
7.81.3.20 poleZeros	327
7.81.3.21 polynomial	327
7.81.3.22 sampleRate	328
7.81.3.23 source	328
7.81.3.24 stage	328
7.81.3.25 stageType	328

7.81.3.26	startTime	328
7.81.3.27	station	328
7.81.3.28	symmetry	329
7.81.3.29	type	329
7.82	Bds::ResponseObj Class Reference	329
7.82.1	Detailed Description	329
7.82.2	Constructor & Destructor Documentation	330
7.82.2.1	ResponseObj()	330
7.82.2.2	~ResponseObj()	330
7.82.3	Member Function Documentation	330
7.82.3.1	getString()	330
7.82.3.2	setString()	330
7.83	Bds::Selection Class Reference	330
7.83.1	Detailed Description	332
7.83.2	Constructor & Destructor Documentation	332
7.83.2.1	Selection()	332
7.83.3	Member Data Documentation	332
7.83.3.1	array	332
7.83.3.2	calibrationName	333
7.83.3.3	channelId	333
7.83.3.4	channels	333
7.83.3.5	completeSegments	333
7.83.3.6	dataTypes	333
7.83.3.7	digitiserId	333
7.83.3.8	endTime	334
7.83.3.9	eventId	334
7.83.3.10	excludeChannels	334
7.83.3.11	id	334
7.83.3.12	locationSelect	334
7.83.3.13	name	334
7.83.3.14	range	335
7.83.3.15	sensorId	335
7.83.3.16	sensorOldId	335
7.83.3.17	startTime	335
7.84	Bds::SelectionChannel Class Reference	335
7.84.1	Detailed Description	336
7.84.2	Constructor & Destructor Documentation	336
7.84.2.1	SelectionChannel()	336
7.84.3	Member Data Documentation	336
7.84.3.1	channel	336
7.84.3.2	network	336
7.84.3.3	source	336

7.84.3.4 station	337
7.85 Bds::SelectionInfo Class Reference	337
7.85.1 Detailed Description	337
7.85.2 Constructor & Destructor Documentation	338
7.85.2.1 SelectionInfo()	338
7.85.3 Member Data Documentation	338
7.85.3.1 arrays	338
7.85.3.2 arraysAndStations	338
7.85.3.3 channels	338
7.85.3.4 endTime	339
7.85.3.5 networks	339
7.85.3.6 numDataChannels	339
7.85.3.7 sources	339
7.85.3.8 startTime	339
7.85.3.9 stations	339
7.86 Bds::Sensor Class Reference	340
7.86.1 Detailed Description	341
7.86.2 Constructor & Destructor Documentation	341
7.86.2.1 Sensor()	341
7.86.3 Member Function Documentation	341
7.86.3.1 getMember()	341
7.86.3.2 getMembers()	341
7.86.3.3 getType()	342
7.86.3.4 setMember()	342
7.86.3.5 setMembers()	342
7.86.4 Member Data Documentation	342
7.86.4.1 endTime	342
7.86.4.2 gain	342
7.86.4.3 gainUnits	342
7.86.4.4 id	343
7.86.4.5 name	343
7.86.4.6 numberChannels	343
7.86.4.7 oldId	343
7.86.4.8 serialNumber	343
7.86.4.9 shared	343
7.86.4.10 startTime	344
7.86.4.11 type	344
7.87 Bds::Source Class Reference	344
7.87.1 Detailed Description	345
7.87.2 Constructor & Destructor Documentation	345
7.87.2.1 Source()	345
7.87.3 Member Function Documentation	345

7.87.3.1 getMember()	345
7.87.3.2 getMembers()	346
7.87.3.3 getType()	346
7.87.3.4 setMember()	346
7.87.3.5 setMembers()	346
7.87.4 Member Data Documentation	346
7.87.4.1 alias	346
7.87.4.2 description	347
7.87.4.3 id	347
7.87.4.4 source	347
7.87.4.5 sourceMeta	347
7.88 Bds::SourcePriority Class Reference	347
7.88.1 Detailed Description	348
7.88.2 Constructor & Destructor Documentation	348
7.88.2.1 SourcePriority()	348
7.88.3 Member Function Documentation	348
7.88.3.1 getMember()	349
7.88.3.2 getMembers()	349
7.88.3.3 getType()	349
7.88.3.4 setMember()	349
7.88.3.5 setMembers()	349
7.88.4 Member Data Documentation	349
7.88.4.1 endTime	350
7.88.4.2 id	350
7.88.4.3 priority	350
7.88.4.4 source	350
7.88.4.5 startTime	350
7.89 Bds::SpecialChannel Class Reference	351
7.89.1 Constructor & Destructor Documentation	351
7.89.1.1 SpecialChannel()	352
7.89.2 Member Function Documentation	352
7.89.2.1 getMember()	352
7.89.2.2 getMembers()	352
7.89.2.3 getType()	352
7.89.2.4 setMember()	352
7.89.2.5 setMembers()	353
7.89.3 Member Data Documentation	353
7.89.3.1 channel	353
7.89.3.2 dataType	353
7.89.3.3 description	353
7.89.3.4 endTime	353
7.89.3.5 id	353

7.89.3.6 network	354
7.89.3.7 startTime	354
7.89.3.8 station	354
7.90 Bds::Station Class Reference	354
7.90.1 Detailed Description	355
7.90.2 Constructor & Destructor Documentation	355
7.90.2.1 Station()	355
7.90.3 Member Data Documentation	355
7.90.3.1 alias	355
7.90.3.2 channels	355
7.90.3.3 description	355
7.90.3.4 id	356
7.90.3.5 name	356
7.90.3.6 network	356
7.90.3.7 type	356
7.91 Bds::TimePeriod Class Reference	356
7.91.1 Detailed Description	357
7.91.2 Constructor & Destructor Documentation	357
7.91.2.1 TimePeriod()	357
7.91.3 Member Function Documentation	357
7.91.3.1 getMember()	357
7.91.3.2 getMembers()	358
7.91.3.3 getType()	358
7.91.3.4 setMember()	358
7.91.3.5 setMembers()	358
7.91.4 Member Data Documentation	358
7.91.4.1 endTime	358
7.91.4.2 startTime	359
7.92 Bds::User Class Reference	359
7.92.1 Detailed Description	360
7.92.2 Constructor & Destructor Documentation	360
7.92.2.1 User()	360
7.92.3 Member Function Documentation	360
7.92.3.1 getMember()	360
7.92.3.2 getMembers()	360
7.92.3.3 getType()	361
7.92.3.4 setMember()	361
7.92.3.5 setMembers()	361
7.92.4 Member Data Documentation	361
7.92.4.1 address	361
7.92.4.2 email	361
7.92.4.3 enabled	361

7.92.4.4 groups	362
7.92.4.5 id	362
7.92.4.6 name	362
7.92.4.7 password	362
7.92.4.8 telephone	362
7.92.4.9 user	362
8 File Documentation	363
8.1 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp File Reference	363
8.2 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.d File Reference	363
8.3 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.h File Reference	363
8.4 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp File Reference	364
8.5 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.d File Reference	364
8.6 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h File Reference	364
8.7 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp File Reference	365
8.8 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.d File Reference	365
8.9 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h File Reference	365
8.10 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp File Reference	365
8.10.1 Macro Definition Documentation	366
8.10.1.1 DEBUG_VELATRACK	366
8.11 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.d File Reference	366
8.12 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h File Reference	366
8.13 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp File Reference	366
8.14 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.d File Reference	367
8.15 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h File Reference	367
8.16 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp File Reference	367
8.17 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.d File Reference	367
8.18 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h File Reference	367
8.19 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp File Reference	368
8.19.1 Macro Definition Documentation	369
8.19.1.1 ALLOW_TIMESTAMP_JITTER	369
8.19.1.2 dl2printf	369
8.19.1.3 dl3printf	369
8.19.1.4 dlprintf	369
8.19.1.5 LDEBUG	369
8.19.1.6 LDEBUG2	369
8.19.1.7 LDEBUG3	370
8.19.1.8 TIMESTAMP_JITTER	370
8.20 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.d File Reference	370
8.21 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h File Reference	370
8.22 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp File Reference	371
8.22.1 Function Documentation	371

8.22.1.1 clip()	371
8.23 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.d File Reference	371
8.24 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h File Reference	371
8.25 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.cpp File Reference	372
8.25.1 Macro Definition Documentation	373
8.25.1.1 ALLOW_TIMESTAMP_JITTER	373
8.25.1.2 dprintf	373
8.25.1.3 htonl	373
8.25.1.4 INCLUDE_CHANNEL_AUTH	373
8.25.1.5 LDEBUG	373
8.25.1.6 MULTIPLE_SEGMENT	373
8.25.1.7 ntohl	374
8.25.1.8 SEGMENT_GAP	374
8.25.1.9 TIMESTAMP_JITTER	374
8.25.2 Variable Documentation	374
8.25.2.1 ErrorFormatNoDataFormat	374
8.26 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.d File Reference	374
8.27 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h File Reference	374
8.28 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp File Reference	375
8.29 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.d File Reference	375
8.30 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h File Reference	375
8.31 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp File Reference	375
8.31.1 Macro Definition Documentation	376
8.31.1.1 DEBUG	376
8.31.1.2 TEST_REORDER	376
8.32 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.d File Reference	376
8.33 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h File Reference	376
8.34 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.cpp File Reference	377
8.34.1 Macro Definition Documentation	377
8.34.1.1 dprintf	377
8.34.1.2 LDEBUG	377
8.35 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.d File Reference	377
8.36 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.h File Reference	377
8.37 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp File Reference	378
8.38 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.d File Reference	378
8.39 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h File Reference	378
8.40 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp File Reference	379
8.41 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.d File Reference	379
8.42 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h File Reference	379
8.43 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.cpp File Reference	379
8.44 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.d File Reference	380
8.45 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.h File Reference	380

8.46 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp File Reference	380
8.46.1 Macro Definition Documentation	381
8.46.1.1 dprintf	381
8.46.1.2 LDEBUG	381
8.47 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.d File Reference	381
8.48 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h File Reference	381
8.49 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp File Reference	381
8.49.1 Macro Definition Documentation	382
8.49.1.1 BDEBUGL1	382
8.50 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.d File Reference	382
8.51 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.h File Reference	382
8.52 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.cpp File Reference	382
8.52.1 Macro Definition Documentation	383
8.52.1.1 BDEBUGL1	383
8.52.1.2 BDEBUGL2	383
8.53 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.d File Reference	383
8.54 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.h File Reference	383
8.55 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp File Reference	384
8.56 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.d File Reference	384
8.57 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h File Reference	384
8.58 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp File Reference	385
8.59 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.d File Reference	385
8.60 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h File Reference	385
8.61 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp File Reference	386
8.61.1 Function Documentation	386
8.61.1.1 parseStringFixedFields()	386
8.62 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.d File Reference	386
8.63 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h File Reference	386
8.64 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp File Reference	387
8.65 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.d File Reference	387
8.66 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h File Reference	387
8.67 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp File Reference	388
8.67.1 Macro Definition Documentation	389
8.67.1.1 BDEBUGL1	389
8.67.1.2 BDEBUGL2	389
8.67.1.3 BDEBUGL3	389
8.67.1.4 DEBUG	389
8.67.1.5 DEBUG_BLOCKETTE	389
8.67.1.6 DEBUG_BLOCKS	389
8.67.1.7 FILL_BLOCKS	389
8.67.1.8 ROUND_TIMESTAMPS_US	390
8.68 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.d File Reference	390

8.69 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h File Reference	390
8.69.1 Typedef Documentation	390
8.69.1.1 MSRecord	390
8.70 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp File Reference	390
8.71 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.d File Reference	391
8.72 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.h File Reference	391
8.73 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.cpp File Reference	391
8.74 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.d File Reference	391
8.75 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.idl File Reference	391
8.76 /src/blacknest/bds/bds/bdsDataLib/canada_compress.d File Reference	391
8.77 /src/blacknest/bds/bds/bdsDataLib/canada_compress.h File Reference	391
8.77.1 Macro Definition Documentation	392
8.77.1.1 CANCOMP_CORRUPT	392
8.77.1.2 CANCOMP_ERR	392
8.77.1.3 CANCOMP_EXCEED	392
8.77.1.4 CANCOMP_NOT_20	392
8.77.1.5 CANCOMP_SUCCESS	393
8.77.2 Function Documentation	393
8.77.2.1 canada_compress()	393
8.77.2.2 canada_uncompress()	393
8.78 BdsC.cc File Reference	393
8.79 BdsC.d File Reference	394
8.80 BdsC.h File Reference	394
8.81 BdsD.cc File Reference	394
8.82 BdsD.d File Reference	395
8.83 BdsD.h File Reference	395
8.83.1 Detailed Description	398
8.84 BdsLib.cpp File Reference	398
8.85 BdsLib.d File Reference	400
8.86 BdsLib.dox File Reference	400
8.87 BdsLib.h File Reference	400
8.87.1 Detailed Description	403
8.88 BdsS.cc File Reference	403
8.89 BdsS.d File Reference	403
8.90 BdsT.cc File Reference	403
8.91 /src/blacknest/bds/bds/doc/bdsApiOverview.dox File Reference	403

Index

405

Chapter 1

Main Page

Author

Dr Terry Barnaby

Version

3.0.12

Date

2022-06-23

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 IPAddress/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:

Bds	21
---------------------	-------	--------------------

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Bds::ArrayChannel	83
BBuffer[external]	
BBufferStore[external]	
Bds::BdsDataPacket	89
Bds::BdsDataBlock	85
Bds::BdsDataBlockHeader	86
Bds::BdsDataBlockPos	87
Bds::BdsDataPacketHeader	91
Bds::BdsDataSegment	93
Bds::BdsDataStreamlet	95
Bds::BdsSeedType	96
Bds::DataFileCssData	225
BObj[external]	
Bds::AccessGroup	51
Bds::Calibration	99
Bds::Change	114
Bds::ChangeGroup	118
Bds::Channel	121
Bds::ChannelInstrument	130
Bds::DataChannel	181
Bds::DataFileInfo	237
Bds::Digitiser	276
Bds::Group	290
Bds::ListRange	293
Bds::Location	296
Bds::Log	300
Bds::Network	305
Bds::Note	308
Bds::Sensor	340
Bds::Source	344
Bds::SourcePriority	347
Bds::SpecialChannel	351
Bds::TimePeriod	356
Bds::User	359
BSocket[external]	

BoapClientObject[external]	
Bds::AdminAccess	54
Bds::DataAccess	138
Bds::DataAddAccess	154
BoapClientObject[external]	
Bds::CdChannel_1v0	106
Bds::CdDataChannel	108
Bds::CdDataFormatFrame_1v0	109
Bds::CdFlag	111
Bds::CdPacketData	112
Bds::ChannelInfo	125
Bds::ChannelInfos	129
Bds::ChannelName	134
Bds::CleanOptions	136
Bds::CompressSteim1	137
Bds::DataAvail	171
Bds::DataAvailChan	172
Bds::DataBlock	175
Bds::DataBlockChannel	177
Bds::DataBlockPos	179
Bds::DataCollate	187
Bds::DataError	188
Bds::DataFile	194
Bds::DataFileAd22	204
Bds::DataFileAscii	206
Bds::DataFileBdrs	209
Bds::DataFileBds	212
Bds::DataFileBknas	218
Bds::DataFileCd	220
Bds::DataFileCss	223
Bds::DataFileGcf	230
Bds::DataFileIdc	232
Bds::DataFileIms	234
Bds::DataFileLac	242
Bds::DataFileLog	244
Bds::DataFileResponse	249
Bds::DataFileSac	251
Bds::DataFileSeed	253
Bds::DataFileStationXml	258
Bds::DataFileTapeDigitiser	260
Bds::DataFileWra	262
Bds::DataFileWraAgso	265
Bds::DataFileOptions	248
Bds::DataFormat	267
Bds::DataFormatAll	269
Bds::DataHandle	271
Bds::DataInfo	272
Bds::Event	280
Bds::Fap	285
Bds::Fir	287
Bds::FirEntry	288
Bds::GcfChannel	289
Bds::LogSelect	304
Bds::Point	314
Bds::PoleZero	316
Bds::Polynomial	317
Bds::PolynomialEntry	320
Bds::Response	321

Bds::ResponseObj	329
Bds::Selection	330
Bds::SelectionChannel	335
Bds::SelectionInfo	337
Bds::Station	354

Chapter 4

Class Index

4.1 Class List

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

Bds::AccessGroup	This holds information on data access groups	51
Bds::AdminAccess	This is the AdminAccess Access API interface	54
Bds::ArrayChannel	This class defines an arrays channel	83
Bds::BdsDataBlock	BdsDataFileBds: internal fixed size BDS Data Block	85
Bds::BdsDataBlockHeader	BdsDataFileBds: internal fixed size BDS Data Block header	86
Bds::BdsDataBlockPos	BdsDataFileBds: internal file storage data block position	87
Bds::BdsDataPacket	BdsDataFileBds: internal file storage packet	89
Bds::BdsDataPacketHeader	BdsDataFileBds internal file storage packet header	91
Bds::BdsDataSegment	BdsDataFileBds: internal file storage data segment	93
Bds::BdsDataStreamlet	BdsDataFileBds: internal file storage data streamlet	95
Bds::BdsSeedType	BdsDataFileSeed internal parent for all SEED types	96
Bds::Calibration	This class defines a calibration setting	99
Bds::CdChannel_1v0	BdsDataFile: Internal CD1.0 channel information	106
Bds::CdDataChannel	BdsDataFile: Internal CD channel information	108
Bds::CdDataFormatFrame_1v0	BdsDataFile: Internal CD1.0 frame information	109
Bds::CdFlag	BdsDataFile: Internal CD flag	111
Bds::CdPacketData	BdsDataFile: Internal CD data packet	112
Bds::Change	This holds information on a medatdata or sensor data change	114

Bds::ChangeGroup	
This holds information on a set of Changes	118
Bds::Channel	
This class defines a seismic data Channel	121
Bds::ChannelInfo	
This class provides information on a channel	125
Bds::ChannelInfos	
This class provides metadata information on a set of channels	129
Bds::ChannelInstrument	
This class defines a Channel 's instrument	130
Bds::ChannelName	
This class defines a full channel name	134
Bds::CleanOptions	
This defines the set of clean options used in the clean() function	136
Bds::CompressSteim1	
Steim1 un-compress class	137
Bds::DataAccess	
This is the Data Access API interface	138
Bds::DataAddAccess	
This is the DataAdd Access API interface	154
Bds::DataAvail	
This class provides availability information on a particular period of data	171
Bds::DataAvailChan	
This class defines availability information on a set of data	172
Bds::DataBlock	
This class provides the actual seismic data values contained within a single data block	175
Bds::DataBlockChannel	
This class provides the actual seismic data values contained within a single data block along with the network:station:channel:source information	177
Bds::DataBlockPos	
This defines the position of a data block in a file. It is used by the BDS data converters to order blocks by time	179
Bds::DataChannel	
This class defines information on a single channels set of data stored in a file	181
Bds::DataCollate	
Not sure if this is used or what it does	187
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	188
Bds::DataFile	
This class defines the interface for generic data file access that all of the BDS data converters share	194
Bds::DataFileAd22	
Data file convertor for AD22 format files	204
Bds::DataFileAscii	
Data file convertor for ASCII format files	206
Bds::DataFileBdrs	
Data file convertor for BDRS format files	209
Bds::DataFileBds	
This class implements the BDS Data File/Stream access system	212
Bds::DataFileBknas	
Data file convertor for BKNAS format files	218
Bds::DataFileCd	
Data file convertor for CD1.0 and CD1.1 file formats	220
Bds::DataFileCss	
Data file convertor for CSS format files	223
Bds::DataFileCssData	
DataFileCss internal CSS data type	225

Bds::DataFileGcf	
Data file convertor for GCF format files	230
Bds::DataFileIdc	
This class defines the interface for IDC response data file access	232
Bds::DataFileIms	
Data file convertor for IMS format files	234
Bds::DataFileInfo	
This class defines information on a sensor data file	237
Bds::DataFileLac	
Data file convertor for LAC format files	242
Bds::DataFileLog	
Data file convertor for LOG format files	244
Bds::DataFileOptions	
This defines a list of BDS data converter options	248
Bds::DataFileResponse	
This class defines the interface for generic response data file access	249
Bds::DataFileSac	
Data file convertor for SAC format files	251
Bds::DataFileSeed	
Data file convertor for SEED file formats	253
Bds::DataFileStationXml	
This class defines the interface for generic response data file access	258
Bds::DataFileTapeDigitiser	
This class implements the TapeDigitiser's file output conversion and storing system	260
Bds::DataFileWra	
Data file convertor for WRA format files	262
Bds::DataFileWraAgso	
Data file convertor for WRA AGSO format files	265
Bds::DataFormat	
This holds information on a seismic data format	267
Bds::DataFormatAll	
This class defines the interface for generic data file access	269
Bds::DataHandle	
This defines a handle to a sensor data stream/file when opened for read or write	271
Bds::DataInfo	
This class defines information on a set of data	272
Bds::Digitiser	
This class defines a seismic Digitiser	276
Bds::Event	
This class defines a seismic event	280
Bds::Fap	
This class defines an entry in an Amplitude/Phase Response table	285
Bds::Fir	
This class defines an FIR response table	287
Bds::FirEntry	
This class defines an entry in a FIR coefficient table	288
Bds::GcfChannel	
DataFileGcf internal GCF channel information	289
Bds::Group	
This holds information on a user security group	290
Bds::ListRange	
This class defines an integer based range	293
Bds::Location	
This class defines the physical location of a Station	296
Bds::Log	
This holds information on a Log entry	300
Bds::LogSelect	
This defines the selection criteria when requesting a set of log entries	304

Bds::Network	This class defines a seismic Network organisation	305
Bds::Note	This holds information on a Note for general information	308
Bds::Point	This class defines an X,Y location	314
Bds::PoleZero	This class defines a Pole/Zero Response	316
Bds::Polynomial	This class defines an Polynomial response table	317
Bds::PolynomialEntry	This class defines an entry in a Polynomial coefficient table	320
Bds::Response	This class defines a seismic Response characteristic	321
Bds::ResponseObj	Response object adding string conversion	329
Bds::Selection	This class defines a generic metadata or seismic data selection	330
Bds::SelectionChannel	This class defines a channel for selection	335
Bds::SelectionInfo	This class defines the set of metadata or seismic data selected when getSelectionInfo() is use	337
Bds::Sensor	This class defines a seismic Sensor	340
Bds::Source	This class defines a seismic data Source	344
Bds::SourcePriority	This class defines a Source Priority entry	347
Bds::SpecialChannel	351
Bds::Station	This class defines a seismic station	354
Bds::TimePeriod	This class defines a TimePeriod	356
Bds::User	This holds information on a user	359

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	363
/src/blacknest/bds/bds/bdsDataLib/BdsCompress.d	363
/src/blacknest/bds/bds/bdsDataLib/BdsCompress.h	363
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp	364
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.d	364
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h	364
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp	365
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.d	365
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h	365
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp	365
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.d	366
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h	366
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp	366
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.d	367
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h	367
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp	367
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.d	367
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h	367
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp	368
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.d	370
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h	370
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp	371
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.d	371
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h	371
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.cpp	372
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.d	374
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h	374
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp	375
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.d	375
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h	375
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp	375
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.d	376
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h	376
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.cpp	377
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.d	377

/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.h	377
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp	378
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.d	378
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h	378
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp	379
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.d	379
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h	379
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.cpp	379
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.d	380
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.h	380
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp	380
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.d	381
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h	381
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp	381
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.d	382
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.h	382
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.cpp	382
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.d	383
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.h	383
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp	384
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.d	384
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h	384
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp	385
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.d	385
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h	385
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp	386
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.d	386
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h	386
/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp	387
/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.d	387
/src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h	387
/src/blacknest/bds/bds/bdsDataLib/canada_compress.d	391
/src/blacknest/bds/bds/bdsDataLib/canada_compress.h	391
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp	388
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.d	390
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h	390
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp	390
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.d	391
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.h	391
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.cpp	391
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.d	391
/src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.idl	391
BdsC.cc	393
BdsC.d	394
BdsC.h	394
BdsD.cc	394
BdsD.d	395
BdsD.h	
BOAP data class definitions for: Bds	395
BdsLib.cpp	398
BdsLib.d	400
BdsLib.h	
General BdsLib API functions	400
BdsS.cc	403
BdsS.d	403
BdsT.cc	403

Chapter 6

Namespace Documentation

6.1 Bds Namespace Reference

Classes

- class [DataAccess](#)
This is the Data Access API interface.
- class [DataAddAccess](#)
This is the DataAdd Access API interface.
- class [AdminAccess](#)
This is the [AdminAccess](#) Access API interface.
- class [Point](#)
This class defines an X,Y location.
- class [TimePeriod](#)
This class defines a [TimePeriod](#).
- class [ListRange](#)
This class defines an integer based range.
- class [Network](#)
This class defines a seismic [Network](#) organisation.
- class [Source](#)
This class defines a seismic data [Source](#).
- class [SourcePriority](#)
This class defines a [Source](#) Priority entry.
- class [ChannelName](#)
This class defines a full channel name.
- class [ArrayChannel](#)
This class defines an arrays channel.
- class [Station](#)
This class defines a seismic station.
- class [Location](#)
This class defines the physical location of a [Station](#).
- class [PoleZero](#)
This class defines a Pole/Zero [Response](#).
- class [Fap](#)
This class defines an entry in an Amplitude/Phase [Response](#) table.
- class [FirEntry](#)

- This class defines an entry in a FIR coefficient table.*

 - class [Fir](#)
- This class defines an FIR response table.*

 - class [PolynomialEntry](#)
- This class defines an entry in a [Polynomial](#) coefficient table.*

 - class [Polynomial](#)
- This class defines an [Polynomial](#) response table.*

 - class [Response](#)
- This class defines a seismic [Response](#) characteristic.*

 - class [Calibration](#)
- This class defines a calibration setting.*

 - class [Digitiser](#)
- This class defines a seismic [Digitiser](#).*

 - class [Sensor](#)
- This class defines a seismic [Sensor](#).*

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

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

 - class [SelectionInfo](#)
- This class defines the set of metadata or seismic data selected when `getSelectionInfo()` is use.*

 - class [SelectionChannel](#)
- This class defines a channel for selection.*

 - class [Selection](#)
- This class defines a generic metadata or seismic data selection.*

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

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

 - class [DataFileInfo](#)
- This class defines information on a sensor data file.*

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

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

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

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

 - class [DataHandle](#)
- This defines a handle to a sensor data stream/file when opened for read or write.*

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

 - class [DataBlockChannel](#)
- This class provides the actual seismic data values contained within a single data block along with the network↔:station:channel:source information.*

 - class [User](#)
- This holds information on a user.*

 - class [Group](#)
- This holds information on a user security group.*

 - class [AccessGroup](#)

- This holds information on data access groups.*

 - class [Change](#)

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

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

This holds information on a [Note](#) for general information.
 - class [Log](#)

This holds information on a [Log](#) entry.
 - class [LogSelect](#)

This defines the selection cirteria when requesting a set of log entries.
 - class [CleanOptions](#)

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

This holds information on a seismic data format.
 - class [SpecialChannel](#)
 - class [Event](#)

This class defines a seismic event.
 - class [ResponseObj](#)

[Response](#) object adding string conversion.
 - class [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.
 - class [CompressSteim1](#)

Steim1 un-compress class.
 - class [DataCollate](#)

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

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

This defines a list of BDS data converter options.
 - class [DataFile](#)

This class defines the interface for generic data file access that all of the BDS data converters 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.
 - struct [BdsDataBlockHeader](#)

BdsDataFileBds: internal fixed size BDS Data Block header.
 - struct [BdsDataBlock](#)

BdsDataFileBds: internal fixed size BDS Data Block.
 - struct [BdsDataPacketHeader](#)

BdsDataFileBds internal file storage packet header.
 - class [BdsDataPacket](#)

BdsDataFileBds: internal file storage packet.
 - class [BdsDataBlockPos](#)

BdsDataFileBds: internal file storage data block position.
 - class [BdsDataSegment](#)

BdsDataFileBds: internal file storage data segment.

- class [BdsDataStreamlet](#)
BdsDataFileBds: internal file storage data streamlet.
- class [DataFileBds](#)
This class implements the BDS Data File/Stream access system.
- class [DataFileBknas](#)
Data file convertor for BKNAS format files.
- struct [CdChannel_1v0](#)
BdsDataFile: Internal CD1.0 channel information.
- struct [CdDataFormatFrame_1v0](#)
BdsDataFile: Internal CD1.0 frame information.
- class [CdDataChannel](#)
BdsDataFile: Internal CD channel information.
- class [CdPacketData](#)
BdsDataFile: Internal CD data packet.
- class [CdFlag](#)
BdsDataFile: Internal CD flag.
- class [DataFileCd](#)
Data file convertor for CD1.0 and CD1.1 file formats.
- class [DataFileCssData](#)
DataFileCss internal CSS data type.
- class [DataFileCss](#)
Data file convertor for CSS format files.
- struct [GcfChannel](#)
DataFileGcf internal GCF channel information.
- class [DataFileGcf](#)
Data file convertor for GCF format files.
- class [DataFileIdc](#)
This class defines the interface for IDC response data file access.
- class [DataFileIms](#)
Data file convertor for IMS format files.
- class [DataFileLac](#)
Data file convertor for LAC format files.
- class [DataFileLog](#)
Data file convertor for LOG format files.
- class [DataFileResponse](#)
This class defines the interface for generic response data file access.
- class [DataFileSac](#)
Data file convertor for SAC format files.
- class [DataFileStationXml](#)
This class defines the interface for generic response data file access.
- 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 [DataFormatAll](#)
This class defines the interface for generic data file access.
- class [DataFileSeed](#)
Data file convertor for SEED file formats.
- class [BdsSeedType](#)
BdsDataFileSeed internal parent for all SEED types.

Typedefs

- typedef **BList**< [DataFormat](#) > [DataFormats](#)

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 [DataFormatSet](#) {
[DataFormatSetNone](#) = 0x00 , [DataFormatSetMetadataRead](#) = 0x01 , [DataFormatSetMetadataWrite](#) = 0x02 ,
[DataFormatSetSensordataRead](#) = 0x04 ,
[DataFormatSetSensordataWrite](#) = 0x08 }
Data format abilities bitset.
- enum [LocationSelect](#) { [LocationSelectAll](#) , [LocationSelectStation](#) , [LocationSelectChannel](#) }
Which Locations to select.
- 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

- BError** [bdsLibInit](#) ([DataAccess](#) &bds)
Initialise the bdsLib with settings from the BdsServer.
- BError** [bdsLibInit](#) ([DataAddAccess](#) &bds)
Initialise the bdsLib with settings from the BdsServer.

- **BError** [bdsLibInit](#) ([AdminAccess](#) &bds)
Initialise the bdsLib with settings from the BdsServer.
- 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 tange 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** [bdsDataInfoFlatten](#) ([DataInfo](#) &dataInfo)
Flattens a [DataInfo](#) to 1 segment per channel for use in dataOpen() calls.
- **BError** [bdsDataInfoMergeFlatten](#) ([DataInfo](#) &dataInfo, const [DataInfo](#) &dataInfoAdd)
Merges a [DataInfo](#) into another flattening the segments to 1 for use in dataOpen() calls.
- **BString** [bdsUnitsConvert](#) ([BString](#) units)
Tidy up units name to standard SI units format.
- static int [responseSort](#) ([Response](#) &a, [Response](#) &b)
- **BError** [bdsMetadataImportFix](#) ([ChannelInfos](#) &channelInfos, [Bool](#) stageRenummer)
Fix up [ChannelInfos](#) from import. Mainly making sure response stages and their units are correct.
- **BError** [bdsMetadataExportFix](#) ([ChannelInfos](#) &channelInfos, [Bool](#) singleResponse, [Bool](#) stageRenummer, [Bool](#) changeUnits, [Bool](#) stageGains, [Bool](#) decimation, [Bool](#) toDisplacement, [Bool](#) toNm)
Fix up [ChannelInfos](#) for export. Mainly making sure response stages and their units are correct.
- **BString** [bdsStationAlias](#) ([Station](#) station)
Returns the station alias if set else its name.
- void [bdsDumpSelection](#) ([Selection](#) sel)
Debug print out a [Selection](#) object.
- void [bdsDumpSelectionInfo](#) ([SelectionInfo](#) sel)
Debug print out a [SelectionInfo](#) 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.
- void [bdsDumpLocation](#) ([Location](#) location)
Debug printout location.
- **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 overall gain of the given [PoleZero](#) transfer function.*
- void [bdsPoleZeroGainPhase](#) (const [PoleZero](#) &poleZero, double frequency, double &gain, double &phase)
 - Calculates the overall 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.*
- **BString** [bdsFileNameExpand](#) (**BString** fileName, [Selection](#) &sel)
 - Default filename from a [Selection](#).*
- void [bdsSpecialChannelsSet](#) (const **BList**< [SpecialChannel](#) > specialChannels)
 - Set the special channels list.*
- **BList**< [SpecialChannel](#) > [bdsSpecialChannels](#) ()
 - Return list of special channels.*
- **Bool** [bdsSpecialChannelIgnore](#) (**BString** network, **BString** station, **BString** channel)
 - Check if channel should be ignored.*
- char [seedChannelInstrumentCode](#) (**BString** dataType)
 - Returns SEED instrument code from dataType.*
- **BString** [seedChannelDataType](#) (**BString** channel)
 - Returns dataType from channel name based on SEED channel name convention.*
- **BStringList** [bdsDataTypes](#) ()
 - Returns all known data types.*
- **BStringList** [bdsUnits](#) ()
 - Returns all known SI units.*
- **BString** [bdsUnitCase](#) (**BString** unit)
 - Converts character case of units.*
- **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** [unitsCode](#) ([Response](#) &r)
- 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 double [roundDigits](#) (double v, int nDigits)

- static void [bdsDataFileSeedLogWarning](#) (char *str)
- static void [bdsDataFileSeedLogError](#) (char *str)
- static hptime_t [seedTime](#) ([BTimeStamp](#) t)
- static [BString](#) [seedTimeString](#) ([BTimeStamp](#) t)
- static [BTimeStamp](#) [fromSeedTimeString](#) ([BString](#) str)
- 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
- static [BList](#)< [SpecialChannel](#) > [bdsSpecialChannelsList](#)
- SeedlCodeToDataType [seedlCodeToDataTypes](#) []
- 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 [crclnitDone](#)
- static char [cm6Table](#) [64]
- static [BUInt8](#) [cm6TableRev](#) [128]
- const char * [node_types](#) []
- const double [Scale](#) = 16777216.0
- [DataFormatAll](#) [dataFormatAll](#)

6.1.1 Typedef Documentation

6.1.1.1 DataFormats

```
typedef BList<DataFormat > Bds::DataFormats
```

6.1.2 Enumeration Type Documentation

6.1.2.1 AvailType

```
enum Bds::AvailType
```

A flag defining the data availability state.

Enumerator

AvailNone	There is no data available There is full data available
AvailPartial	There is partial data available
AvailFull	There is no data available

6.1.2.2 BdsDataType

```
enum Bds::BdsDataType
```

BdsDataFileBds: internal file block type field.

Enumerator

BdsDataTypeBlock	
BdsDataTypeInfo	
BdsDataTypeData	
BdsDataTypeInfoExtra	

6.1.2.3 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.

6.1.2.4 DataFormatSet

```
enum Bds::DataFormatSet
```

Data format abilities bitset.

Enumerator

DataFormatSetNone	
DataFormatSetMetadataRead	
DataFormatSetMetadataWrite	
DataFormatSetSensordataRead	
DataFormatSetSensordataWrite	

6.1.2.5 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
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

6.1.2.6 FileHeaderType

```
enum Bds::FileHeaderType
```

Enumerator

FileHeaderType_Standard	
FileHeaderType_TapeDigitiser	

6.1.2.7 FileSampleType

```
enum Bds::FileSampleType
```

Enumerator

FileSampleType_Unknown	
FileSampleType_Float32	
FileSampleType_Float64	
FileSampleType_Int16	
FileSampleType_Int32	

6.1.2.8 LocationSelect

```
enum Bds::LocationSelect
```

Which Locations to select.

Enumerator

LocationSelectAll	
LocationSelectStation	
LocationSelectChannel	

6.1.2.9 Mode

```
enum Bds::Mode
```

BdsServer mode.

Enumerator

ModeMaster	BdsServer is a master
ModeSlave	BdsServer is a slave

6.1.2.10 Priority

```
enum Bds::Priority
```

Priority levels.

Enumerator

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

6.1.2.11 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

6.1.2.12 SelectionGroup

```
enum Bds::SelectionGroup
```

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

Enumerator

SelectionGroupData	Select items from Sensor data
SelectionGroupMetaData	Select items from Metadata
SelectionGroupDataWithCount	Select items from Sensor data and return the number of items found

6.1.3 Function Documentation

6.1.3.1 bdsChannelGetName()

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

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

6.1.3.2 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

6.1.3.3 bdsDataChannelInfo()

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

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

6.1.3.4 bdsDataChannelOverallResponse()

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

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

6.1.3.5 bdsDataChannelRef() [1/2]

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

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

6.1.3.6 bdsDataChannelRef() [2/2]

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

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

6.1.3.7 bdsDataFileSeedLogError()

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

6.1.3.8 bdsDataFileSeedLogWarning()

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

6.1.3.9 bdsDataInfoFlatten()

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

Flattens a [DataInfo](#) to 1 segment per channel for use in dataOpen() calls.

6.1.3.10 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.

6.1.3.11 bdsDataInfoMergeFlatten()

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

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

6.1.3.12 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.

6.1.3.13 bdsDataTypes()

```
BStringList Bds::bdsDataTypes ( )
```

Returns all known data types.

6.1.3.14 bdsDumpChannelInfos()

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

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

6.1.3.15 bdsDumpData()

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

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

6.1.3.16 bdsDumpDataInfo()

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

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

6.1.3.17 bdsDumpLocation()

```
void Bds::bdsDumpLocation (
    Location location )
```

Debug printout location.

6.1.3.18 bdsDumpPoleZeros()

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

Debug print out a PoleZeros object.

6.1.3.19 bdsDumpSelection()

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

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

6.1.3.20 bdsDumpSelectionInfo()

```
void Bds::bdsDumpSelectionInfo (
    SelectionInfo sel )
```

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

6.1.3.21 bdsFileNameExpand() [1/3]

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

Default filename from a [ChannelInfo](#).

6.1.3.22 bdsFileNameExpand() [2/3]

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

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

6.1.3.23 bdsFileNameExpand() [3/3]

```
BString Bds::bdsFileNameExpand (
    BString fileName,
    Selection & sel )
```

Default filename from a [Selection](#).

6.1.3.24 bdsInfoFromDataInfo()

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

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

6.1.3.25 bdsLibInit() [1/3]

```
BError Bds::bdsLibInit (
    AdminAccess & bds )
```

Initialise the bdsLib with settings from the BdsServer.

6.1.3.26 bdsLibInit() [2/3]

```
BError Bds::bdsLibInit (
    DataAccess & bds )
```

Initialise the bdsLib with settings from the BdsServer.

6.1.3.27 bdsLibInit() [3/3]

```
BError Bds::bdsLibInit (
    DataAddAccess & bds )
```

Initialise the bdsLib with settings from the BdsServer.

6.1.3.28 bdsMetadataExportFix()

```
BError Bds::bdsMetadataExportFix (
    ChannelInfos & channelInfos,
    Bool singleResponse,
    Bool stageRenummer,
    Bool changeUnits,
    Bool stageGains,
    Bool decimation,
    Bool toDisplacement,
    Bool toNm )
```

Fix up [ChannelInfos](#) for export. Mainly making sure response stages and their units are correct.

6.1.3.29 bdsMetadataImportFix()

```
BError Bds::bdsMetadataImportFix (
    ChannelInfos & channelInfos,
    Bool stageRenummer )
```

Fix up [ChannelInfos](#) from import. Mainly making sure response stages and their units are correct.

6.1.3.30 bdsPoleZeroGain()

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

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

6.1.3.31 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.

6.1.3.32 bdsPoleZeroToFap()

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

Convert PoleZero to FAP.

6.1.3.33 bdsSelectionChannelInfo()

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

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

6.1.3.34 bdsSpecialChannelIgnore()

```
Bool Bds::bdsSpecialChannelIgnore (
    BString network,
    BString station,
    BString channel )
```

Check if channel should be ignored.

6.1.3.35 bdsSpecialChannels()

```
BList< SpecialChannel > Bds::bdsSpecialChannels ( )
```

Return list of special channels.

6.1.3.36 bdsSpecialChannelsSet()

```
void Bds::bdsSpecialChannelsSet (
    const BList< SpecialChannel > specialChannels )
```

Set the special channels list.

6.1.3.37 bdsStationAlias()

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

Returns the station alias if set else its name.

6.1.3.38 bdsUnCompressCm8()

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

Uncompress CM8 formatted data.

6.1.3.39 bdsUnCompressSteim1()

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

Uncompress STEIM1 formatted data.

6.1.3.40 bdsUnitCase()

```
BString Bds::bdsUnitCase (
    BString unit )
```

Converts character case of units.

6.1.3.41 bdsUnits()

```
BStringList Bds::bdsUnits ( )
```

Returns all known SI units.

6.1.3.42 bdsUnitsConvert()

```
BString Bds::bdsUnitsConvert (
    BString units )
```

Tidy up units name to standard SI units format.

6.1.3.43 crc()

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

6.1.3.44 crc64()

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

6.1.3.45 crcInit()

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

6.1.3.46 dataCalculateDifference()

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

6.1.3.47 dataCalculateUnDifference()

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


6.1.3.48 dataChecksum()

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

6.1.3.49 dataCompressCm6()

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

6.1.3.50 dataConvert() [1/3]

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

6.1.3.51 dataConvert() [2/3]

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

6.1.3.52 dataConvert() [3/3]

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

6.1.3.53 dataDeCompressCm6()

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

6.1.3.54 duplicateDump()

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

6.1.3.55 fileNameTime()

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

6.1.3.56 fixedString()

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

6.1.3.57 fixedWidthValue()

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

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

6.1.3.58 fromSeedTimeString()

```
static BTimeStamp Bds::fromSeedTimeString (
    BString str ) [static]
```

6.1.3.59 getHexString()

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

6.1.3.60 nullString()

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

6.1.3.61 record_handler()

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

6.1.3.62 removeCR()

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

6.1.3.63 responseSort()

```
static int Bds::responseSort (  
    Response & a,  
    Response & b ) [static]
```

6.1.3.64 roundDigits()

```
static double Bds::roundDigits (  
    double v,  
    int nDigits ) [static]
```

6.1.3.65 seedChannelDataType()

```
BString Bds::seedChannelDataType (  
    BString channel )
```

Returns dataType from channel name based on SEED channel name convention.

6.1.3.66 seedChannelInstrumentCode()

```
char Bds::seedChannelInstrumentCode (
    BString dataType )
```

Returns SEED instrument code from dataType.

6.1.3.67 seedTime()

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

6.1.3.68 seedTimeString()

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

6.1.3.69 stringFormat()

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

6.1.3.70 unitsCode()

```
static BString Bds::unitsCode (
    Response & r ) [static]
```

6.1.4 Variable Documentation

6.1.4.1 apiVersion

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

6.1.4.2 BdsDataFileVersion

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

6.1.4.3 bdsSpecialChannelsList

```
BList<SpecialChannel> Bds::bdsSpecialChannelsList [static]
```

6.1.4.4 ChannelAuxLen

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

Maximum [Channel](#) Aux length.

6.1.4.5 ChannelTypeLen

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

Maximum [Channel](#) type name length.

6.1.4.6 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'
}
```

6.1.4.7 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, 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,
}
```

6.1.4.8 crcInitDone

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

6.1.4.9 crcVec

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

6.1.4.10 dataFormatAll

```
DataFormatAll Bds::dataFormatAll
```

6.1.4.11 NetworkNameLen

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

Maximum [Network](#) name length.

6.1.4.12 node_types

```
const char* Bds::node_types[]
```

Initial value:

```
= {
    "null", "document", "element", "pcdata", "cdata", "comment", "pi", "declaration"
}
```

6.1.4.13 Scale

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

6.1.4.14 seedICodeToDataTypes

```
SeedICodeToDataType Bds::seedICodeToDataTypes[]
```

Initial value:

```
= {
    { 'H', "seismic" },
    { 'L', "seismic" },
    { 'G', "seismic" },
    { 'M', "seismic" },
    { 'N', "seismic" },
    { 'Y', "data" },
    { 'A', "tilt" },
    { 'B', "creep" },
    { 'C', "calibration" },
    { 'D', "pressure" },
    { 'E', "testpoint" },
    { 'F', "magnetometer" },
    { 'I', "humidity" },
    { 'J', "rotation" },
    { 'K', "temperature" },
    { 'O', "waterCurrent" },
    { 'P', "geophone" },
    { 'Q', "voltage" },
    { 'R', "rainfall" },
    { 'S', "linearStrain" },
    { 'T', "tide" },
    { 'U', "bolometer" },
    { 'V', "volumetricStrain" },
    { 'W', "wind" },
    { 'X', "generated" },
    { 'Z', "beam" },
    { 0, 0 }
}
```

6.1.4.15 SourceLen

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

Maximum [Source](#) length.

6.1.4.16 StationNameLen

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

Maximum [Station](#) name length.

Chapter 7

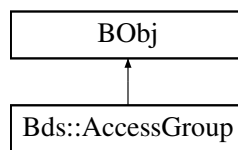
Class Documentation

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:

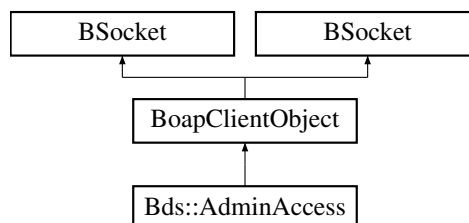
- [BdsD.h](#)
- [BdsD.cc](#)

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** [userGetOptions](#) (**BDict**< **BString** > &items)
Get user options.
- **BError** [userSetOptions](#) (**BDict**< **BString** > &items)
Set user options.
- **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](#), [Channel](#) or both Locations depending on the sel.locationSelect parameter.
- **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** `channelInstrumentGetList` ([Selection](#) sel, **BList**< [ChannelInstrument](#) > &channelInstruments)
Get list of [Channel](#) Instruments.
- **BError** `channelInstrumentUpdate` (**BInt32** append, [ChannelInstrument](#) channelInstrument, **BUInt32** &id)
Add or update a Instrument entry.
- **BError** `channelInstrumentDelete` (**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** [eventGetList](#) ([Selection](#) sel, [BList](#)< [Event](#) > &events)

Get list of [Events](#).
- **BError** [eventUpdate](#) ([BInt32](#) append, [Event](#) event, [BUInt32](#) &id)

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

Delete a [Event](#) entry.
- **BError** [specialChannelGetList](#) ([Selection](#) sel, [BList](#)< [SpecialChannel](#) > &specialChannels)

Get list of [Special Channels](#).
- **BError** [specialChannelUpdate](#) ([BInt32](#) append, [SpecialChannel](#) specialChannel, [BUInt32](#) &id)

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

Delete a [SpecialChannel](#) entry.
- **BError** [metadataGetChannelInfo](#) ([Selection](#) sel, [ChannelInfos](#) &channelInfos)

Return the channel [MetaData](#) in structured form.
- **BError** [metadataGetFormatted](#) ([Selection](#) sel, [BString](#) format, [BArray](#)< [BUInt8](#) > &data)

Return the channel [MetaData](#) in a particular format.
- **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. If num > 0 segment into this number of fixed time segments.
- **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.

- **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` > ¬es)
Get notes on the data file.
- **Error** `dataGetWarnings` (`DataHandle` dataHandle, `BList`< `BString` > &warnings)
Get information on the data file.
- **Error** `dataGetBlock` (`DataHandle` dataHandle, `BUInt32` channel, `BUInt32` segment, `BUInt32` block↵
Number, `DataBlock` & data)
Return a block of data.
- **Error** `dataSeekBlock` (`DataHandle` dataHandle, `BUInt32` channel, `BUInt32` segment, `BTimeStamp` time,
`BUInt32` &blockNumber)
Searches for a data block matching the time given.
- **Error** `dataSetInfo` (`DataHandle` dataHandle, `DataInfo` dataInfo)
Set the info when writing to a file.
- **Error** `dataPutBlock` (`DataHandle` dataHandle, `DataBlock` data)
Send a block of data.
- **Error** `dataClose` (`DataHandle` dataHandle, **Error** 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** `dataRealtimeConfig` (`BInt32` enable, `Selection` sel)
Configures the sending of real-time data blocks.
- **Error** `dataRealtimeGet` (`BUInt32` numBlocks, `BUInt32` &numBlocksAvailable, `BList`< `DataBlockChannel`
> &dataBlocks)
Returns the number of data blocks available and up to numBlocks of these.
- **Error** `changeGroupStart` (`ChangeGroup` changeGroup)
Start a new `ChangeGroup` when making a set of changes to the BDS's database.
- **Error** `changeGroupEnd` ()
End a `ChangeGroup`.
- **Error** `changeGroupGetList` (`ListRange` range, `BList`< `ChangeGroup` > &changeGroups)
Return a list of `ChangeGroups`.
- **Error** `changeGroupDelete` (`BTimeStamp` beforeDate, `BString` type, `BInt32` empty)
Delete a `ChangeGroup`.
- **Error** `changeGetListNumber` (`BUInt32` id, `BUInt32` & number)
Get the number of changes in a `ChangeGroup`.
- **Error** `changeGetList` (`BUInt32` id, `ListRange` range, `BList`< `Change` > &changes)
Get a list of `Changes`.
- **Error** `changeDelete` (`BTimeStamp` beforeDate, `BString` type)
Delete a `Change`.
- **Error** `noteGetList` (`Selection` sel, `BList`< `Note` > ¬es)
Get a list of `Notes`.
- **Error** `noteUpdate` (`BInt32` append, `Note` note, `BUInt32` &id)
Add or update a `Note`.
- **Error** `noteDelete` (`BUInt32` id)
Delete a `Note`.
- **Error** `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** [serverConfigurationGet](#) (**BDict**< **BString** > &items)

Get server configuration parameters.
- **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.
- **BError** [extraCall](#) (**BUInt32** function, **BString** args, **BString** &result)

A special function to add new functions to the API prior to a full API update.

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 unresiticed 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. If num > 0 segment into this number of fixed time segments.

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 Bds::DataFlags .
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 dataRealtimeConfig()

```
BError Bds::AdminAccess::dataRealtimeConfig (
    BInt32 enable,
    Selection sel )
```

Configures the sending of real-time data blocks.

7.2.3.44 dataRealtimeGet()

```
BError Bds::AdminAccess::dataRealtimeGet (
    BUInt32 numBlocks,
    BUInt32 & numBlocksAvailable,
    BList< DataBlockChannel > & dataBlocks )
```

Returns the number of data blocks available and up to numBlocks of these.

7.2.3.45 dataSearch()

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

Search for data matching the given selection parameters.

7.2.3.46 dataSeekBlock()

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

Searches for a data block matching the time given.

7.2.3.47 dataSetInfo()

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

Set the info when writing to a file.

7.2.3.48 digitiserDelete()

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

Delete a [Digitiser](#) entry.

7.2.3.49 digitiserGet()

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

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

7.2.3.50 digitiserGetList()

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

Get list of Digitisers.

7.2.3.51 digitiserUpdate()

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

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

7.2.3.52 eventDelete()

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

Delete a [Event](#) entry.

7.2.3.53 eventGetList()

```
BError Bds::AdminAccess::eventGetList (
    Selection sel,
    BList< Event > & events )
```

Get list of Events.

7.2.3.54 eventUpdate()

```
BError Bds::AdminAccess::eventUpdate (
    BInt32 append,
    Event event,
    BUInt32 & id )
```

Add or update a **Event** entry.

7.2.3.55 extraCall()

```
BError Bds::AdminAccess::extraCall (
    BUInt32 function,
    BString args,
    BString & result )
```

A special function to add new functions to the API prior to a full API update.

7.2.3.56 getSelectionInfo()

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

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

7.2.3.57 getSelections()

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

Get selection list.

7.2.3.58 getVersion()

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

Gets the software version and server name.

7.2.3.59 groupDelete()

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

Delete a group entry.

7.2.3.60 groupGetList()

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

Get list of Groups.

7.2.3.61 groupUpdate()

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

Update or append a group entry.

7.2.3.62 locationDelete()

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

Delete a [Station Location](#) entry.

7.2.3.63 locationGetList()

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

Get list of [Station](#), [Channel](#) or both Locations depending on the sel.locationSelect parameter.

7.2.3.64 locationUpdate()

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

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

7.2.3.65 logAppend()

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

Append a log item.

7.2.3.66 logDelete()

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

Delete a [Log](#) item.

7.2.3.67 logGetList()

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

Get list of log entries.

7.2.3.68 logUpdate()

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

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

7.2.3.69 metadataGetChannelInfo()

```
BError Bds::AdminAccess::metadataGetChannelInfo (
    Selection sel,
    ChannelInfos & channelInfos )
```

Return the channel MetaData in structured form.

7.2.3.70 metadataGetFormatted()

```
BError Bds::AdminAccess::metadataGetFormatted (
    Selection sel,
    BString format,
    BArray< BUInt8 > & data )
```

Return the channel MetaData in a particular format.

7.2.3.71 modeSet()

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

Changes the system mode from Master to slave.

7.2.3.72 modeSnapshotPause()

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

Enables/disables backup synchronisation pause.

7.2.3.73 networkDelete()

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

Delete a [Network](#) entry.

7.2.3.74 networkGetList()

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

Get list of Networks.

7.2.3.75 networkUpdate()

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

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

7.2.3.76 noteDelete()

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

Delete a [Note](#).

7.2.3.77 noteGetList()

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

Get a list of Notes.

7.2.3.78 noteReadDocument()

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

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

7.2.3.79 noteUpdate()

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

Add or update a [Note](#).

7.2.3.80 noteWriteDocument()

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

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

7.2.3.81 responseDelete()

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

Delete a [Response](#) entry.

7.2.3.82 responseGetList()

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

Get list of Responses.

7.2.3.83 responseUpdate()

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

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

7.2.3.84 sensorDelete()

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

Delete a [Sensor](#) entry.

7.2.3.85 sensorGet()

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

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

7.2.3.86 sensorGetList()

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

Get list of Sensors.

7.2.3.87 sensorUpdate()

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

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

7.2.3.88 serverConfigurationGet()

```
BError Bds::AdminAccess::serverConfigurationGet (
    BDict< BString > & items )
```

Get server configuration parameters.

7.2.3.89 setUser()

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

Sets user to given name or email.

7.2.3.90 setUserReal()

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

Sets user back to real user.

7.2.3.91 sourceDelete()

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

Delete a [Source](#) entry.

7.2.3.92 sourceGetList()

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

Get list of Sources.

7.2.3.93 sourcePriorityDelete()

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

7.2.3.94 sourcePriorityGetList()

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

Get list of SourcePriorities.

7.2.3.95 sourcePriorityUpdate()

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

7.2.3.96 sourceUpdate()

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

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

7.2.3.97 specialChannelDelete()

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

Delete a [SpecialChannel](#) entry.

7.2.3.98 specialChannelGetList()

```
BError Bds::AdminAccess::specialChannelGetList (
    Selection sel,
    BList< SpecialChannel > & specialChannels )
```

Get list of Special Channels.

7.2.3.99 specialChannelUpdate()

```
BError Bds::AdminAccess::specialChannelUpdate (
    BInt32 append,
    SpecialChannel specialChannel,
    BUInt32 & id )
```

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

7.2.3.100 sqlQuery()

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

A low level SQL access function.

7.2.3.101 stationDelete()

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

Delete a [Station](#) entry.

7.2.3.102 stationGetList()

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

Get list of Stations.

7.2.3.103 stationUpdate()

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

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

7.2.3.104 statisticsGet()

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

Get a list of system statistics.

7.2.3.105 transactionEnd()

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

Ends a set of transactions.

7.2.3.106 transactionStart()

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

Starts a set of transactions.

7.2.3.107 userDelete()

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

Delete a user entry.

7.2.3.108 userGet()

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

Get user info.

7.2.3.109 userGetFromId()

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

Get user info given user ID.

7.2.3.110 userGetGroups()

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

Get list of groups the user belongs to.

7.2.3.111 userGetList()

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

Get list of Users.

7.2.3.112 userGetOptions()

```
BError Bds::AdminAccess::userGetOptions (
    BDict< BString > & items )
```

Get user options.

7.2.3.113 userSet()

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

Set user info.

7.2.3.114 userSetOptions()

```
BError Bds::AdminAccess::userSetOptions (
    BDict< BString > & items )
```

Set user options.

7.2.3.115 userUpdate()

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

Update or append a user entry.

7.2.3.116 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](#) network= [BString](#)(), [BString](#) station= [BString](#)(), [BString](#) channel= [BString](#)(), [BFloat64](#) arrayOffsetEast=0, [BFloat64](#) arrayOffsetNorth=0)

Public Attributes

- [BString](#) network
The [Network](#) this station belongs to if for a particular network.
- [BString](#) station
The Stations name.
- [BString](#) channel
The Channels name.
- [BFloat64](#) arrayOffsetEast
The Array offset in in an array in an easterly direction in metres.
- [BFloat64](#) arrayOffsetNorth
The Array offset in in an array in a northerly direction in metres.

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 network = BString(),
    BString station = BString(),
    BString channel = BString(),
    BFloat64 arrayOffsetEast = 0,
    BFloat64 arrayOffsetNorth = 0 )
```

7.3.3 Member Data Documentation

7.3.3.1 arrayOffsetEast

```
BFloat64 Bds::ArrayChannel::arrayOffsetEast
```

The Array offset in in an array in an easterly direction in metres.

7.3.3.2 arrayOffsetNorth

```
BFloat64 Bds::ArrayChannel::arrayOffsetNorth
```

The Array offset in in an array in a northerly direction in metres.

7.3.3.3 channel

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

The Channels name.

7.3.3.4 network

```
BString Bds::ArrayChannel::network
```

The [Network](#) this station belongs to if for a particular network.

7.3.3.5 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:

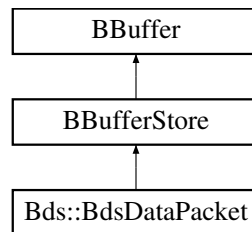
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h

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()

```
BError 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:

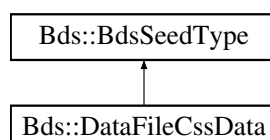
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h](#)

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:

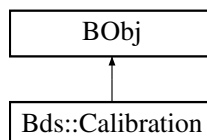
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/[BdsSeedType.h](#)
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/[BdsSeedType.cpp](#)

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(), BString calibrationUnitsDesc= BString(), BFloat64 rawCalibrationFrequency=0, BFloat64 rawCalibrationFactor=0, BString rawCalibrationUnits= BString(), BFloat64 depth=0, BFloat64 waterLevel=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 in Hz.
- **BFloat64** [calibrationFrequency](#)
The post response compensation frequency that the CalibrationFactor value is valid for in Hz.
- **BFloat64** [calibrationFactor](#)
The post response compensation 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 post response compensation measurement SI units such as "m".
- **BString** [calibrationUnitsDesc](#)
A description of the calibrationUnits.
- **BFloat64** [rawCalibrationFrequency](#)
The raw data frequency that the CalibrationFactor value is valid for in Hz.
- **BFloat64** [rawCalibrationFactor](#)
The raw data scaling value to apply to the data to normalise data to the units. This is a measured value at the calibration frequency.
- **BString** [rawCalibrationUnits](#)
The raw data measurement SI units such as "m".
- **BFloat64** [depth](#)
The depth of the sensor below ground level in meters.
- **BFloat64** [waterLevel](#)
Elevation of the water surface in meters for underwater sites, where 0 is sea level.
- **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(),
    BString calibrationUnitsDesc = BString(),
    BFloat64 rawCalibrationFrequency = 0,
    BFloat64 rawCalibrationFactor = 0,
    BString rawCalibrationUnits = BString(),
    BFloat64 depth = 0,
    BFloat64 waterLevel = 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 post response compensation 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 post response compensation frequency that the CalibrationFactor value is valid for in Hz.

7.12.4.3 calibrationUnits

BString Bds::Calibration::calibrationUnits

The post response compensation measurment SI units such as "m".

7.12.4.4 calibrationUnitsDesc

BString Bds::Calibration::calibrationUnitsDesc

A description of the calibrationUnits.

7.12.4.5 channel

BString Bds::Calibration::channel

The channel.

7.12.4.6 depth

BFloat64 Bds::Calibration::depth

The depth of the sensor below ground level in meters.

7.12.4.7 endTime

BTimeStamp Bds::Calibration::endTime

The End Time.

7.12.4.8 horizontalAngle

BFloat64 Bds::Calibration::horizontalAngle

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

7.12.4.9 id

BUInt32 Bds::Calibration::id

The ID.

7.12.4.10 name

BString Bds::Calibration::name

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

7.12.4.11 network

BString Bds::Calibration::network

The Network/Organisation Name.

7.12.4.12 rawCalibrationFactor

BFloat64 Bds::Calibration::rawCalibrationFactor

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

7.12.4.13 rawCalibrationFrequency

BFloat64 Bds::Calibration::rawCalibrationFrequency

The raw data frequency that the CalibrationFactor value is valid for in Hz.

7.12.4.14 rawCalibrationUnits

BString Bds::Calibration::rawCalibrationUnits

The raw data measurment SI units such as "m".

7.12.4.15 samplingFrequency

BFloat64 Bds::Calibration::samplingFrequency

The sample rate used in Hz.

7.12.4.16 source

BString Bds::Calibration::source

The source.

7.12.4.17 startTime

BTimeStamp Bds::Calibration::startTime

The Start Time.

7.12.4.18 station

BString Bds::Calibration::station

The station.

7.12.4.19 verticalAngle

BFloat64 Bds::Calibration::verticalAngle

The Sensors channel placement vertical angle in degrees degrees with zero = vertically up.

7.12.4.20 waterLevel

BFloat64 Bds::Calibration::waterLevel

Elevation of the water surface in meters for underwater sites, where 0 is sea level.

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]
- char [stationName](#) [16]
- char [channelName](#) [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 channelName

char Bds::CdChannel_1v0::channelName[16]

7.13.2.6 compress

BUInt8 Bds::CdChannel_1v0::compress

7.13.2.7 name

char Bds::CdChannel_1v0::name[16]

7.13.2.8 spare0

BUInt8 Bds::CdChannel_1v0::spare0

7.13.2.9 spare1

BUInt8 Bds::CdChannel_1v0::spare1

7.13.2.10 stationName

char Bds::CdChannel_1v0::stationName[16]

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:

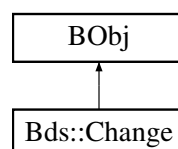
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h](#)

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:

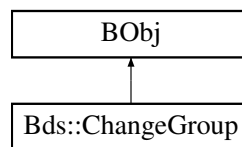
- [BdsD.h](#)
- [BdsD.cc](#)

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:

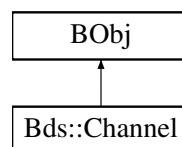
- [BdsD.h](#)
- [BdsD.cc](#)

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= **BTime**↔**Stamp**(), **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** stationLocation=**Location**(), **Channel** channel=**Channel**(), **Location** channelLocation=**Location**(), **BString** source= **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** stationLocation
The [Station](#) location.
- **Channel** channel
The [Channel](#) data.
- **Location** channelLocation
The [Channel](#) location.
- **BString** source
The data source.
- **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 stationLocation = Location(),
    Channel channel = Channel(),
    Location channelLocation = Location(),
    BString source = 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 channelLocation

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

The [Channel](#) location.

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 responses

`BList<Response> Bds::ChannelInfo::responses`

The list of frequency responses.

7.21.3.7 sensor

`Sensor Bds::ChannelInfo::sensor`

The [Sensor](#) in use.

7.21.3.8 source

`BString Bds::ChannelInfo::source`

The data source.

7.21.3.9 startTime

`BTimeStamp Bds::ChannelInfo::startTime`

The Start Time.

7.21.3.10 station

`Station Bds::ChannelInfo::station`

The [Station](#) info.

7.21.3.11 stationLocation

`Location` Bds::ChannelInfo::stationLocation

The `Station` location.

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` (`Station` array=`Station`(), `BArray`< `BArray`< `ChannelInfo` > > `channels`=`BArray`< `BArray`< `ChannelInfo` > >())

Public Attributes

- `Station` array
The array if all channels are from an array.
- `BArray`< `BArray`< `ChannelInfo` > > `channels`
The per channel per segment Metadata.

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 consistant metadata over a particular time period. The BdsServer will create a `ChannelInfo` object on each cahnge 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 (
    Station array = Station(),
    BArray< BArray< ChannelInfo > > channels = BArray< BArray<ChannelInfo > >()
)
```

7.22.3 Member Data Documentation

7.22.3.1 array

`Station` `Bds::ChannelInfos::array`

The array if all channels are from an array.

7.22.3.2 channels

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

The per channel per segment Metadata.

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

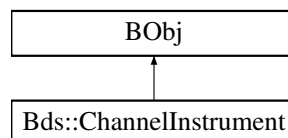
- [BdsD.h](#)
- [BdsD.cc](#)

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::CompressSteiml::clear ( )
```

7.26.3.2 setByteOrder()

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

7.26.3.3 unCompress()

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

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

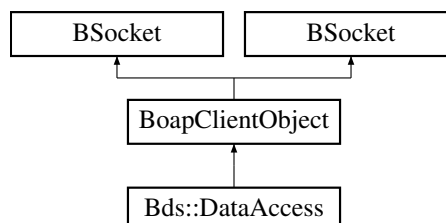
- [/src/blacknest/bds/bds/bdsDataLib/BdsCompress.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp](#)

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 userGetOptions](#) ([BDict](#)< [BString](#) > &items)
Get user options.
- [BError userSetOptions](#) ([BDict](#)< [BString](#) > &items)
Set user options.
- [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 channelInstrumentGetList](#) ([Selection](#) sel, [BList](#)< [ChannelInstrument](#) > &channelInstruments)
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.
- **BError** `locationGetList` (`Selection` sel, `BList`< `Location` > &locations)
Get list of `Station`, `Channel` or both `Locations` depending on the sel.locationSelect parameter.
- **BError** `eventGetList` (`Selection` sel, `BList`< `Event` > &events)
Get list of Events.
- **BError** `specialChannelGetList` (`Selection` sel, `BList`< `SpecialChannel` > &specialChannels)
Get list of Special Channels.
- **BError** `metadataGetChannelInfo` (`Selection` sel, `ChannelInfos` &channelInfos)
Return the channel MetaData in structured form.
- **BError** `metadataGetFormatted` (`Selection` sel, `BString` format, `BArray`< `BUInt8` > & data)
Return the channel MetaData in a particular format.
- **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. If num > 0 segment ito this number of fixed time segments.
- **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` > ¬es)
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** `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** `dataRealtimeConfig` (`BInt32` enable, `Selection` sel)
Configures the sending of real-time data blocks.
- **BError** `dataRealtimeGet` (`BUInt32` numBlocks, `BUInt32` &numBlocksAvailable, `BList`< `DataBlockChannel`
> &dataBlocks)
Returns the number of data blocks available and up to numBlocks of these.

- **BError** [noteGetList](#) ([Selection](#) sel, **BList**< [Note](#) > ¬es)
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)
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** [serverConfigurationGet](#) (**BDict**< **BString** > &items)
Get server configuration parameters.
- **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. If num > 0 segment ito this number of fixed time segments.

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 Bds::DataFlags .
out	<i>dataHandle</i>	The handle for the open data set

7.27.3.20 dataRealtimeConfig()

```
BError Bds::DataAccess::dataRealtimeConfig (
    BInt32 enable,
    Selection sel )
```

Configures the sending of real-time data blocks.

7.27.3.21 dataRealtimeGet()

```
BError Bds::DataAccess::dataRealtimeGet (
    BUInt32 numBlocks,
    BUInt32 & numBlocksAvailable,
    BList< DataBlockChannel > & dataBlocks )
```

Returns the number of data blocks available and up to numBlocks of these.

7.27.3.22 dataSearch()

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

Search for data matching the given selection parameters.

7.27.3.23 dataSeekBlock()

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

Searches for a data block matching the time given.

7.27.3.24 digitiserGet()

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

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

7.27.3.25 digitiserGetList()

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

Get list of Digitisers.

7.27.3.26 eventGetList()

```
BError Bds::DataAccess::eventGetList (
    Selection sel,
    BList< Event > & events )
```

Get list of Events.

7.27.3.27 getSelectionInfo()

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

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

7.27.3.28 getSelections()

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

Get selection list.

7.27.3.29 getVersion()

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

Gets the software version and server name.

7.27.3.30 groupGetList()

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

Get list of Groups.

7.27.3.31 locationGetList()

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

Get list of [Station](#), [Channel](#) or both Locations depending on the sel.locationSelect parameter.

7.27.3.32 logAppend()

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

Append a log item.

7.27.3.33 logUpdate()

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

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

7.27.3.34 metadataGetChannelInfo()

```
BError Bds::DataAccess::metadataGetChannelInfo (
    Selection sel,
    ChannelInfos & channelInfos )
```

Return the channel MetaData in structured form.

7.27.3.35 metadataGetFormatted()

```
BError Bds::DataAccess::metadataGetFormatted (
    Selection sel,
    BString format,
    BArray< BUInt8 > & data )
```

Return the channel MetaData in a particular format.

7.27.3.36 modeSet()

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

Changes the system mode from Master to slave.

7.27.3.37 modeSnapshotPause()

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

Enables/disables backup synchronisation pause.

7.27.3.38 networkGetList()

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

Get list of Networks.

7.27.3.39 noteGetList()

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

Return a list of Notes.

7.27.3.40 noteReadDocument()

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

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

7.27.3.41 noteUpdate()

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

Add or update a [Note](#).

7.27.3.42 noteWriteDocument()

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

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

7.27.3.43 responseGetList()

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

Get list of Responses.

7.27.3.44 sensorGet()

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

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

7.27.3.45 sensorGetList()

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

Get list of Sensors.

7.27.3.46 serverConfigurationGet()

```
BError Bds::DataAccess::serverConfigurationGet (
    BDict< BString > & items )
```

Get server configuration parameters.

7.27.3.47 setUser()

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

Sets user to given name or email.

7.27.3.48 setUserReal()

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

Sets user back to real user.

7.27.3.49 sourceGetList()

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

Get list of Sources.

7.27.3.50 sourcePriorityGetList()

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

Get list of SourcePriorities.

7.27.3.51 specialChannelGetList()

```
BError Bds::DataAccess::specialChannelGetList (
    Selection sel,
    BList< SpecialChannel > & specialChannels )
```

Get list of Special Channels.

7.27.3.52 stationGetList()

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

Get list of Stations.

7.27.3.53 statisticsGet()

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

Get a list of system statistics.

7.27.3.54 userGet()

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

Get user info.

7.27.3.55 userGetFromId()

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

Get user info given user ID.

7.27.3.56 userGetGroups()

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

Get list of groups the user belongs to.

7.27.3.57 userGetOptions()

```
BError Bds::DataAccess::userGetOptions (
    BDict< BString > & items )
```

Get user options.

7.27.3.58 userSet()

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

Set user info.

7.27.3.59 userSetOptions()

```
BEError Bds::DataAccess::userSetOptions (
    BDict< BString > & items )
```

Set user options.

7.27.3.60 validateUser()

```
BEError 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:

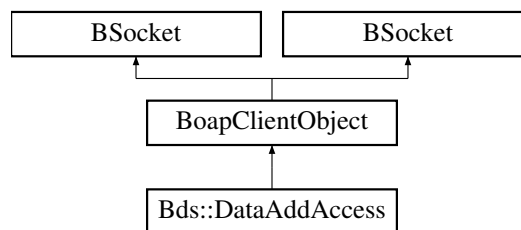
- [BdsC.h](#)
- [BdsC.cc](#)
- [BdsLib.dox](#)

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 userGetOptions** (**BDict**< **BString** > &items)
Get user options.
- **BError userSetOptions** (**BDict**< **BString** > &items)
Set user options.
- **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 channelInstrumentGetList** (**Selection** sel, **BList**< **ChannelInstrument** > &channelInstruments)
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`, `Channel` or both `Locations` depending on the sel.locationSelect parameter.
- **BError** `eventGetList` (`Selection` sel, `BList`< `Event` > &events)
Get list of Events.
- **BError** `eventUpdate` (`BInt32` append, `Event` event, `BUInt32` &id)
Add or update a `Event` entry.
- **BError** `eventDelete` (`BUInt32` id)
Delete a `Event` entry.
- **BError** `specialChannelGetList` (`Selection` sel, `BList`< `SpecialChannel` > &specialChannels)
Get list of Special Channels.
- **BError** `metadataGetChannelInfo` (`Selection` sel, `ChannelInfos` &channelInfos)
Return the channel MetaData in structured form.
- **BError** `metadataGetFormatted` (`Selection` sel, `BString` format, `BArray`< `BUInt8` > &data)
Return the channel MetaData in a particular format.
- **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. If num > 0 segment ito this number of fixed time segments.
- **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` > ¬es)
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.*
- **Error** `dataFormattedGetLength` (`DataHandle` dataHandle, `BUInt64` & length)
 - The total length in bytes of the formatted data.*
- **Error** `dataRealtimeConfig` (`BInt32` enable, `Selection` sel)
 - Configures the sending of real-time data blocks.*
- **Error** `dataRealtimeGet` (`BUInt32` numBlocks, `BUInt32` &numBlocksAvailable, `BList`< `DataBlockChannel` > &dataBlocks)
 - Returns the number of data blocks available and up to numBlocks of these.*
- **Error** `noteGetList` (`Selection` sel, `BList`< `Note` > ¬es)
 - 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)
 - Append a log item ///< 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.*
- **Error** `databaseBackup` (`BString` &ref)
 - Backup the database.*
- **Error** `statisticsGet` (`BDict`< `BString` > &info)
 - Get a list of system statistics.*
- **Error** `serverConfigurationGet` (`BDict`< `BString` > &items)
 - Get server configuration parameters.*
- **Error** `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. If num > 0 segment into this number of fixed time segments.

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 Bds::DataFlags .
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 dataRealtimeConfig()

```
BError Bds::DataAddAccess::dataRealtimeConfig (
    BInt32 enable,
    Selection sel )
```

Configures the sending of real-time data blocks.

7.28.3.22 dataRealtimeGet()

```
BError Bds::DataAddAccess::dataRealtimeGet (
    BUInt32 numBlocks,
    BUInt32 & numBlocksAvailable,
    BList< DataBlockChannel > & dataBlocks )
```

Returns the number of data blocks available and up to numBlocks of these.

7.28.3.23 dataSearch()

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

Search for data matching the given selection parameters.

7.28.3.24 dataSeekBlock()

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

Searches for a data block matching the time given.

7.28.3.25 dataSetInfo()

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

Set the info when writing to a file.

7.28.3.26 digitiserGet()

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

Get a *Digitiser* object given its ID.

7.28.3.27 digitiserGetList()

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

Get list of Channels.

7.28.3.28 eventDelete()

```
BError Bds::DataAddAccess::eventDelete (
    BUInt32 id )
```

Delete a *Event* entry.

7.28.3.29 eventGetList()

```
BError Bds::DataAddAccess::eventGetList (
    Selection sel,
    BList< Event > & events )
```

Get list of Events.

7.28.3.30 eventUpdate()

```
BError Bds::DataAddAccess::eventUpdate (
    BInt32 append,
    Event event,
    BUInt32 & id )
```

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

7.28.3.31 getSelectionInfo()

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

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

7.28.3.32 getSelections()

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

Get selection list.

7.28.3.33 getVersion()

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

Gets the software version and server name.

7.28.3.34 groupGetList()

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

Get list of Groups.

7.28.3.35 locationGetList()

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

Get list of [Station](#), [Channel](#) or both Locations depending on the sel.locationSelect parameter.

7.28.3.36 logAppend()

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

Append a log item.

7.28.3.37 logUpdate()

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

Append a log item ///< Add or update a [Log](#) item.

7.28.3.38 metadataGetChannelInfo()

```
BError Bds::DataAddAccess::metadataGetChannelInfo (
    Selection sel,
    ChannelInfos & channelInfos )
```

Return the channel MetaData in structured form.

7.28.3.39 metadataGetFormatted()

```
BError Bds::DataAddAccess::metadataGetFormatted (
    Selection sel,
    BString format,
    BArray< BUInt8 > & data )
```

Return the channel MetaData in a particular format.

7.28.3.40 modeSet()

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

Changes the system mode from Master to slave.

7.28.3.41 modeSnapshotPause()

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

Enables/disables backup synchronisation pause.

7.28.3.42 networkGetList()

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

Get list of Networks.

7.28.3.43 noteGetList()

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

Return a list of Notes.

7.28.3.44 noteReadDocument()

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

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

7.28.3.45 noteUpdate()

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

Add or update a [Note](#).

7.28.3.46 noteWriteDocument()

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

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

7.28.3.47 responseGetList()

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

Get list of Responses.

7.28.3.48 sensorGet()

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

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

7.28.3.49 sensorGetList()

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

Get list of Sensors.

7.28.3.50 serverConfigurationGet()

```
BError Bds::DataAddAccess::serverConfigurationGet (
    BDict< BString > & items )
```

Get server configuration parameters.

7.28.3.51 setUser()

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

Sets user to given name or email.

7.28.3.52 setUserReal()

```
BError Bds::DataAddAccess::setUserReal ( )
```

Sets user back to real user.

7.28.3.53 sourceGetList()

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

Get list of Sources.

7.28.3.54 sourcePriorityGetList()

```
BError Bds::DataAddAccess::sourcePriorityGetList (
    BList< SourcePriority > & sourcePriorities )
```

Get list of SourcePriorities.

7.28.3.55 specialChannelGetList()

```
BError Bds::DataAddAccess::specialChannelGetList (
    Selection sel,
    BList< SpecialChannel > & specialChannels )
```

Get list of Special Channels.

7.28.3.56 stationGetList()

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

Get list of Stations.

7.28.3.57 statisticsGet()

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

Get a list of system statistics.

7.28.3.58 userGet()

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

Get user info.

7.28.3.59 userGetFromId()

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

Get user info given user ID.

7.28.3.60 userGetGroups()

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

Get list of groups the user belongs to.

7.28.3.61 userGetOptions()

```
BError Bds::DataAddAccess::userGetOptions (
    BDict< BString > & items )
```

Get user options.

7.28.3.62 userSet()

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

Set user info.

7.28.3.63 userSetOptions()

```
BError Bds::DataAddAccess::userSetOptions (
    BDict< BString > & items )
```

Set user options.

7.28.3.64 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 cahnnel). 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::DataBlockChannel Class Reference

This class provides the actual seismic data values contained within a single data block along with the network↔:station:channel:source information.

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

Public Member Functions

- [DataBlockChannel](#) ([BString](#) network= [BString](#)(), [BString](#) station= [BString](#)(), [BString](#) channel= [BString](#)(), [BString](#) source= [BString](#)())

Public Attributes

- [BString](#) network
The [Network](#) Name or names.
- [BString](#) station
The [Station](#) name or Stations.
- [BString](#) channel
The Channels name or Channels.
- [BString](#) source
The Data [Source](#) or Sources.

7.32.1 Detailed Description

This class provides the actual seismic data values contained within a single data block along with the network↔:station:channel:source information.

7.32.2 Constructor & Destructor Documentation

7.32.2.1 DataBlockChannel()

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

7.32.3 Member Data Documentation

7.32.3.1 channel

```
BString Bds::DataBlockChannel::channel
```

The Channels name or Channels.

7.32.3.2 network

```
BString Bds::DataBlockChannel::network
```

The [Network](#) Name or names.

7.32.3.3 source

```
BString Bds::DataBlockChannel::source
```

The Data [Source](#) or Sources.

7.32.3.4 station

```
BString Bds::DataBlockChannel::station
```

The [Station](#) name or Stations.

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

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

7.33 Bds::DataBlockPos Class Reference

This defines the position of a data block in a file. It is used by the BDS data converters 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)
- [operator<](#) (const [DataBlockPos](#) &b) const

Public Attributes

- [BTimeStamp](#) [startTime](#)
- [BTimeStamp](#) [endTime](#)
- [BUInt64](#) [position](#)
- [BUInt](#) [order](#)
- [int](#) [ref](#)
- [BUInt](#) [numSamples](#)

7.33.1 Detailed Description

This defines the position of a data block in a file. It is used by the BDS data converters to order blocks by time.

7.33.2 Constructor & Destructor Documentation

7.33.2.1 DataBlockPos()

```
Bds::DataBlockPos::DataBlockPos (
    BTimeStamp startTime = 0,
    BTimeStamp endTime = 0,
    BUInt64 position = 0,
    BUInt order = 0,
    int ref = 0,
    BUInt numSamples = 0 ) [inline]
```

7.33.3 Member Function Documentation

7.33.3.1 operator<()

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

7.33.4 Member Data Documentation

7.33.4.1 endTime

```
BTimeStamp Bds::DataBlockPos::endTime
```

7.33.4.2 numSamples

```
BUInt Bds::DataBlockPos::numSamples
```

7.33.4.3 order

```
BUInt Bds::DataBlockPos::order
```

7.33.4.4 position

```
BUInt64 Bds::DataBlockPos::position
```

7.33.4.5 ref

```
int Bds::DataBlockPos::ref
```

7.33.4.6 startTime

```
BTimeStamp Bds::DataBlockPos::startTime
```

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

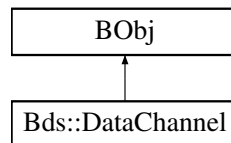
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h](#)

7.34 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.

7.34.1 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.

7.34.2 Constructor & Destructor Documentation

7.34.2.1 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 >() )
```

7.34.3 Member Function Documentation

7.34.3.1 getMember()

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

Reimplemented from **BObj**.

7.34.3.2 getMembers()

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

Reimplemented from **BObj**.

7.34.3.3 getType()

```
BString Bds::DataChannel::getType ( )
```

7.34.3.4 setMember()

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

Reimplemented from **BObj**.

7.34.3.5 setMembers()

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

Reimplemented from **BObj**.

7.34.4 Member Data Documentation

7.34.4.1 channel

BString Bds::DataChannel::channel

The Channels name.

7.34.4.2 dataFileChannel

BUInt32 Bds::DataChannel::dataFileChannel

The Data File [Channel](#) number. The channel number within the data file. (1, 2, 3 ...)

7.34.4.3 dataFileId

BUInt32 Bds::DataChannel::dataFileId

The Data File Id. This links to the particular [DataFileInfo](#) where the data is stored.

7.34.4.4 endTime

BTimeStamp Bds::DataChannel::endTime

The End Time.

7.34.4.5 id

BUInt32 Bds::DataChannel::id

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

7.34.4.6 importFilename

BString Bds::DataChannel::importFilename

The original data file name.

7.34.4.7 importFormat

BString Bds::DataChannel::importFormat

The original data format.

7.34.4.8 importStartTime

BTimeStamp Bds::DataChannel::importStartTime

The original import files start time.

7.34.4.9 info

BDict< **BString** > Bds::DataChannel::info

Extra info on the channel.

7.34.4.10 network

BString Bds::DataChannel::network

The [Network](#) Name.

7.34.4.11 numBlocks

BUInt32 Bds::DataChannel::numBlocks

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

7.34.4.12 numSamples

BUInt64 Bds::DataChannel::numSamples

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

7.34.4.13 sampleFormat

BUInt32 Bds::DataChannel::sampleFormat

The data sample format.

7.34.4.14 sampleRate

BFloat64 Bds::DataChannel::sampleRate

The data's sample rate.

7.34.4.15 source

BString Bds::DataChannel::source

The Data [Source](#).

7.34.4.16 startTime

BTimeStamp Bds::DataChannel::startTime

The Start Time.

7.34.4.17 station

BString Bds::DataChannel::station

The [Station](#) name.

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

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

7.35 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)

7.35.1 Detailed Description

Not sure if this is used or what it does.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 DataCollate()

```
Bds::DataCollate::DataCollate ( )
```

7.35.2.2 ~DataCollate()

```
Bds::DataCollate::~~DataCollate ( )
```

7.35.3 Member Function Documentation

7.35.3.1 addSource()

```
BError Bds::DataCollate::addSource (
    DataFile & dataFile,
    BUInt channel )
```

7.35.3.2 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

7.36 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

- **BInt32** [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.

7.36.1 Detailed Description

This stores a data error. It includes and error number and a string as well as information on what seismic channel it is for.

7.36.2 Constructor & Destructor Documentation

7.36.2.1 DataError() [1/2]

```
Bds::DataError::DataError ( )
```

7.36.2.2 DataError() [2/2]

```
Bds::DataError::DataError (
    int errorNumber,
    BString title,
    BString filename,
    BTimeStamp startTime,
    BTimeStamp endTime,
    DataInfo & dataInfo,
    BUInt channel,
    BString description,
    BString user = "" )
```

7.36.3 Member Function Documentation

7.36.3.1 `getErrorNumber()`

```
int Bds::DataError::getErrorNumber ( ) const
```

Get The error number.

7.36.3.2 `getString()`

```
BString Bds::DataError::getString ( ) const
```

Get error message.

7.36.3.3 `getTitle()`

```
BString Bds::DataError::getTitle ( ) const
```

Get the title.

7.36.3.4 `mergeDataInfo()`

```
void Bds::DataError::mergeDataInfo (
    const DataInfo & dataInfo,
    BUInt channel )
```

7.36.3.5 `num()`

```
int Bds::DataError::num ( ) const
```

Get The error number.

7.36.3.6 operator int()

```
Bds::DataError::operator int ( ) const
```

Return error number.

7.36.3.7 set()

```
DataError & Bds::DataError::set (
    int errorNumber,
    BString title,
    BString importFilename,
    BTimeStamp startTime,
    BTimeStamp endTime,
    DataInfo & dataInfo,
    BUInt channel,
    BString description,
    BString user = "" )
```

7.36.3.8 setString()

```
BError Bds::DataError::setString (
    BString str )
```

Set from string.

7.36.3.9 setStringUser()

```
BError Bds::DataError::setStringUser (
    BString str,
    BString user )
```

Set from string given by user on command line.

7.36.3.10 str()

```
const char * Bds::DataError::str ( ) const
```

Return a char* string.

7.36.4 Member Data Documentation

7.36.4.1 ochannel

BString Bds::DataError::ochannel

The channel name.

7.36.4.2 odescription

BString Bds::DataError::odescription

The description.

7.36.4.3 oendTime

BTimeStamp Bds::DataError::oendTime

The end Time.

7.36.4.4 oerrorNumber

BInt32 Bds::DataError::oerrorNumber

Error number.

7.36.4.5 ofilename

BString Bds::DataError::ofilename

The import filename;.

7.36.4.6 onetwork

BString Bds::DataError::onetwork

The network Name.

7.36.4.7 osource

BString Bds::DataError::osource

The data [Source](#).

7.36.4.8 ostartTime

BTimeStamp Bds::DataError::ostartTime

The start Time.

7.36.4.9 ostation

BString Bds::DataError::ostation

The station/array name.

7.36.4.10 otitle

BString Bds::DataError::otitle

The title.

7.36.4.11 ouser

BString Bds::DataError::ouser

The user.

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

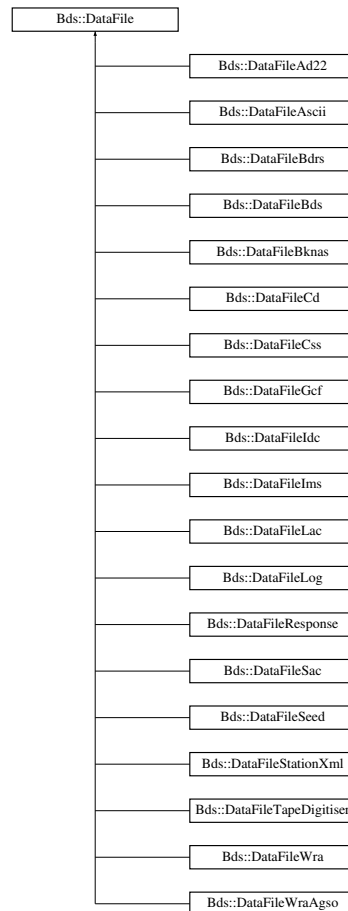
- [BdsLib.h](#)
- [BdsLib.cpp](#)

7.37 Bds::DataFile Class Reference

This class defines the interface for generic data file access that all of the BDS data converters 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 ,
[ReadOptionFixCorruptions](#) = 0x04 ,
[ReadOptionReorder](#) = 0x08 , [ReadOptionDeleteDuplicates](#) = 0x10 , [ReadOptionInfoExtra](#) = 0x20 ,
[ReadOptionIgnoreSamplerate](#) = 0x40 ,
[ReadOptionPrintBlocks](#) = 0x80 , [ReadOptionFixSampleRate](#) = 0x100 }

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 [BString](#) [getFixesInfo](#) ()
Get readable list of fixes that can be applied to faulty data files.
- 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, [BUInt32](#) options, **BList**< [DataError](#) > &errors)
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.
- [BUInt64](#) [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 [DataFormats](#) [getFormats](#) ()
Get list of supported formats.

Protected Attributes

- **BString** [ofileName](#)
- **BString** [omode](#)
- **BTimeStamp** [ofileNameTime](#)
- **BFile** [ofile](#)
- **BString** [ofomat](#)

7.37.1 Detailed Description

This class defines the interface for generic data file access that all of the BDS data converters share.

7.37.2 Member Enumeration Documentation

7.37.2.1 DataOrder

```
enum Bds::DataFile::DataOrder
```

Enumerator

DataOrderUnknown	
DataOrderAll	
DataOrderSample	
DataOrderChannel	

7.37.2.2 Features

```
enum Bds::DataFile::Features
```

Enumerator

FeatureNone	
FeatureCanWrite	
FeatureCanRead	

7.37.2.3 ReadOptionsList

```
enum Bds::DataFile::ReadOptionsList
```

Enumerator

ReadOptionNone	
ReadOptionValidate	
ReadOptionFileNameProcess	
ReadOptionFixCorruptions	
ReadOptionReorder	
ReadOptionDeleteDuplicates	
ReadOptionInfoExtra	
ReadOptionIgnoreSamplerate	
ReadOptionPrintBlocks	
ReadOptionFixSampleRate	

7.37.2.4 WriteOptionsList

```
enum Bds::DataFile::WriteOptionsList
```

Enumerator

WriteOptionNone	
WriteOptionSensorData	
WriteOptionNoMetadata	

7.37.3 Constructor & Destructor Documentation

7.37.3.1 DataFile()

```
Bds::DataFile::DataFile ( )
```

7.37.3.2 ~DataFile()

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

7.37.4 Member Function Documentation

7.37.4.1 close()

```
BError Bds::DataFile::close ( ) [virtual]
```

Close the file.

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

7.37.4.2 dataErrorFixup()

```
void Bds::DataFile::dataErrorFixup (
    const DataInfo & dataInfo,
    BList< DataError > & errors )
```

Fixup data errors, mainly start/end times to be within data.

7.37.4.3 duplicateCheck()

```
int Bds::DataFile::duplicateCheck (
    const DataBlock & data1,
    const DataBlock & data2,
    BUInt channel = 0 )
```

Check if blocks are duplicates.

7.37.4.4 end()

```
BError Bds::DataFile::end ( ) [virtual]
```

End write segment.

Reimplemented in [Bds::DataFileSeed](#), [Bds::DataFileLog](#), [Bds::DataFileIms](#), and [Bds::DataFileAscii](#).

7.37.4.5 fileNameProcess()

```
BError Bds::DataFile::fileNameProcess ( ) [virtual]
```

Parse the file name for a date/time.

7.37.4.6 flush()

BError Bds::DataFile::flush () [virtual]

Flush data to disk.

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

7.37.4.7 getDataOrder()

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

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

Reimplemented in [Bds::DataFileSeed](#), [Bds::DataFileWraAgso](#), [Bds::DataFileWra](#), [Bds::DataFileLog](#), [Bds::DataFileLac](#), [Bds::DataFileImms](#), [Bds::DataFileGcf](#), [Bds::DataFileCss](#), [Bds::DataFileCd](#), [Bds::DataFileBds](#), [Bds::DataFileBdrs](#), [Bds::DataFileAscii](#), and [Bds::DataFileAd22](#).

7.37.4.8 getFeatures()

int Bds::DataFile::getFeatures () [virtual]

Get bitmask of supported features.

Reimplemented in [Bds::DataFileSeed](#), [Bds::DataFileWraAgso](#), [Bds::DataFileWra](#), [Bds::DataFileStationXml](#), [Bds::DataFileSac](#), [Bds::DataFileResponse](#), [Bds::DataFileLog](#), [Bds::DataFileLac](#), [Bds::DataFileImms](#), [Bds::DataFileIdc](#), [Bds::DataFileGcf](#), [Bds::DataFileCss](#), [Bds::DataFileCd](#), [Bds::DataFileBdrs](#), [Bds::DataFileAscii](#), and [Bds::DataFileAd22](#).

7.37.4.9 getFileName()

BString Bds::DataFile::getFileName () [virtual]

Return the file name.

7.37.4.10 getFilePosition()

BUInt64 Bds::DataFile::getFilePosition ()

7.37.4.11 getFixesInfo()

```
BString Bds::DataFile::getFixesInfo ( ) [virtual]
```

Get readable list of fixes that can be applied to faulty data files.

Reimplemented in [Bds::DataFileSeed](#), [Bds::DataFileWra](#), [Bds::DataFileLac](#), [Bds::DataFileGcf](#), [Bds::DataFileCd](#), [Bds::DataFileBdrs](#), and [Bds::DataFileAd22](#).

7.37.4.12 getFormat()

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

Get sub-format.

7.37.4.13 getFormats()

```
DataFormats Bds::DataFile::getFormats ( ) [static]
```

Get list of supported formats.

7.37.4.14 getInfo()

```
BError Bds::DataFile::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

Reimplemented in [Bds::DataFileSeed](#), [Bds::DataFileWraAgso](#), [Bds::DataFileWra](#), [Bds::DataFileTapeDigitiser](#), [Bds::DataFileLog](#), [Bds::DataFileLac](#), [Bds::DataFileGcf](#), [Bds::DataFileCss](#), [Bds::DataFileCd](#), [Bds::DataFileBds](#), [Bds::DataFileBdrs](#), and [Bds::DataFileAd22](#).

7.37.4.15 getMetaData()

```
BError Bds::DataFile::getMetaData (
    ChannelInfos & channelInfos,
    BUInt32 options,
    BList< DataError > & errors ) [virtual]
```

Return all known MetaData in the file.

Reimplemented in [Bds::DataFileSeed](#), [Bds::DataFileStationXml](#), [Bds::DataFileSac](#), [Bds::DataFileResponse](#), [Bds::DataFileIms](#), and [Bds::DataFileIdc](#).

7.37.4.16 init()

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

Initialise.

7.37.4.17 open()

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

Open the file for read or write.

Reimplemented in [Bds::DataFileTapeDigitiser](#), [Bds::DataFileLog](#), [Bds::DataFileIm](#)s, [Bds::DataFileBknas](#), [Bds::DataFileBds](#), and [Bds::DataFileAscii](#).

7.37.4.18 readData()

```
BError Bds::DataFile::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

Reimplemented in [Bds::DataFileBds](#), [Bds::DataFileSeed](#), [Bds::DataFileWraAgso](#), [Bds::DataFileWra](#), [Bds::DataFileTapeDigitiser](#), [Bds::DataFileLog](#), [Bds::DataFileLac](#), [Bds::DataFileGcf](#), [Bds::DataFileCss](#), [Bds::DataFileCd](#), [Bds::DataFileBdrs](#), and [Bds::DataFileAd22](#).

7.37.4.19 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](#).

7.37.4.20 setFormat()

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

Set the sub-format.

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

7.37.4.21 setInfo()

```
BError Bds::DataFile::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options = WriteOptionNone ) [virtual]
```

Set information on data for write.

Reimplemented in [Bds::DataFileSeed](#), [Bds::DataFileLog](#), [Bds::DataFileBknas](#), [Bds::DataFileBds](#), [Bds::DataFileAscii](#), [Bds::DataFileSac](#), [Bds::DataFileIms](#), [Bds::DataFileStationXml](#), [Bds::DataFileResponse](#), and [Bds::DataFileIdc](#).

7.37.4.22 start()

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

Start writing next segment of data.

Reimplemented in [Bds::DataFileSeed](#), [Bds::DataFileLog](#), [Bds::DataFileIms](#), and [Bds::DataFileAscii](#).

7.37.4.23 timeCompare()

```
BInt64 Bds::DataFile::timeCompare (
    BTimeStamp t1,
    BTimeStamp t2,
    BUInt diff )
```

Compare timestamps with a margin.

7.37.4.24 writeData()

```
BError Bds::DataFile::writeData (
    const DataBlock & data ) [virtual]
```

Write a block of data.

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

7.37.5 Member Data Documentation

7.37.5.1 ofile

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

7.37.5.2 ofileName

```
BString Bds::DataFile::ofileName [protected]
```

7.37.5.3 ofileNameTime

```
BTimeStamp Bds::DataFile::ofileNameTime [protected]
```

7.37.5.4 oformat

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

7.37.5.5 omode

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

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

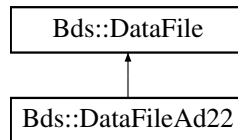
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp](#)

7.38 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.
- **BString** [getFixesInfo](#) ()
Get readable list of fixes that can be applied to faulty data files.
- **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 [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.38.1 Detailed Description

Data file convertor for AD22 format files.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 DataFileAd22()

```
Bds::DataFileAd22::DataFileAd22 ( )
```

7.38.3 Member Function Documentation

7.38.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileAd22::getDataOrder ( ) [virtual]
```

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

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

7.38.3.2 getFeatures()

```
int Bds::DataFileAd22::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.38.3.3 getFixesInfo()

```
BString Bds::DataFileAd22::getFixesInfo ( ) [virtual]
```

Get readable list of fixes that can be applied to faulty data files.

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

7.38.3.4 getFormats()

```
DataFormats Bds::DataFileAd22::getFormats ( ) [static]
```

7.38.3.5 getInfo()

```
BError Bds::DataFileAd22::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.38.3.6 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:

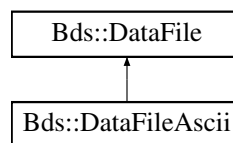
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp](#)

7.39 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 [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.39.1 Detailed Description

Data file convertor for ASCII format files.

7.39.2 Constructor & Destructor Documentation

7.39.2.1 DataFileAscii()

```
Bds::DataFileAscii::DataFileAscii ( )
```

7.39.3 Member Function Documentation

7.39.3.1 end()

```
BEError Bds::DataFileAscii::end ( ) [virtual]
```

End write segment.

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

7.39.3.2 getDataOrder()

```
DataFile::DataOrder Bds::DataFileAscii::getDataOrder ( ) [virtual]
```

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

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

7.39.3.3 getFeatures()

```
int Bds::DataFileAscii::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.39.3.4 getFormats()

```
DataFormats Bds::DataFileAscii::getFormats ( ) [static]
```

7.39.3.5 open()

```
BError Bds::DataFileAscii::open (
    BString fileName,
    BString mode ) [virtual]
```

Open the file for read or write.

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

7.39.3.6 setFormat()

```
BError Bds::DataFileAscii::setFormat (
    BString format ) [virtual]
```

Set the sub-format.

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

7.39.3.7 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](#).

7.39.3.8 start()

```
BError Bds::DataFileAscii::start (
    BUInt channel,
    BUInt segment ) [virtual]
```

Start writing next segment of data.

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

7.39.3.9 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:

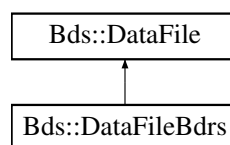
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp](#)

7.40 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.
- **BString** [getFixesInfo](#) ()
Get readable list of fixes that can be applied to faulty data files.
- **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 [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.40.1 Detailed Description

Data file convertor for BDRS format files.

7.40.2 Constructor & Destructor Documentation

7.40.2.1 DataFileBdrs()

```
Bds::DataFileBdrs::DataFileBdrs ( )
```

7.40.3 Member Function Documentation

7.40.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileBdrs::getDataOrder ( ) [virtual]
```

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

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

7.40.3.2 getFeatures()

```
int Bds::DataFileBdrs::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.40.3.3 getFixesInfo()

```
BString Bds::DataFileBdrs::getFixesInfo ( ) [virtual]
```

Get readable list of fixes that can be applied to faulty data files.

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

7.40.3.4 getFormats()

```
DataFormats Bds::DataFileBdrs::getFormats ( ) [static]
```

7.40.3.5 getInfo()

```
BError Bds::DataFileBdrs::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.40.3.6 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:

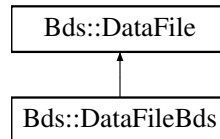
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp](#)

7.41 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.

- **BError** [streamletToChannel](#) (**BUInt** streamlet, **BUInt** &channel)

Find streamlet given channel.

- **BError** [setWritePositionForAppend](#) ()

Sets the next packet write position.

- **BError** [setReadPositionToStart](#) ()

- **BError** [packetRead](#) (**BdsDataPacket** &packet)

Reads a packet from the file.

- **BError** [packetWrite](#) (**BdsDataPacket** &packet)

Writes a packet to the file.

Static Public Member Functions

- static [DataFormats](#) [getFormats](#) ()

Get the names of the supported formats.

Additional Inherited Members

7.41.1 Detailed Description

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

7.41.2 Member Enumeration Documentation

7.41.2.1 anonymous enum

anonymous enum

Enumerator

StreamsMax	
------------	--

7.41.2.2 anonymous enum

anonymous enum

Enumerator

DefaultBlockSize	
------------------	--

7.41.2.3 PackFormat

enum [Bds::DataFileBds::PackFormat](#)

Enumerator

PackFormat_Unknown	
PackFormat_SM	
PackFormat_CM	
PackFormat_SM_CC	

7.41.3 Constructor & Destructor Documentation

7.41.3.1 DataFileBds()

[Bds::DataFileBds::DataFileBds](#) ()

7.41.3.2 ~DataFileBds()

[Bds::DataFileBds::~~DataFileBds](#) ()

7.41.4 Member Function Documentation

7.41.4.1 close()

BError [Bds::DataFileBds::close](#) () [virtual]

Close file.

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

7.41.4.2 flush()

```
BError Bds::DataFileBds::flush ( ) [virtual]
```

Flush any data to disk even if blocks are not full.

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

7.41.4.3 getDataOrder()

```
DataFile::DataOrder Bds::DataFileBds::getDataOrder ( ) [virtual]
```

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

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

7.41.4.4 getDiskBlockSize()

```
uint32_t Bds::DataFileBds::getDiskBlockSize ( )
```

Returns the data block size in bytes.

7.41.4.5 getFormats()

```
DataFormats Bds::DataFileBds::getFormats ( ) [static]
```

Get the names of the supported formats.

7.41.4.6 getInfo()

```
BError Bds::DataFileBds::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get information on open file.

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

7.41.4.7 open()

```
BError Bds::DataFileBds::open (
    BString fileName,
    BString mode ) [virtual]
```

Open the file for reading or writing.

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

7.41.4.8 packetRead()

```
BError Bds::DataFileBds::packetRead (
    BdsDataPacket & packet )
```

Reads a packet from the file.

7.41.4.9 packetWrite()

```
BError Bds::DataFileBds::packetWrite (
    BdsDataPacket & packet )
```

Writes a packet to the file.

7.41.4.10 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](#).

7.41.4.11 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](#).

7.41.4.12 setDiskBlockSize()

```
BError Bds::DataFileBds::setDiskBlockSize (
    BUInt32 blockSize )
```

Sets up file/stream block size.

7.41.4.13 setFormat()

```
BError Bds::DataFileBds::setFormat (
    BString format ) [virtual]
```

Sets the sub-format.

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

7.41.4.14 setInfo()

```
BError Bds::DataFileBds::setInfo (
    const DataInfo & dataInfo,
    const ChannelInfos & channelInfos,
    WriteOptionsList options = WriteOptionSensorData ) [virtual]
```

Sets the information.

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

7.41.4.15 setReadPositionToStart()

```
BError Bds::DataFileBds::setReadPositionToStart ( )
```

7.41.4.16 setWritePositionForAppend()

```
BError Bds::DataFileBds::setWritePositionForAppend ( )
```

Sets the next packet write position.

7.41.4.17 streamletToChannel()

```
BError Bds::DataFileBds::streamletToChannel (
    BUInt streamlet,
    BUInt & channel )
```

Find streamlet given channel.

7.41.4.18 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:

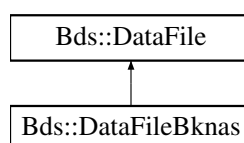
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp](#)

7.42 Bds::DataFileBknas Class Reference

Data file convertor for BKNAS format files.

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

Inheritance diagram for Bds::DataFileBknas:



Public Member Functions

- [DataFileBknas](#) ()
- **BError** [open](#) (**BString** fileName, **BString** mode)
Open the file for read or write.
- **BError** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannellInfos](#) &channellInfos, [WriteOptionsList](#) options=[WriteOptionSensorData](#))
Set information on data for write.
- **BError** [writeData](#) (const [DataBlock](#) & data)
Write a block of data.

Static Public Member Functions

- static [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.42.1 Detailed Description

Data file convertor for BKNAS format files.

7.42.2 Constructor & Destructor Documentation

7.42.2.1 DataFileBknas()

```
Bds::DataFileBknas::DataFileBknas ( )
```

7.42.3 Member Function Documentation

7.42.3.1 getFormats()

```
DataFormats Bds::DataFileBknas::getFormats ( ) [static]
```

7.42.3.2 open()

```
BError Bds::DataFileBknas::open (
    BString fileName,
    BString mode ) [virtual]
```

Open the file for read or write.

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

7.42.3.3 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](#).

7.42.3.4 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:

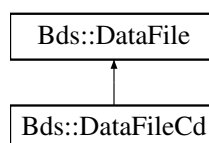
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp](#)

7.43 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.
- [BString](#) [getFixesInfo](#) ()
Get readable list of fixes that can be applied to faulty data files.
- [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 [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.43.1 Detailed Description

Data file convertor for CD1.0 and CD1.1 file formats.

7.43.2 Constructor & Destructor Documentation

7.43.2.1 DataFileCd()

```
Bds::DataFileCd::DataFileCd ( )
```

7.43.3 Member Function Documentation

7.43.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileCd::getDataOrder ( ) [virtual]
```

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

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

7.43.3.2 getFeatures()

```
int Bds::DataFileCd::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.43.3.3 getFixesInfo()

```
BString Bds::DataFileCd::getFixesInfo ( ) [virtual]
```

Get readable list of fixes that can be applied to faulty data files.

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

7.43.3.4 getFormats()

```
DataFormats Bds::DataFileCd::getFormats ( ) [static]
```

7.43.3.5 getInfo()

```
BError Bds::DataFileCd::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.43.3.6 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:

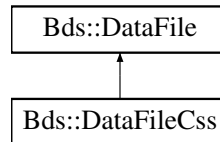
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.cpp](#)

7.44 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 [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.44.1 Detailed Description

Data file convertor for CSS format files.

7.44.2 Constructor & Destructor Documentation

7.44.2.1 DataFileCss()

```
Bds::DataFileCss::DataFileCss ( )
```

7.44.3 Member Function Documentation

7.44.3.1 `getDataOrder()`

```
DataFile::DataOrder Bds::DataFileCss::getDataOrder ( ) [virtual]
```

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

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

7.44.3.2 `getFeatures()`

```
int Bds::DataFileCss::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.44.3.3 `getFormats()`

```
DataFormats Bds::DataFileCss::getFormats ( ) [static]
```

7.44.3.4 `getInfo()`

```
BError Bds::DataFileCss::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.44.3.5 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:

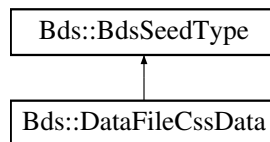
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp](#)

7.45 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](#)

7.45.1 Detailed Description

[DataFileCss](#) internal CSS data type.

7.45.2 Constructor & Destructor Documentation

7.45.2.1 DataFileCssData()

```
Bds::DataFileCssData::DataFileCssData ( )
```

7.45.2.2 ~DataFileCssData()

```
Bds::DataFileCssData::~~DataFileCssData ( )
```

7.45.3 Member Function Documentation

7.45.3.1 set()

```
BError Bds::DataFileCssData::set (
    BString line )
```

7.45.4 Member Data Documentation

7.45.4.1 calibrationFactor

```
double Bds::DataFileCssData::calibrationFactor
```

7.45.4.2 calibrationFreq

```
double Bds::DataFileCssData::calibrationFreq
```

7.45.4.3 chan

```
BString Bds::DataFileCssData::chan
```

7.45.4.4 chanid

```
int Bds::DataFileCssData::chanid
```

7.45.4.5 clip

```
BString Bds::DataFileCssData::clip
```

7.45.4.6 commId

```
int Bds::DataFileCssData::commId
```

7.45.4.7 datatype

```
BString Bds::DataFileCssData::datatype
```

7.45.4.8 dirName

```
BString Bds::DataFileCssData::dirName
```

7.45.4.9 endTime

```
double Bds::DataFileCssData::endTime
```

7.45.4.10 file

```
BFile* Bds::DataFileCssData::file
```

7.45.4.11 fileName

```
BString Bds::DataFileCssData::fileName
```

7.45.4.12 fileOffset

```
BUInt32 Bds::DataFileCssData::fileOffset
```

7.45.4.13 instType

```
BString Bds::DataFileCssData::instType
```

7.45.4.14 jdate

```
int Bds::DataFileCssData::jdate
```

7.45.4.15 loadDate

```
BString Bds::DataFileCssData::loadDate
```

7.45.4.16 nsamp

```
int Bds::DataFileCssData::nsamp
```

7.45.4.17 sampleBigEndian

```
int Bds::DataFileCssData::sampleBigEndian
```

7.45.4.18 sampleFormat

```
BUInt32 Bds::DataFileCssData::sampleFormat
```

7.45.4.19 sampleRate

```
double Bds::DataFileCssData::sampleRate
```

7.45.4.20 sampleSize

```
BUInt32 Bds::DataFileCssData::sampleSize
```

7.45.4.21 segtype

```
BString Bds::DataFileCssData::segtype
```

7.45.4.22 sta

```
BString Bds::DataFileCssData::sta
```

7.45.4.23 startTime

```
double Bds::DataFileCssData::startTime
```

7.45.4.24 wfid

```
int Bds::DataFileCssData::wfid
```

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

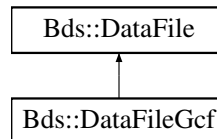
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp](#)

7.46 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.
- [BString getFixesInfo](#) ()
Get readable list of fixes that can be applied to faulty data files.
- [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 [DataFormats getFormats](#) ()

Additional Inherited Members

7.46.1 Detailed Description

Data file convertor for GCF format files.

7.46.2 Constructor & Destructor Documentation

7.46.2.1 DataFileGcf()

```
Bds::DataFileGcf::DataFileGcf ( )
```

7.46.3 Member Function Documentation

7.46.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileGcf::getDataOrder ( ) [virtual]
```

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

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

7.46.3.2 getFeatures()

```
int Bds::DataFileGcf::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.46.3.3 getFixesInfo()

```
BString Bds::DataFileGcf::getFixesInfo ( ) [virtual]
```

Get readable list of fixes that can be applied to faulty data files.

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

7.46.3.4 getFormats()

```
DataFormats Bds::DataFileGcf::getFormats ( ) [static]
```

7.46.3.5 getInfo()

```
BError Bds::DataFileGcf::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.46.3.6 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:

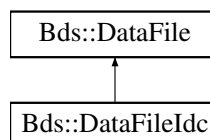
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp](#)

7.47 Bds::DataFileIdc Class Reference

This class defines the interface for IDC response data file access.

```
#include <BdsDataFileIdc.h>
```

Inheritance diagram for Bds::DataFileIdc:



Public Member Functions

- [DataFileIdc](#) ()
- int [getFeatures](#) ()
 - Get bitmask of supported features.*
- **BError** [getMetaData](#) ([ChannelInfos](#) &channelInfos, **BUInt32** options, **BList**< [DataError](#) > &errors)
 - 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 [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.47.1 Detailed Description

This class defines the interface for IDC response data file access.

7.47.2 Constructor & Destructor Documentation

7.47.2.1 DataFileIdc()

```
Bds::DataFileIdc::DataFileIdc ( )
```

7.47.3 Member Function Documentation

7.47.3.1 getFeatures()

```
int Bds::DataFileIdc::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.47.3.2 getFormats()

```
DataFormats Bds::DataFileIdc::getFormats ( ) [static]
```

7.47.3.3 getMetaData()

```
BError Bds::DataFileIdc::getMetaData (
    ChannelInfos & channelInfos,
    BUInt32 options,
    BList< DataError > & errors ) [virtual]
```

Return all known MetaData in the file.

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

7.47.3.4 setInfo()

```
BError Bds::DataFileIdc::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:

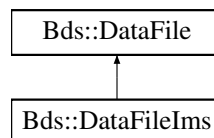
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.cpp](#)

7.48 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 [ChannelInfos](#) &channelInfos, [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](#) ([ChannelInfos](#) &channelInfos, **BUInt32** options, **BList**< [DataError](#) > &errors)
Return all known MetaData in the file.

Static Public Member Functions

- static [DataFormats](#) `getFormats` ()

Additional Inherited Members

7.48.1 Detailed Description

Data file convertor for IMS format files.

7.48.2 Constructor & Destructor Documentation

7.48.2.1 DataFileIms()

```
Bds::DataFileIms::DataFileIms ( )
```

7.48.3 Member Function Documentation

7.48.3.1 close()

```
BEError Bds::DataFileIms::close ( ) [virtual]
```

Close the file.

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

7.48.3.2 end()

```
BEError Bds::DataFileIms::end ( ) [virtual]
```

End write segment.

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

7.48.3.3 getDataOrder()

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

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

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

7.48.3.4 getFeatures()

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

Get bitmask of supported features.

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

7.48.3.5 getFormats()

```
DataFormats Bds::DataFileIms::getFormats ( ) [static]
```

7.48.3.6 getMetaData()

```
BError Bds::DataFileIms::getMetaData (
    ChannelInfos & channelInfos,
    BUInt32 options,
    BList< DataError > & errors ) [virtual]
```

Return all known MetaData in the file.

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

7.48.3.7 open()

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

Open the file for read or write.

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

7.48.3.8 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](#).

7.48.3.9 start()

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

Start writing next segment of data.

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

7.48.3.10 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:

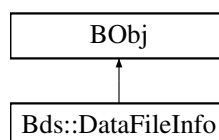
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp](#)

7.49 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** stream= **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** stream
The real-time data stream.
- **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)

7.49.1 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.

7.49.2 Constructor & Destructor Documentation

7.49.2.1 DataFileInfo()

```
Bds::DataFileInfo::DataFileInfo (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString location = BString(),
    BString format = BString(),
    BString url = BString(),
    BString stream = BString(),
    BString comment = BString(),
    BUInt32 importUserId = 0,
    BTimeStamp importTime = BTimeStamp(),
    BString state = BString() )
```

7.49.3 Member Function Documentation

7.49.3.1 getMember()

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

Reimplemented from **BObj**.

7.49.3.2 getMembers()

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

Reimplemented from **BObj**.

7.49.3.3 getType()

```
BString Bds::DataFileInfo::getType ( )
```

7.49.3.4 setMember()

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

Reimplemented from **BObj**.

7.49.3.5 setMembers()

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

Reimplemented from **BObj**.

7.49.4 Member Data Documentation

7.49.4.1 comment

```
BString Bds::DataFileInfo::comment
```

A comment on the file.

7.49.4.2 endTime

```
BTimeStamp Bds::DataFileInfo::endTime
```

The End Time.

7.49.4.3 format

```
BString Bds::DataFileInfo::format
```

The data format.

7.49.4.4 id

```
BUInt32 Bds::DataFileInfo::id
```

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

7.49.4.5 importTime

```
BTimeStamp Bds::DataFileInfo::importTime
```

The Time the data was imported.

7.49.4.6 importUserId

BUInt32 Bds::DataFileInfo::importUserId

The user ID of the importing user.

7.49.4.7 location

BString Bds::DataFileInfo::location

The storage location.

7.49.4.8 startTime

BTimeStamp Bds::DataFileInfo::startTime

The Start Time.

7.49.4.9 state

BString Bds::DataFileInfo::state

Status info on the import (importing, realtime, failed, ok etc)

7.49.4.10 stream

BString Bds::DataFileInfo::stream

The real-time data stream.

7.49.4.11 url

BString Bds::DataFileInfo::url

The URL for file access.

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

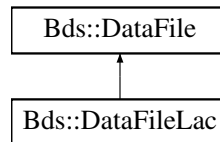
- [BdsD.h](#)
- [BdsD.cc](#)

7.50 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.
- **BString** [getFixesInfo](#) ()
Get readable list of fixes that can be applied to faulty data files.
- **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 [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.50.1 Detailed Description

Data file convertor for LAC format files.

7.50.2 Constructor & Destructor Documentation

7.50.2.1 DataFileLac()

```
Bds::DataFileLac::DataFileLac ( )
```


7.50.3 Member Function Documentation

7.50.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileLac::getDataOrder ( ) [virtual]
```

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

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

7.50.3.2 getFeatures()

```
int Bds::DataFileLac::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.50.3.3 getFixesInfo()

```
BString Bds::DataFileLac::getFixesInfo ( ) [virtual]
```

Get readable list of fixes that can be applied to faulty data files.

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

7.50.3.4 getFormats()

```
DataFormats Bds::DataFileLac::getFormats ( ) [static]
```

7.50.3.5 getInfo()

```
BError Bds::DataFileLac::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.50.3.6 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:

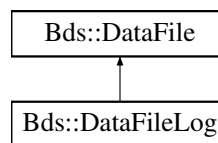
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp](#)

7.51 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 [DataFormats](#) `getFormats` ()

Additional Inherited Members

7.51.1 Detailed Description

Data file convertor for LOG format files.

7.51.2 Constructor & Destructor Documentation

7.51.2.1 DataFileLog()

```
Bds::DataFileLog::DataFileLog ( )
```

7.51.3 Member Function Documentation

7.51.3.1 end()

```
BEError Bds::DataFileLog::end ( ) [virtual]
```

End write segment.

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

7.51.3.2 getDataOrder()

```
DataFile::DataOrder Bds::DataFileLog::getDataOrder ( ) [virtual]
```

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

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

7.51.3.3 getFeatures()

```
int Bds::DataFileLog::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.51.3.4 getFormats()

```
DataFormats Bds::DataFileLog::getFormats ( ) [static]
```

7.51.3.5 getInfo()

```
BError Bds::DataFileLog::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.51.3.6 open()

```
BError Bds::DataFileLog::open (
    BString fileName,
    BString mode ) [virtual]
```

Open the file for read or write.

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

7.51.3.7 readData()

```
BError Bds::DataFileLog::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

7.51.3.8 setFormat()

```
BError Bds::DataFileLog::setFormat (
    BString format ) [virtual]
```

Set the sub-format.

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

7.51.3.9 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](#).

7.51.3.10 start()

```
BError Bds::DataFileLog::start (
    BUInt channel,
    BUInt segment ) [virtual]
```

Start writing next segment of data.

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

7.51.3.11 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](#)

7.52 Bds::DataFileOptions Class Reference

This defines a list of BDS data converter options.

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

Public Member Functions

- [DataFileOptions](#) (int options=0)
- [operator int](#) ()
- [DataFileOptions](#) & [operator|=](#) (int o)

Public Attributes

- int [optionList](#)
- [BArray< BUInt >](#) [oignoreBlockList](#)

7.52.1 Detailed Description

This defines a list of BDS data converter options.

7.52.2 Constructor & Destructor Documentation

7.52.2.1 DataFileOptions()

```
Bds::DataFileOptions::DataFileOptions (  
    int options = 0 ) [inline]
```

7.52.3 Member Function Documentation

7.52.3.1 operator int()

```
Bds::DataFileOptions::operator int ( ) [inline]
```

7.52.3.2 operator" |=()

```
DataFileOptions& Bds::DataFileOptions::operator|= (  
    int o ) [inline]
```

7.52.4 Member Data Documentation

7.52.4.1 oignoreBlockList

BArray< BUInt> Bds::DataFileOptions::oignoreBlockList

7.52.4.2 ooptionList

int Bds::DataFileOptions::ooptionList

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

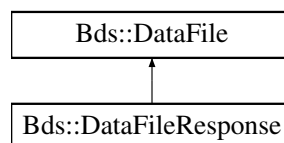
- /src/blacknest/bds/bds/bdsDataLib/[BdsDataFile.h](#)

7.53 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](#) ([ChannellInfos](#) &channellInfos, **BUInt32** options, **BList**< [DataError](#) > &errors)
Return all known MetaData in the file.
- **BError** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannellInfos](#) &channellInfos, [WriteOptionsList](#) options)
Set information on data for write.

Static Public Member Functions

- static [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.53.1 Detailed Description

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

7.53.2 Constructor & Destructor Documentation

7.53.2.1 DataFileResponse()

```
Bds::DataFileResponse::DataFileResponse ( )
```

7.53.3 Member Function Documentation

7.53.3.1 getFeatures()

```
int Bds::DataFileResponse::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.53.3.2 getFormats()

```
DataFormats Bds::DataFileResponse::getFormats ( ) [static]
```

7.53.3.3 getMetaData()

```
BError Bds::DataFileResponse::getMetaData (
    ChannelInfos & channelInfos,
    BUInt32 options,
    BList< DataError > & errors ) [virtual]
```

Return all known MetaData in the file.

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

7.53.3.4 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:

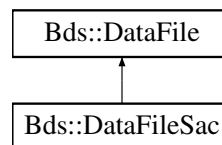
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp](#)

7.54 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** [getMetaData](#) ([ChannelInfos](#) &channelInfos, **BUInt32** options, **BList**< [DataError](#) > &errors)
Return all known MetaData in the file.
- **BError** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannelInfos](#) &channelInfos, [WriteOptionsList](#) options=[WriteOptionNone](#))
Set information on data for write.

Static Public Member Functions

- static [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.54.1 Detailed Description

Data file convertor for SAC format files.

7.54.2 Constructor & Destructor Documentation

7.54.2.1 DataFileSac()

```
Bds::DataFileSac::DataFileSac ( )
```

7.54.3 Member Function Documentation

7.54.3.1 getFeatures()

```
int Bds::DataFileSac::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.54.3.2 getFormats()

```
DataFormats Bds::DataFileSac::getFormats ( ) [static]
```

7.54.3.3 getMetaData()

```
BError Bds::DataFileSac::getMetaData (
    ChannelInfos & channelInfos,
    BUInt32 options,
    BList< DataError > & errors ) [virtual]
```

Return all known MetaData in the file.

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

7.54.3.4 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:

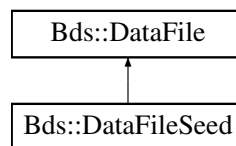
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp](#)

7.55 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.
- **BString** [getFixesInfo](#) ()
Get readable list of fixes that can be applied to faulty data files.
- **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** [getMetaData](#) ([ChannellInfos](#) &channellInfos, **BUInt32** options, **BList**< [DataError](#) > &errors)
Return all known MetaData in the file.

- **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 [DataFormats](#) `getFormats` ()

Static Public Attributes

- static **BError** [omsrErr](#)
MSR processing error.
- static int [onoLock](#)
Disable libmseed locking.

Additional Inherited Members

7.55.1 Detailed Description

Data file convertor for SEED file formats.

7.55.2 Constructor & Destructor Documentation

7.55.2.1 DataFileSeed()

```
Bds::DataFileSeed::DataFileSeed ( )
```

7.55.2.2 ~DataFileSeed()

```
Bds::DataFileSeed::~~DataFileSeed ( )
```

7.55.3 Member Function Documentation

7.55.3.1 close()

BError Bds::DataFileSeed::close () [virtual]

Close the file.

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

7.55.3.2 end()

BError Bds::DataFileSeed::end () [virtual]

End write segment.

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

7.55.3.3 getDataOrder()

[DataFile::DataOrder](#) Bds::DataFileSeed::getDataOrder () [virtual]

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

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

7.55.3.4 getFeatures()

int Bds::DataFileSeed::getFeatures () [virtual]

Get bitmask of supported features.

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

7.55.3.5 getFixesInfo()

BString Bds::DataFileSeed::getFixesInfo () [virtual]

Get readable list of fixes that can be applied to faulty data files.

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

7.55.3.6 getFormats()

```
DataFormats Bds::DataFileSeed::getFormats ( ) [static]
```

7.55.3.7 getInfo()

```
BError Bds::DataFileSeed::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.55.3.8 getMetaData()

```
BError Bds::DataFileSeed::getMetaData (
    ChannelInfos & channelInfos,
    BUInt32 options,
    BList< DataError > & errors ) [virtual]
```

Return all known MetaData in the file.

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

7.55.3.9 msrFileWrite()

```
void Bds::DataFileSeed::msrFileWrite (
    void * data,
    int len )
```

7.55.3.10 readData()

```
BError Bds::DataFileSeed::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

7.55.3.11 setFormat()

```
BError Bds::DataFileSeed::setFormat (
    BString format ) [virtual]
```

Set the sub-format.

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

7.55.3.12 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](#).

7.55.3.13 start()

```
BError Bds::DataFileSeed::start (
    BUInt channel,
    BUInt segment ) [virtual]
```

Start writing next segment of data.

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

7.55.3.14 writeData()

```
BError Bds::DataFileSeed::writeData (
    const DataBlock & data ) [virtual]
```

Write a block of data.

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

7.55.4 Member Data Documentation

7.55.4.1 omsrErr

BError Bds::DataFileSeed::omsrErr [static]

MSR processing error.

7.55.4.2 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:

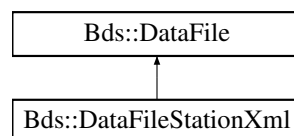
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp

7.56 Bds::DataFileStationXml Class Reference

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

```
#include <BdsDataFileStationXml.h>
```

Inheritance diagram for Bds::DataFileStationXml:



Public Member Functions

- [DataFileStationXml](#) ()
- int [getFeatures](#) ()
Get bitmask of supported features.
- **BError** [setInfo](#) (const [DataInfo](#) &dataInfo, const [ChannellInfos](#) &channellInfos, [WriteOptionsList](#) options)
Set information on data for write.
- **BError** [getMetaData](#) ([ChannellInfos](#) &channellInfos, **BUInt32** options, **BList**< [DataError](#) > &errors)
Return all known MetaData in the file.

Static Public Member Functions

- static [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.56.1 Detailed Description

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

7.56.2 Constructor & Destructor Documentation

7.56.2.1 DataFileStationXml()

```
Bds::DataFileStationXml::DataFileStationXml ( )
```

7.56.3 Member Function Documentation

7.56.3.1 getFeatures()

```
int Bds::DataFileStationXml::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.56.3.2 getFormats()

```
DataFormats Bds::DataFileStationXml::getFormats ( ) [static]
```

7.56.3.3 getMetaData()

```
BError Bds::DataFileStationXml::getMetaData (
    ChannelInfos & channelInfos,
    BUInt32 options,
    BList< DataError > & errors ) [virtual]
```

Return all known MetaData in the file.

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

7.56.3.4 setInfo()

```
BError Bds::DataFileStationXml::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:

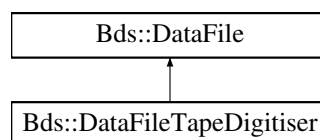
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.cpp](#)

7.57 Bds::DataFileTapeDigitiser Class Reference

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

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

Inheritance diagram for Bds::DataFileTapeDigitiser:



Public Member Functions

- [DataFileTapeDigitiser](#) ()
- **BError** [open](#) (**BString** fileName, **BString** mode)
Open the file for reading or writing.
- **BError** [getInfo](#) ([DataInfo](#) &dataInfo, [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 [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.57.1 Detailed Description

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

7.57.2 Constructor & Destructor Documentation

7.57.2.1 DataFileTapeDigitiser()

```
Bds::DataFileTapeDigitiser::DataFileTapeDigitiser ( )
```

7.57.3 Member Function Documentation

7.57.3.1 getFormats()

```
DataFormats Bds::DataFileTapeDigitiser::getFormats ( ) [static]
```

7.57.3.2 getInfo()

```
BError Bds::DataFileTapeDigitiser::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.57.3.3 open()

```
BError Bds::DataFileTapeDigitiser::open (
    BString fileName,
    BString mode ) [virtual]
```

Open the file for reading or writing.

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

7.57.3.4 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:

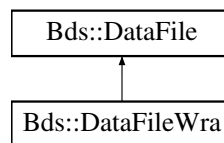
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp](#)

7.58 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.
- **BString** [getFixesInfo](#) ()
Get readable list of fixes that can be applied to faulty data files.
- **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 [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.58.1 Detailed Description

Data file convertor for WRA format files.

7.58.2 Constructor & Destructor Documentation

7.58.2.1 DataFileWra()

```
Bds::DataFileWra::DataFileWra ( )
```

7.58.3 Member Function Documentation

7.58.3.1 getDataOrder()

```
DataFile::DataOrder Bds::DataFileWra::getDataOrder ( ) [virtual]
```

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

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

7.58.3.2 getFeatures()

```
int Bds::DataFileWra::getFeatures ( ) [virtual]
```

Get bitmask of supported features.

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

7.58.3.3 getFixesInfo()

```
BString Bds::DataFileWra::getFixesInfo ( ) [virtual]
```

Get readable list of fixes that can be applied to faulty data files.

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

7.58.3.4 getFormats()

```
DataFormats Bds::DataFileWra::getFormats ( ) [static]
```

7.58.3.5 getInfo()

```
BError Bds::DataFileWra::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.58.3.6 readData()

```
BError Bds::DataFileWra::readData (
    BUInt32 channel,
    BUInt segment,
    BUInt32 blockNumber,
    DataBlock & dataBlock ) [virtual]
```

Read a block.

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

7.58.3.7 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:

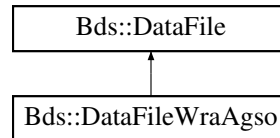
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h](#)
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp](#)

7.59 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 [DataFormats](#) [getFormats](#) ()

Additional Inherited Members

7.59.1 Detailed Description

Data file convertor for WRA AGSO format files.

7.59.2 Constructor & Destructor Documentation

7.59.2.1 DataFileWraAgso()

```
Bds::DataFileWraAgso::DataFileWraAgso ( )
```

7.59.3 Member Function Documentation

7.59.3.1 `getDataOrder()`

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

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

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

7.59.3.2 `getFeatures()`

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

Get bitmask of supported features.

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

7.59.3.3 `getFormats()`

```
DataFormats Bds::DataFileWraAgso::getFormats ( ) [static]
```

7.59.3.4 `getInfo()`

```
BError Bds::DataFileWraAgso::getInfo (
    DataInfo & dataInfo,
    DataFileOptions options,
    BList< DataError > & errors ) [virtual]
```

Get info on data.

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

7.59.3.5 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](#)

7.60 Bds::DataFormat Class Reference

This holds information on a seismic data format.

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

Public Member Functions

- [DataFormat](#) ([BList](#)< [BString](#) > [names](#)= [BList](#)< [BString](#) >(), [BInt32](#) [dataRead](#)=0, [BInt32](#) [dataWrite](#)=0, [BInt32](#) [metadataRead](#)=0, [BInt32](#) [metadataWrite](#)=0, [BString](#) [extension](#)= [BString](#)(), [BString](#) [description](#)= [BString](#)())

Public Attributes

- [BList](#)< [BString](#) > [names](#)
The format names.
- [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.
- [BString](#) [description](#)
The description.

7.60.1 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..

7.60.2 Constructor & Destructor Documentation

7.60.2.1 DataFormat()

```
Bds::DataFormat::DataFormat (
    BList< BString > names = BList< BString >(),
    BInt32 dataRead = 0,
    BInt32 dataWrite = 0,
    BInt32 metadataRead = 0,
    BInt32 metadataWrite = 0,
    BString extension = BString(),
    BString description = BString() )
```

7.60.3 Member Data Documentation

7.60.3.1 dataRead

```
BInt32 Bds::DataFormat::dataRead
```

Ability to read data.

7.60.3.2 dataWrite

```
BInt32 Bds::DataFormat::dataWrite
```

Ability to write data.

7.60.3.3 description

```
BString Bds::DataFormat::description
```

The description.

7.60.3.4 extension

BString Bds::DataFormat::extension

Default filename extension.

7.60.3.5 metadataRead

Int32 Bds::DataFormat::metadataRead

MetaData read supported.

7.60.3.6 metadataWrite

Int32 Bds::DataFormat::metadataWrite

MetaData write supported.

7.60.3.7 names

List< BString > Bds::DataFormat::names

The format names.

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

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

7.61 Bds::DataFormatAll Class Reference

This class defines the interface for generic data file access.

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

Public Member Functions

- [DataFormatAll](#) ()
- [~DataFormatAll](#) ()
- **BError** formatList ([DataFormats](#) &formats)
- **BError** formatGet (**BString** format, [DataFile](#) *&dataFile, [DataFormatSet](#) formatSet=[DataFormatSetNone](#))
Returns a list of all data converters.
- **BString** formatGetExtension (**BString** format)
Searches all of the BDS data converters to find out if any supports the given format. If one is found the appropriate data converter is initialised and a pointer to it returned in file.

Protected Member Functions

- int `findFormat` (`DataFormats` dataFormats, `BString` string, `DataFormatSet` formatSet)
Returns the default file extension for the given format.

7.61.1 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.

7.61.2 Constructor & Destructor Documentation

7.61.2.1 DataFormatAll()

```
Bds::DataFormatAll::DataFormatAll ( )
```

7.61.2.2 ~DataFormatAll()

```
Bds::DataFormatAll::~~DataFormatAll ( )
```

7.61.3 Member Function Documentation

7.61.3.1 findFormat()

```
int Bds::DataFormatAll::findFormat (
    DataFormats dataFormats,
    BString string,
    DataFormatSet formatSet ) [protected]
```

Returns the default file extension for the given format.

7.61.3.2 formatGet()

```
BError Bds::DataFormatAll::formatGet (
    BString format,
    DataFile *& dataFile,
    DataFormatSet formatSet = DataFormatSetNone )
```

Returns a list of all data converters.

7.61.3.3 formatGetExtension()

```
BString Bds::DataFormatAll::formatGetExtension (
    BString format )
```

Searches all of the BDS data converters to find out if any supports the given format. If one is found the appropriate data converter is initialised and a pointer to it returned in file.

7.61.3.4 formatList()

```
BError Bds::DataFormatAll::formatList (
    DataFormats & 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](#)

7.62 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.

7.62.1 Detailed Description

This defines a handle to a sensor data stream/file when opened for read or write.

7.62.2 Constructor & Destructor Documentation

7.62.2.1 DataHandle()

```
Bds::DataHandle::DataHandle (
    BUInt32 handle = 0,
    BUInt32 dataFileId = 0 )
```

7.62.3 Member Data Documentation

7.62.3.1 dataFileId

```
BUInt32 Bds::DataHandle::dataFileId
```

The data file ID if opened for write.

7.62.3.2 handle

```
BUInt32 Bds::DataHandle::handle
```

Opaque file handle.

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

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

7.63 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.

7.63.1 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.

7.63.2 Constructor & Destructor Documentation

7.63.2.1 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 >() )
```

7.63.3 Member Data Documentation

7.63.3.1 array

```
BString Bds::DataInfo::array
```

The Seismic Array that all of the channels are from, if just one.

7.63.3.2 channels

```
BArray< BArray<DataChannel > > Bds::DataInfo::channels
```

The Data channels. Each channel can have multiple segments of data.

7.63.3.3 description

```
BString Bds::DataInfo::description
```

The Comment.

7.63.3.4 endTime

```
BTimeStamp Bds::DataInfo::endTime
```

The End Time.

7.63.3.5 info

```
BDict< BString > Bds::DataInfo::info
```

Info on the set of channels.

7.63.3.6 infoExtra

```
BDict< BString > Bds::DataInfo::infoExtra
```

Extra Info on the set of channels. Used for extended error/logging information.

7.63.3.7 startTime

```
BTimeStamp Bds::DataInfo::startTime
```

The Start Time.

7.63.3.8 synchronous

```
BUInt32 Bds::DataInfo::synchronous
```

The channels are synchronously sampled.

7.63.3.9 warnings

```
BList< BString > Bds::DataInfo::warnings
```

Warnings on the data set.

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

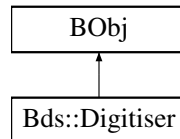
- [BdsD.h](#)
- [BdsD.cc](#)

7.64 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](#)(),
[BUInt32](#) 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.

7.64.1 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.

7.64.2 Constructor & Destructor Documentation

7.64.2.1 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 )
```

7.64.3 Member Function Documentation

7.64.3.1 getMember()

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

Reimplemented from **BObj**.

7.64.3.2 getMembers()

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

Reimplemented from **BObj**.

7.64.3.3 getType()

```
BString Bds::Digitiser::getType ( )
```

7.64.3.4 setMember()

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

Reimplemented from **BObj**.

7.64.3.5 setMembers()

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

Reimplemented from **BObj**.

7.64.4 Member Data Documentation

7.64.4.1 baseSamplingFrequency

```
BFloat64 Bds::Digitiser::baseSamplingFrequency
```

The base sampling frequency.

7.64.4.2 endTime

```
BTimeStamp Bds::Digitiser::endTime
```

The End Time the channel was available.

7.64.4.3 gain

```
BFloat64 Bds::Digitiser::gain
```

The overall gain of the digitiser at the manufacturers calibration frequency. (For information only)

7.64.4.4 id

```
BUInt32 Bds::Digitiser::id
```

The ID.

7.64.4.5 initialSamplingFrequency

```
BFloat64 Bds::Digitiser::initialSamplingFrequency
```

The initial pre-decimation sampling frequency.

7.64.4.6 name

```
BString Bds::Digitiser::name
```

The Digitisers name.

7.64.4.7 numberChannels

```
BUInt32 Bds::Digitiser::numberChannels
```

The number of supported channels.

7.64.4.8 serialNumber

```
BString Bds::Digitiser::serialNumber
```

The digitisers's serial number.

7.64.4.9 shared

```
BInt32 Bds::Digitiser::shared
```

This digitiser is shared.

7.64.4.10 `startTime`

BTimeStamp `Bds::Digitiser::startTime`

The Start Time.

7.64.4.11 `type`

BString `Bds::Digitiser::type`

The Digitisers type.

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

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

7.65 `Bds::Event` Class Reference

This class defines a seismic event.

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

Public Member Functions

- [Event](#) ([BUInt32](#) `id`=0, [BUInt32](#) `userId`=0, [BString](#) `type`= [BString](#)(), [BString](#) `title`= [BString](#)(), [BString](#) `network`= [BString](#)(), [BString](#) `source`= [BString](#)(), [BTimeStamp](#) `startTime`= [BTimeStamp](#)(), [BTimeStamp](#) `endTime`= [BTimeStamp](#)(), [BTimeStamp](#) `eventTime`= [BTimeStamp](#)(), [BFloat64](#) `longitude`=0, [BFloat64](#) `latitude`=0, [BFloat64](#) `elevation`=0, [BFloat64](#) `waterDepth`=0, [BFloat64](#) `magnitude`=0, [BString](#) `magnitudeUnits`= [BString](#)(), [BString](#) `description`= [BString](#)(), [BString](#) `notes`= [BString](#)(), [BDict](#)< [BString](#) > `extra`= [BDict](#)< [BString](#) >(), [BList](#)< [SelectionChannel](#) > `dataChannels`= [BList](#)< [SelectionChannel](#) >())

Public Attributes

- [BUInt32](#) `id`
Unique id defining this event within the BDS system.
- [BUInt32](#) `userId`
User ID for initial creator of this event. This allows them to edit these events.
- [BString](#) `type`
The event type (a hierarchy of types)
- [BString](#) `title`
Some text describing the event.
- [BString](#) `network`
Unique network for this event if a project.
- [BString](#) `source`
Unique source for this event if a project.

- **BTimeStamp** [startTime](#)
The startTime of the event to the nearest microsecond. For data access.
- **BTimeStamp** [endTime](#)
The endTime of the event to the nearest microsecond. For data access.
- **BTimeStamp** [eventTime](#)
The actual time of the event to the nearest microsecond.
- **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** [waterDepth](#)
Water depth of the event if in water.
- **BFloat64** [magnitude](#)
Magnitude of the event.
- **BString** [magnitudeUnits](#)
Magnitude Units.
- **BString** [description](#)
General description of the Event/Project.
- **BString** [notes](#)
General notes on the Event/project.
- **BDict**< **BString** > [extra](#)
An array of name value pairs for extra metadata specific to particular events.
- **BList**< [SelectionChannel](#) > [dataChannels](#)
List of BDS Channels of associated sensor data files if any.

7.65.1 Detailed Description

This class defines a seismic event.

Each event defines Metadata for a seismic event. this includes a list of all of the data channels relavent to the event.

7.65.2 Constructor & Destructor Documentation

7.65.2.1 Event()

```
Bds::Event::Event (
    BUInt32 id = 0,
    BUInt32 userId = 0,
    BString type = BString(),
    BString title = BString(),
    BString network = BString(),
    BString source = BString(),
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BTimeStamp eventTime = BTimeStamp(),
```

```

    BFloat64 longitude = 0,
    BFloat64 latitude = 0,
    BFloat64 elevation = 0,
    BFloat64 waterDepth = 0,
    BFloat64 magnitude = 0,
    BString magnitudeUnits = BString(),
    BString description = BString(),
    BString notes = BString(),
    BDict< BString > extra = BDict< BString >(),
    BList< SelectionChannel > dataChannels = BList<SelectionChannel >() )

```

7.65.3 Member Data Documentation

7.65.3.1 dataChannels

```
BList<SelectionChannel > Bds::Event::dataChannels
```

List of BDS Channels of associated sensor data files if any.

7.65.3.2 description

```
BString Bds::Event::description
```

General description of the Event/Project.

7.65.3.3 elevation

```
BFloat64 Bds::Event::elevation
```

The ground level elevation in meters from the WGS84 ellipsoid (Sea level)

7.65.3.4 endTime

```
BTimeStamp Bds::Event::endTime
```

The endTime of the event to the nearest microsecond. For data access.

7.65.3.5 eventTime

BTimeStamp Bds::Event::eventTime

The actual time of the event to the nearest microsecond.

7.65.3.6 extra

BDict< **BString** > Bds::Event::extra

An array of name value pairs for extra metadata specific to particular events.

7.65.3.7 id

BUInt32 Bds::Event::id

Unique id defining this event within the BDS system.

7.65.3.8 latitude

BFloat64 Bds::Event::latitude

The Latitude in degrees using the WGS84 datum.

7.65.3.9 longitude

BFloat64 Bds::Event::longitude

The longitude in degrees using the WGS84 datum.

7.65.3.10 magnitude

BFloat64 Bds::Event::magnitude

Magnitude of the event.

7.65.3.11 magnitudeUnits

BString Bds::Event::magnitudeUnits

Magnitude Units.

7.65.3.12 network

BString Bds::Event::network

Unique network for this event if a project.

7.65.3.13 notes

BString Bds::Event::notes

General notes on the Event/project.

7.65.3.14 source

BString Bds::Event::source

Unique source for this event if a project.

7.65.3.15 startTime

BTimeStamp Bds::Event::startTime

The startTime of the event to the nearest microsecond. For data access.

7.65.3.16 title

BString Bds::Event::title

Some text describing the event.

7.65.3.17 type

BString Bds::Event::type

The event type (a hierarchy of types)

7.65.3.18 userId

BUInt32 Bds::Event::userId

User ID for initial creator of this event. This allows them to edit these events.

7.65.3.19 waterDepth

BFloat64 Bds::Event::waterDepth

Water depth of the event if in water.

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

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

7.66 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.

7.66.1 Detailed Description

This class defines an entry in an Amplitude/Phase [Response](#) table.

7.66.2 Constructor & Destructor Documentation

7.66.2.1 Fap()

```
Bds::Fap::Fap (
    BFloat64 frequency = 0,
    BFloat64 amplitude = 0,
    BFloat64 phase = 0 )
```

7.66.3 Member Data Documentation

7.66.3.1 amplitude

```
BFloat64 Bds::Fap::amplitude
```

The Amplitude.

7.66.3.2 frequency

```
BFloat64 Bds::Fap::frequency
```

The frequency.

7.66.3.3 phase

```
BFloat64 Bds::Fap::phase
```

The Phase.

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

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

7.67 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.

7.67.1 Detailed Description

This class defines an FIR response table.

This has an array of the A and B coefficients.

7.67.2 Constructor & Destructor Documentation

7.67.2.1 Fir()

```
Bds::Fir::Fir (
    BArray< FirEntry > b = BArray<FirEntry >(),
    BArray< FirEntry > a = BArray<FirEntry >() )
```

7.67.3 Member Data Documentation

7.67.3.1 a

```
BArray<FirEntry > Bds::Fir::a
```

Denominator.

7.67.3.2 b

```
BArray<FirEntry > Bds::Fir::b
```

Numerator.

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

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

7.68 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.

7.68.1 Detailed Description

This class defines an entry in a FIR coefficient table.

7.68.2 Constructor & Destructor Documentation

7.68.2.1 FirEntry()

```
Bds::FirEntry::FirEntry (
    BFloat64 coefficient = 0,
    BFloat64 error = 0 )
```

7.68.3 Member Data Documentation

7.68.3.1 coefficient

BFloat64 Bds::FirEntry::coefficient

Value.

7.68.3.2 error

BFloat64 Bds::FirEntry::error

Error.

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

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

7.69 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](#)

7.69.1 Detailed Description

[DataFileGcf](#) internal GCF channel information.

7.69.2 Member Data Documentation

7.69.2.1 channel

BUInt32 Bds::GcfChannel::channel

7.69.2.2 format

```
BUInt Bds::GcfChannel::format
```

7.69.2.3 sampleRate

```
BUInt Bds::GcfChannel::sampleRate
```

7.69.2.4 streamId

```
BString Bds::GcfChannel::streamId
```

7.69.2.5 systemId

```
BString Bds::GcfChannel::systemId
```

7.69.2.6 type

```
BUInt Bds::GcfChannel::type
```

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

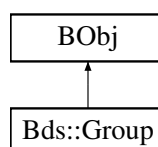
- [/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h](#)

7.70 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.

7.70.1 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.

7.70.2 Constructor & Destructor Documentation

7.70.2.1 Group()

```
Bds::Group::Group (
    BUInt32 id = 0,
    BString group = BString(),
    BString description = BString() )
```

7.70.3 Member Function Documentation

7.70.3.1 getMember()

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

Reimplemented from [BObj](#).

7.70.3.2 getMembers()

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

Reimplemented from **BObj**.

7.70.3.3 getType()

```
BString Bds::Group::getType ( )
```

7.70.3.4 setMember()

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

Reimplemented from **BObj**.

7.70.3.5 setMembers()

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

Reimplemented from **BObj**.

7.70.4 Member Data Documentation

7.70.4.1 description

```
BString Bds::Group::description
```

The Groups description.

7.70.4.2 group

BString Bds::Group::group

The [Group](#) name.

7.70.4.3 id

BUInt32 Bds::Group::id

The unique id.

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

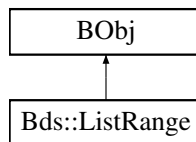
- [BdsD.h](#)
- [BdsD.cc](#)

7.71 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.

7.71.1 Detailed Description

This class defines an integer based range.

It is used for limit the number of items returned in selections etc.

7.71.2 Constructor & Destructor Documentation

7.71.2.1 ListRange()

```
Bds::ListRange::ListRange (
    BUInt32 start = 0,
    BUInt32 number = 0,
    BInt32 reverse = 0 )
```

7.71.3 Member Function Documentation

7.71.3.1 getMember()

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

Reimplemented from **BObj**.

7.71.3.2 getMembers()

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

Reimplemented from **BObj**.

7.71.3.3 getType()

```
BString Bds::ListRange::getType ( )
```

7.71.3.4 setMember()

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

Reimplemented from **BObj**.

7.71.3.5 setMembers()

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

Reimplemented from **BObj**.

7.71.4 Member Data Documentation

7.71.4.1 number

```
BUInt32 Bds::ListRange::number
```

The number of items.

7.71.4.2 reverse

```
BInt32 Bds::ListRange::reverse
```

List from end.

7.71.4.3 start

```
BUInt32 Bds::ListRange::start
```

The start position.

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

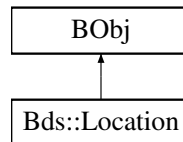
- [BdsD.h](#)
- [BdsD.cc](#)

7.72 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 channel= BString(), BString datum= BString(), BFloat64 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 channel
The channel this location is for. If blank the location is for the [Station](#) itself.
- 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 an array in an easterly direction if array known/given.
- BFloat64 arrayOffsetNorth
The Array offset in an array in a northerly direction if array known/given.

7.72.1 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.

7.72.2 Constructor & Destructor Documentation

7.72.2.1 Location()

```
Bds::Location::Location (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString network = BString(),
    BString station = BString(),
    BString channel = BString(),
    BString datum = BString(),
    BFloat64 longitude = 0,
    BFloat64 latitude = 0,
    BFloat64 elevation = 0,
    BFloat64 arrayOffsetEast = 0,
    BFloat64 arrayOffsetNorth = 0 )
```

7.72.3 Member Function Documentation

7.72.3.1 getMember()

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

Reimplemented from **BObj**.

7.72.3.2 getMembers()

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

Reimplemented from **BObj**.

7.72.3.3 getType()

```
BString Bds::Location::getType ( )
```

7.72.3.4 setMember()

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

Reimplemented from **BObj**.

7.72.3.5 setMembers()

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

Reimplemented from **BObj**.

7.72.4 Member Data Documentation

7.72.4.1 arrayOffsetEast

```
BFloat64 Bds::Location::arrayOffsetEast
```

The Array offset in an array in an easterly direction if array known/given.

7.72.4.2 arrayOffsetNorth

```
BFloat64 Bds::Location::arrayOffsetNorth
```

The Array offset in an array in a northerly direction if array known/given.

7.72.4.3 channel

```
BString Bds::Location::channel
```

The channel this location is for. If blank the location is for the [Station](#) itself.

7.72.4.4 datum

BString Bds::Location::datum

The locations Datum.

7.72.4.5 elevation

BFloat64 Bds::Location::elevation

The ground level elevation in meters from the WGS84 ellipsoid (Sea level)

7.72.4.6 endTime

BTimeStamp Bds::Location::endTime

The End Time the channel was available.

7.72.4.7 id

BUInt32 Bds::Location::id

The ID.

7.72.4.8 latitude

BFloat64 Bds::Location::latitude

The Latitude in degrees using the WGS84 datum.

7.72.4.9 longitude

BFloat64 Bds::Location::longitude

The longitude in degrees using the WGS84 datum.

7.72.4.10 network

BString Bds::Location::network

The Network/Organisation Name.

7.72.4.11 startTime

BTimeStamp Bds::Location::startTime

The Start Time.

7.72.4.12 station

BString Bds::Location::station

The station this location is for.

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

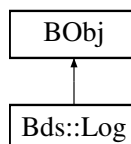
- [BdsD.h](#)
- [BdsD.cc](#)

7.73 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.

7.73.1 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.

7.73.2 Constructor & Destructor Documentation

7.73.2.1 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() )
```

7.73.3 Member Function Documentation

7.73.3.1 getMember()

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

Reimplemented from **BObj**.

7.73.3.2 getMembers()

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

Reimplemented from **BObj**.

7.73.3.3 getType()

```
BString Bds::Log::getType ( )
```

7.73.3.4 setMember()

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

Reimplemented from **BObj**.

7.73.3.5 setMembers()

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

Reimplemented from **BObj**.

7.73.4 Member Data Documentation

7.73.4.1 description

```
BString Bds::Log::description
```

The Description of the change.

7.73.4.2 id

BUInt32 Bds::Log::id

The unique id.

7.73.4.3 priority

BUInt32 Bds::Log::priority

The priority 0 to 5.

7.73.4.4 subSystem

BString Bds::Log::subSystem

The SubSystem.

7.73.4.5 time

BTimeStamp Bds::Log::time

The Time.

7.73.4.6 title

BString Bds::Log::title

The Changes title.

7.73.4.7 type

BString Bds::Log::type

The Type.

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

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

7.74 Bds::LogSelect Class Reference

This defines the selection criteria 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.

7.74.1 Detailed Description

This defines the selection criteria when requesting a set of log entries.

7.74.2 Constructor & Destructor Documentation

7.74.2.1 LogSelect()

```
Bds::LogSelect::LogSelect (
    BTimeStamp startTime = BTimeStamp(),
    BString type = BString(),
    BUInt32 priority = 0,
    BString subSystem = BString() )
```

7.74.3 Member Data Documentation

7.74.3.1 priority

BUInt32 Bds::LogSelect::priority

The priority 0 to 5.

7.74.3.2 startTime

BTimeStamp Bds::LogSelect::startTime

The start time.

7.74.3.3 subSystem

BString Bds::LogSelect::subSystem

The SubSystem.

7.74.3.4 type

BString Bds::LogSelect::type

The Type.

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

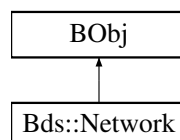
- [BdsD.h](#)
- [BdsD.cc](#)

7.75 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.

7.75.1 Detailed Description

This class defines a seismic [Network](#) organisation.

Typical Seismic Networks are "BN", IDC" etc.

7.75.2 Constructor & Destructor Documentation

7.75.2.1 Network()

```
Bds::Network::Network (
    BUInt32 id = 0,
    BString network = BString(),
    BString description = BString(),
    BList< BString > stations = BList< BString >() )
```

7.75.3 Member Function Documentation

7.75.3.1 getMember()

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

Reimplemented from **BObj**.

7.75.3.2 getMembers()

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

Reimplemented from **BObj**.

7.75.3.3 getType()

```
BString Bds::Network::getType ( )
```

7.75.3.4 setMember()

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

Reimplemented from **BObj**.

7.75.3.5 setMembers()

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

Reimplemented from **BObj**.

7.75.4 Member Data Documentation

7.75.4.1 description

```
BString Bds::Network::description
```

The organisations description.

7.75.4.2 id

```
BUInt32 Bds::Network::id
```

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

7.75.4.3 network

```
BString Bds::Network::network
```

The name.

7.75.4.4 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:

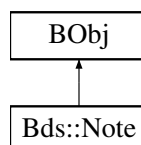
- [BdsD.h](#)
- [BdsD.cc](#)

7.76 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(), BUInt32 eventId=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.
- 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.
- BUInt32 eventId
The event ID associated with this note.

7.76.1 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.

7.76.2 Constructor & Destructor Documentation

7.76.2.1 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(),
    BUInt32 eventId = 0 )
```

7.76.3 Member Function Documentation

7.76.3.1 getMember()

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

Reimplemented from **BObj**.

7.76.3.2 getMembers()

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

Reimplemented from **BObj**.

7.76.3.3 getType()

```
BString Bds::Note::getType ( )
```

7.76.3.4 setMember()

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

Reimplemented from **BObj**.

7.76.3.5 setMembers()

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

Reimplemented from **BObj**.

7.76.4 Member Data Documentation

7.76.4.1 channel

```
BString Bds::Note::channel
```

The Channels name.

7.76.4.2 dataFileId

BUInt32 Bds::Note::dataFileId

The data file id associated with this note.

7.76.4.3 description

BString Bds::Note::description

The Description.

7.76.4.4 docFormat

BString Bds::Note::docFormat

Document format if any.

7.76.4.5 docUrl

BString Bds::Note::docUrl

Document Url if any.

7.76.4.6 endTime

BTimeStamp Bds::Note::endTime

The End Time note is for.

7.76.4.7 errorNumber

BInt32 Bds::Note::errorNumber

Error number if error.

7.76.4.8 eventId

BUInt32 Bds::Note::eventId

The event ID associated with this note.

7.76.4.9 id

BUInt32 Bds::Note::id

The unique id.

7.76.4.10 importFilename

BString Bds::Note::importFilename

The import filename.

7.76.4.11 network

BString Bds::Note::network

The [Network](#) Name.

7.76.4.12 source

BString Bds::Note::source

The Data [Source](#).

7.76.4.13 startTime

BTimeStamp Bds::Note::startTime

The Start Time note is for.

7.76.4.14 station

BString Bds::Note::station

The Station/Array name.

7.76.4.15 timeAdded

BTimeStamp Bds::Note::timeAdded

The Time Entered.

7.76.4.16 title

BString Bds::Note::title

The title.

7.76.4.17 type

BString Bds::Note::type

The Type (note, warning, error ...)

7.76.4.18 user

BString Bds::Note::user

The user.

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

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

7.77 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](#)

7.77.1 Detailed Description

This class defines an X,Y location.

The class simply stores the x and y point values.

7.77.2 Constructor & Destructor Documentation

7.77.2.1 Point()

```
Bds::Point::Point (
    BFloat64 x = 0,
    BFloat64 y = 0 )
```

7.77.3 Member Data Documentation

7.77.3.1 x

```
BFloat64 Bds::Point::x
```

7.77.3.2 y

```
BFloat64 Bds::Point::y
```

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

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

7.78 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.

7.78.1 Detailed Description

This class defines a Pole/Zero [Response](#).

It consists of an array of Complex poles and an array of Complex zeros.

7.78.2 Constructor & Destructor Documentation

7.78.2.1 PoleZero()

```
Bds::PoleZero::PoleZero (
    BArray< BComplex > poles = BArray< BComplex >(),
    BArray< BComplex > zeros = BArray< BComplex >() )
```

7.78.3 Member Data Documentation

7.78.3.1 poles

```
BArray< BComplex > Bds::PoleZero::poles
```

Poles.

7.78.3.2 zeros

```
BArray< BComplex > Bds::PoleZero::zeros
```

Zeros.

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

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

7.79 Bds::Polynomial Class Reference

This class defines an [Polynomial](#) response table.

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

Public Member Functions

- [Polynomial](#) ([BString](#) [transferType](#)= [BString](#)(), [BString](#) [approximationType](#)= [BString](#)(), [BString](#) [validFrequencyUnits](#)= [BString](#)(), [BFloat64](#) [frequencyLowerBound](#)=0, [BFloat64](#) [frequencyUpperBound](#)=0, [BFloat64](#) [approximationLowerBound](#)=0, [BFloat64](#) [approximationUpperBound](#)=0, [BFloat64](#) [maximumError](#)=0, [BArray](#)< [PolynomialEntry](#) > [coefficients](#)= [BArray](#)< [PolynomialEntry](#) >())

Public Attributes

- [BString](#) [transferType](#)
Transfer Type.
- [BString](#) [approximationType](#)
Polynomial Approximation Type.
- [BString](#) [validFrequencyUnits](#)
Valid Frequency Units.
- [BFloat64](#) [frequencyLowerBound](#)
Lower Valid Frequency Bound.
- [BFloat64](#) [frequencyUpperBound](#)
Upper Valid Frequency Bound.
- [BFloat64](#) [approximationLowerBound](#)
Lower Bound of Approximation.
- [BFloat64](#) [approximationUpperBound](#)
Upper Bound of Approximation.
- [BFloat64](#) [maximumError](#)
Maximum Absolute Error.
- [BArray](#)< [PolynomialEntry](#) > [coefficients](#)
The Coefficients.

7.79.1 Detailed Description

This class defines an [Polynomial](#) response table.

This has an array of the polynomial coefficients.

7.79.2 Constructor & Destructor Documentation

7.79.2.1 Polynomial()

```
Bds::Polynomial::Polynomial (
    BString transferType = BString(),
    BString approximationType = BString(),
    BString validFrequencyUnits = BString(),
    BFloat64 frequencyLowerBound = 0,
    BFloat64 frequencyUpperBound = 0,
    BFloat64 approximationLowerBound = 0,
    BFloat64 approximationUpperBound = 0,
    BFloat64 maximumError = 0,
    BArray< PolynomialEntry > coefficients = BArray<PolynomialEntry >() )
```

7.79.3 Member Data Documentation

7.79.3.1 approximationLowerBound

```
BFloat64 Bds::Polynomial::approximationLowerBound
```

Lower Bound of Approximation.

7.79.3.2 approximationType

```
BString Bds::Polynomial::approximationType
```

[Polynomial](#) Approximation Type.

7.79.3.3 approximationUpperBound

```
BFloat64 Bds::Polynomial::approximationUpperBound
```

Upper Bound of Approximation.

7.79.3.4 coefficients

BArray<[PolynomialEntry](#) > Bds::Polynomial::coefficients

The Coefficients.

7.79.3.5 frequencyLowerBound

BFloat64 Bds::Polynomial::frequencyLowerBound

Lower Valid Frequency Bound.

7.79.3.6 frequencyUpperBound

BFloat64 Bds::Polynomial::frequencyUpperBound

Upper Valid Frequency Bound.

7.79.3.7 maximumError

BFloat64 Bds::Polynomial::maximumError

Maximum Absolute Error.

7.79.3.8 transferType

BString Bds::Polynomial::transferType

Transfer Type.

7.79.3.9 validFrequencyUnits

BString Bds::Polynomial::validFrequencyUnits

Valid Frequency Units.

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

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

7.80 Bds::PolynomialEntry Class Reference

This class defines an entry in a [Polynomial](#) coefficient table.

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

Public Member Functions

- [PolynomialEntry](#) ([BFloat64](#) *coefficient*=0, [BFloat64](#) *plusError*=0, [BFloat64](#) *minusError*=0, [BString](#) *measurementMethod*= [BString](#)())

Public Attributes

- [BFloat64](#) *coefficient*
The coefficient.
- [BFloat64](#) *plusError*
Plus uncertainty or error in measured value.
- [BFloat64](#) *minusError*
Minus uncertainty or error in measured value.
- [BString](#) *measurementMethod*
The measurement method.

7.80.1 Detailed Description

This class defines an entry in a [Polynomial](#) coefficient table.

7.80.2 Constructor & Destructor Documentation

7.80.2.1 PolynomialEntry()

```
Bds::PolynomialEntry::PolynomialEntry (
    BFloat64 coefficient = 0,
    BFloat64 plusError = 0,
    BFloat64 minusError = 0,
    BString measurementMethod = BString() )
```

7.80.3 Member Data Documentation

7.80.3.1 coefficient

BFloat64 Bds::PolynomialEntry::coefficient

The coefficient.

7.80.3.2 measurementMethod

BString Bds::PolynomialEntry::measurementMethod

The measurement method.

7.80.3.3 minusError

BFloat64 Bds::PolynomialEntry::minusError

Minus uncertainty or error in measured value.

7.80.3.4 plusError

BFloat64 Bds::PolynomialEntry::plusError

Plus uncertainty or error in measured value.

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

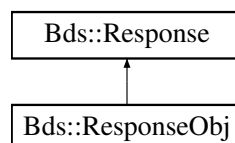
- [BdsD.h](#)
- [BdsD.cc](#)

7.81 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**(),
BString source= **BString**(), **BUInt32** stage=0, **BString** name= **BString**(), **BString** type= **BString**(),
PoleZero poleZeros=**PoleZero**(), **BArray**< **Fap** > faps= **BArray**< **Fap** >(), **Fir** fir=**Fir**(), **Polynomial**
polynomial=**Polynomial**(), **BFloat64** gain=0, **BFloat64** gainFrequency=0, **BString** stageType= **BString**(),
BFloat64 decimation=0, **BFloat64** decimationOffset=0, **BFloat64** decimationDelay=0, **BFloat64**
decimationCorr=0, **BString** symmetry= **BString**(), **BString** description= **BString**(), **BInt32** measured=0,
BFloat64 sampleRate=0, **BString** inputUnits= **BString**(), **BString** inputUnitsDesc= **BString**(), **BString**
outputUnits= **BString**(), **BString** outputUnitsDesc= **BString**())

Public Attributes

- **BUInt32** id
The ID.
- **BTimeStamp** startTime
The Start Time.
- **BTimeStamp** endTime
The End Time the channel was available.
- **BString** network
The Network/Organisation Name.
- **BString** station
The station.
- **BString** channel
The channel.
- **BString** source
The source.
- **BUInt32** stage
The stage (0, 1, 2, 3, ...)
- **BString** name
The response name. (Anti-Aliasing filter, [Digitiser](#), post filter etc)
- **BString** type
The type of response ([PoleZero](#), FIR, FAP, [Polynomial](#) or blank if no frequency response)
- **PoleZero** poleZeros
The PoleZeros.
- **BArray**< **Fap** > faps
The FAP Frequency/Amplitude/Phase table.
- **Fir** fir
The FIR filters coefficients.
- **Polynomial** polynomial
The polynomial response.
- **BFloat64** gain
Overall gain at gainFrequency.
- **BFloat64** gainFrequency
Frequency that gain is valid for.
- **BString** stageType
The stage type: A - Analog (rad/sec), B - Analog (Hz), C - Composite, D - Digital.
- **BFloat64** decimation
Decimation performed post filter.
- **BFloat64** decimationOffset

- Decimation sample offset.*
 - **BFloat64** [decimationDelay](#)
- Decimation delay.*
 - **BFloat64** [decimationCorr](#)
- Decimation correction performed.*
 - **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.*
 - **BString** [inputUnits](#)
- The units of the data as input from the perspective of data acquisition. After correcting data for this response, these would be the resulting units.*
 - **BString** [inputUnitsDesc](#)
- The input units description.*
 - **BString** [outputUnits](#)
- The units of the data as output from the perspective of data acquisition. These would be the units of the data prior to correcting for this response.*
 - **BString** [outputUnitsDesc](#)
- The output units description.*

7.81.1 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.

7.81.2 Constructor & Destructor Documentation

7.81.2.1 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(),
    Polynomial polynomial = Polynomial(),
    BFloat64 gain = 0,
    BFloat64 gainFrequency = 0,
    BString stageType = BString(),
    BFloat64 decimation = 0,
    BFloat64 decimationOffset = 0,
    BFloat64 decimationDelay = 0,
    BFloat64 decimationCorr = 0,
    BString symmetry = BString(),
    BString description = BString(),
    BInt32 measured = 0,
    BFloat64 sampleRate = 0,
    BString inputUnits = BString(),
    BString inputUnitsDesc = BString(),
    BString outputUnits = BString(),
    BString outputUnitsDesc = BString() )

```

7.81.3 Member Data Documentation

7.81.3.1 channel

BString Bds::Response::channel

The channel.

7.81.3.2 decimation

BFloat64 Bds::Response::decimation

Decimation performed post filter.

7.81.3.3 decimationCorr

BFloat64 Bds::Response::decimationCorr

Decimation correction performed.

7.81.3.4 decimationDelay

BFloat64 Bds::Response::decimationDelay

Decimation delay.

7.81.3.5 decimationOffset

BFloat64 Bds::Response::decimationOffset

Decimation sample offset.

7.81.3.6 description

BString Bds::Response::description

Misc description.

7.81.3.7 endTime

BTimeStamp Bds::Response::endTime

The End Time the channel was available.

7.81.3.8 faps

BArray<[Fap](#)> Bds::Response::faps

The FAP Frequency/Amplitude/Phase table.

7.81.3.9 fir

[Fir](#) Bds::Response::fir

The FIR filters coefficients.

7.81.3.10 gain

BFloat64 Bds::Response::gain

Overall gain at gainFrequency.

7.81.3.11 gainFrequency

BFloat64 Bds::Response::gainFrequency

Frequency that gain is valid for.

7.81.3.12 id

BUInt32 Bds::Response::id

The ID.

7.81.3.13 inputUnits

BString Bds::Response::inputUnits

The units of the data as input from the perspective of data acquisition. After correcting data for this response, these would be the resulting units.

7.81.3.14 inputUnitsDesc

BString Bds::Response::inputUnitsDesc

The input units description.

7.81.3.15 measured

BInt32 Bds::Response::measured

If response was a measured response.

7.81.3.16 name

BString Bds::Response::name

The response name. (Anti-Aliasing filter, [Digitiser](#), post filter etc)

7.81.3.17 network

BString Bds::Response::network

The Network/Organisation Name.

7.81.3.18 outputUnits

BString Bds::Response::outputUnits

The units of the data as output from the perspective of data acquisition. These would be the units of the data prior to correcting for this response.

7.81.3.19 outputUnitsDesc

BString Bds::Response::outputUnitsDesc

The output units description.

7.81.3.20 poleZeros

[PoleZero](#) Bds::Response::poleZeros

The PoleZeros.

7.81.3.21 polynomial

[Polynomial](#) Bds::Response::polynomial

The polynomial response.

7.81.3.22 sampleRate

BFloat64 Bds::Response::sampleRate

The stage's sample rate.

7.81.3.23 source

BString Bds::Response::source

The source.

7.81.3.24 stage

BUInt32 Bds::Response::stage

The stage (0, 1, 2, 3, ...)

7.81.3.25 stageType

BString Bds::Response::stageType

The stage type: A - Analog (rad/sec), B - Analog (Hz), C - Composite, D - Digital.

7.81.3.26 startTime

BTimeStamp Bds::Response::startTime

The Start Time.

7.81.3.27 station

BString Bds::Response::station

The station.

7.81.3.28 symmetry

BString Bds::Response::symmetry

Symmetry for FIR coefficients (A = asymmetric, B = symmetric[odd], C = symmetric[even])

7.81.3.29 type

BString Bds::Response::type

The type of response ([PoleZero](#), FIR, FAP, [Polynomial](#) or blank if no frequency response)

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

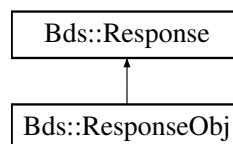
- [BdsD.h](#)
- [BdsD.cc](#)

7.82 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

7.82.1 Detailed Description

[Response](#) object adding string conversion.

7.82.2 Constructor & Destructor Documentation

7.82.2.1 ResponseObj()

```
Bds::ResponseObj::ResponseObj (
    const Response & response )
```

7.82.2.2 ~ResponseObj()

```
Bds::ResponseObj::~~ResponseObj ( )
```

7.82.3 Member Function Documentation

7.82.3.1 getString()

```
BString Bds::ResponseObj::getString ( )
```

7.82.3.2 setString()

```
void Bds::ResponseObj::setString (
    BString str )
```

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

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

7.83 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(), BString array= BString(), BUInt32 eventId=0, BString name= BString(), LocationSelect locationSelect=LocationSelect(), BString dataTypes= BString(), BString excludeChannels= 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.
- **BString array**
Channels are based on the given array (for array offsets)
- **BUInt32 eventId**
A particular event's ID.
- **BString name**
Match the name, title or some other string in the objects to select.
- **LocationSelect locationSelect**
Which locations to select, those for stations, channels or all of them when used in a locationGetList() call.
- **BString dataTypes**
Place holder as yet for: A comma separated list of Channel dataTypes to select when returning Channels. If set to null all Channel dataTypes are returned.
- **BString excludeChannels**
Comma separated wildcard list of channels to be excluded.

7.83.1 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.

7.83.2 Constructor & Destructor Documentation

7.83.2.1 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(),
    BString array = BString(),
    BUInt32 eventId = 0,
    BString name = BString(),
    LocationSelect locationSelect = LocationSelect(),
    BString dataTypes = BString(),
    BString excludeChannels = BString() )
```

7.83.3 Member Data Documentation

7.83.3.1 array

```
BString Bds::Selection::array
```

Channels are based on the given array (for array offsets)

7.83.3.2 calibrationName

BString Bds::Selection::calibrationName

Calibration name to use.

7.83.3.3 channelId

BUInt32 Bds::Selection::channelId

The Channel id.

7.83.3.4 channels

BList<SelectionChannel > Bds::Selection::channels

The data channels to select.

7.83.3.5 completeSegments

BInt32 Bds::Selection::completeSegments

Do not clip the segment times to match the required time period.

7.83.3.6 dataTypes

BString Bds::Selection::dataTypes

Place holder as yet for: A comma separated list of Channel dataTypes to select when returning Channels. If set to null all Channel dataTypes are returned.

7.83.3.7 digitiserId

BUInt32 Bds::Selection::digitiserId

The Digitiser id.

7.83.3.8 endTime

BTimeStamp Bds::Selection::endTime

The End Time.

7.83.3.9 eventId

BUInt32 Bds::Selection::eventId

A particular event's ID.

7.83.3.10 excludeChannels

BString Bds::Selection::excludeChannels

Comma separated wildcard list of channels to be excluded.

7.83.3.11 id

BUInt32 Bds::Selection::id

The ID of the record to return.

7.83.3.12 locationSelect

LocationSelect Bds::Selection::locationSelect

Which locations to select, those for stations, channels or all of them when used in a locationGetList() call.

7.83.3.13 name

BString Bds::Selection::name

Match the name, title or some other string in the objects to select.

7.83.3.14 range

[ListRange](#) Bds::Selection::range

The range of data to return.

7.83.3.15 sensorId

BUInt32 Bds::Selection::sensorId

The [Sensor](#) id.

7.83.3.16 sensorOldId

BUInt32 Bds::Selection::sensorOldId

The [Sensor](#) old id.

7.83.3.17 startTime

BTimeStamp Bds::Selection::startTime

The Start Time.

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

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

7.84 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](#)

7.84.1 Detailed Description

This class defines a channel for selection.

It contains the network:station:channel:source names.

7.84.2 Constructor & Destructor Documentation

7.84.2.1 SelectionChannel()

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

7.84.3 Member Data Documentation

7.84.3.1 channel

```
BString Bds::SelectionChannel::channel
```

7.84.3.2 network

```
BString Bds::SelectionChannel::network
```

7.84.3.3 source

```
BString Bds::SelectionChannel::source
```

7.84.3.4 station

```
BString Bds::SelectionChannel::station
```

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

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

7.85 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.

7.85.1 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.

7.85.2 Constructor & Destructor Documentation

7.85.2.1 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 )
```

7.85.3 Member Data Documentation

7.85.3.1 arrays

```
BList< BString > Bds::SelectionInfo::arrays
```

The list of Array names.

7.85.3.2 arraysAndStations

```
BList< BString > Bds::SelectionInfo::arraysAndStations
```

The list of Array and [Station](#) names.

7.85.3.3 channels

```
BList< BString > Bds::SelectionInfo::channels
```

The list of Channels.

7.85.3.4 endTime

BTimeStamp Bds::SelectionInfo::endTime

The End Time.

7.85.3.5 networks

BList< BString > Bds::SelectionInfo::networks

The list of [Network](#) Names.

7.85.3.6 numDataChannels

BUInt32 Bds::SelectionInfo::numDataChannels

The number of sets of data in the system matching the criteria.

7.85.3.7 sources

BList< BString > Bds::SelectionInfo::sources

The list of Data Sources.

7.85.3.8 startTime

BTimeStamp Bds::SelectionInfo::startTime

The Start Time.

7.85.3.9 stations

BList< BString > Bds::SelectionInfo::stations

The list of [Station](#) names.

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

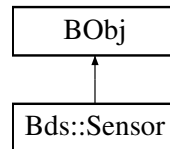
- [BdsD.h](#)
- [BdsD.cc](#)

7.86 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= [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)
- [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.

7.86.1 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.

7.86.2 Constructor & Destructor Documentation

7.86.2.1 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 )
```

7.86.3 Member Function Documentation

7.86.3.1 getMember()

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

Reimplemented from **BObj**.

7.86.3.2 getMembers()

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

Reimplemented from **BObj**.

7.86.3.3 getType()

```
BString Bds::Sensor::getType ( )
```

7.86.3.4 setMember()

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

Reimplemented from **BObj**.

7.86.3.5 setMembers()

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

Reimplemented from **BObj**.

7.86.4 Member Data Documentation

7.86.4.1 endTime

```
BTimeStamp Bds::Sensor::endTime
```

The End Time.

7.86.4.2 gain

```
BFloat64 Bds::Sensor::gain
```

The overall gain of the sensor at the manufacturers calibration frequency. (For information only)

7.86.4.3 gainUnits

```
BString Bds::Sensor::gainUnits
```

The gain units.

7.86.4.4 id

```
BUInt32 Bds::Sensor::id
```

The ID.

7.86.4.5 name

```
BString Bds::Sensor::name
```

The Sensors name.

7.86.4.6 numberChannels

```
BUInt32 Bds::Sensor::numberChannels
```

The number of supported channels.

7.86.4.7 oldId

```
BUInt32 Bds::Sensor::oldId
```

The Id from the old Autodrm database.

7.86.4.8 serialNumber

```
BString Bds::Sensor::serialNumber
```

The sensor's serial number. Only used when there is a unique physical sensor.

7.86.4.9 shared

```
BInt32 Bds::Sensor::shared
```

This sensor is shared.

7.86.4.10 startTime

BTimeStamp Bds::Sensor::startTime

The Start Time.

7.86.4.11 type

BString Bds::Sensor::type

The type of sensor. (Seismometer, Hydrophone etc)

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

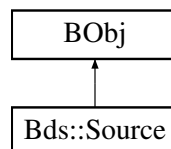
- [BdsD.h](#)
- [BdsD.cc](#)

7.87 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.

7.87.1 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.

7.87.2 Constructor & Destructor Documentation

7.87.2.1 Source()

```
Bds::Source::Source (
    BUInt32 id = 0,
    BString source = BString(),
    BString sourceMeta = BString(),
    BString alias = BString(),
    BString description = BString() )
```

7.87.3 Member Function Documentation

7.87.3.1 getMember()

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

Reimplemented from **BObj**.

7.87.3.2 getMembers()

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

Reimplemented from **BObj**.

7.87.3.3 getType()

```
BString Bds::Source::getType ( )
```

7.87.3.4 setMember()

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

Reimplemented from **BObj**.

7.87.3.5 setMembers()

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

Reimplemented from **BObj**.

7.87.4 Member Data Documentation

7.87.4.1 alias

```
BString Bds::Source::alias
```

The short alias for data files.

7.87.4.2 description

BString Bds::Source::description

The description.

7.87.4.3 id

BUInt32 Bds::Source::id

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

7.87.4.4 source

BString Bds::Source::source

The sensor data's source name.

7.87.4.5 sourceMeta

BString Bds::Source::sourceMeta

The associated metadata's source name.

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

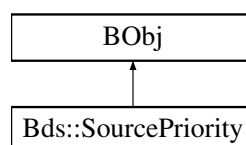
- [BdsD.h](#)
- [BdsD.cc](#)

7.88 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.

7.88.1 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.

7.88.2 Constructor & Destructor Documentation

7.88.2.1 SourcePriority()

```
Bds::SourcePriority::SourcePriority (
    BUInt32 id = 0,
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp(),
    BString source = BString(),
    BUInt32 priority = 0 )
```

7.88.3 Member Function Documentation

7.88.3.1 getMember()

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

Reimplemented from **BObj**.

7.88.3.2 getMembers()

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

Reimplemented from **BObj**.

7.88.3.3 getType()

```
BString Bds::SourcePriority::getType ( )
```

7.88.3.4 setMember()

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

Reimplemented from **BObj**.

7.88.3.5 setMembers()

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

Reimplemented from **BObj**.

7.88.4 Member Data Documentation

7.88.4.1 endTime

`BTimeStamp Bds::SourcePriority::endTime`

The End Time the channel was available.

7.88.4.2 id

`BUInt32 Bds::SourcePriority::id`

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

7.88.4.3 priority

`BUInt32 Bds::SourcePriority::priority`

The priority order, highest first.

7.88.4.4 source

`BString Bds::SourcePriority::source`

The source name.

7.88.4.5 startTime

`BTimeStamp Bds::SourcePriority::startTime`

The Start Time.

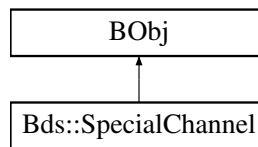
The documentation for this class was generated from the following files:

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

7.89 Bds::SpecialChannel Class Reference

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

Inheritance diagram for Bds::SpecialChannel:



Public Member Functions

- [SpecialChannel](#) ([BUInt32](#) id=0, [BTimeStamp](#) startTime= [BTimeStamp](#)(), [BTimeStamp](#) endTime= [BTimeStamp](#)(), [BString](#) network= [BString](#)(), [BString](#) station= [BString](#)(), [BString](#) channel= [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, wildcards allowed.
- [BString](#) [station](#)
The Stations name, wildcards allowed.
- [BString](#) [channel](#)
The channels name, wildcards allowed (often as <channelType>_<channelAux>)
- [BString](#) [dataType](#)
The Type of channel (ignore)
- [BString](#) [description](#)
The channels description.

7.89.1 Constructor & Destructor Documentation

7.89.1.1 SpecialChannel()

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

7.89.2 Member Function Documentation

7.89.2.1 getMember()

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

Reimplemented from **BObj**.

7.89.2.2 getMembers()

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

Reimplemented from **BObj**.

7.89.2.3 getType()

```
BString Bds::SpecialChannel::getType ( )
```

7.89.2.4 setMember()

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

Reimplemented from **BObj**.

7.89.2.5 setMembers()

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

Reimplemented from **BObj**.

7.89.3 Member Data Documentation

7.89.3.1 channel

```
BString Bds::SpecialChannel::channel
```

The channels name, wildcards allowed (often as <channelType>_<channelAux>)

7.89.3.2 dataType

```
BString Bds::SpecialChannel::dataType
```

The Type of channel (ignore)

7.89.3.3 description

```
BString Bds::SpecialChannel::description
```

The channels description.

7.89.3.4 endTime

```
BTimeStamp Bds::SpecialChannel::endTime
```

The End Time the channel was available.

7.89.3.5 id

```
BUInt32 Bds::SpecialChannel::id
```

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

7.89.3.6 network

BString Bds::SpecialChannel::network

The [Network](#) Name, wildcards allowed.

7.89.3.7 startTime

BTimeStamp Bds::SpecialChannel::startTime

The Start Time.

7.89.3.8 station

BString Bds::SpecialChannel::station

The Stations name, wildcards allowed.

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

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

7.90 Bds::Station Class Reference

This class defines a seismic station.

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

Public Member Functions

- [Station](#) ([BUInt32](#) id=0, [BString](#) network= [BString](#)(), [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](#) network
The [Network](#) this station belongs to if for a partricular network.
- [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.

7.90.1 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](#)

7.90.2 Constructor & Destructor Documentation

7.90.2.1 Station()

```
Bds::Station::Station (
    BUInt32 id = 0,
    BString network = BString(),
    BString name = BString(),
    BString alias = BString(),
    BString type = BString(),
    BString description = BString(),
    BList< ArrayChannel > channels = BList<ArrayChannel >() )
```

7.90.3 Member Data Documentation

7.90.3.1 alias

```
BString Bds::Station::alias
```

Alias name to be returned to the user.

7.90.3.2 channels

```
BList<ArrayChannel > Bds::Station::channels
```

List of channels if an Array.

7.90.3.3 description

```
BString Bds::Station::description
```

Description.

7.90.3.4 id

```
BUInt32 Bds::Station::id
```

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

7.90.3.5 name

```
BString Bds::Station::name
```

The name.

7.90.3.6 network

```
BString Bds::Station::network
```

The [Network](#) this station belongs to if for a particular network.

7.90.3.7 type

```
BString Bds::Station::type
```

The [Station](#) type. Set to "array" or "station".

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

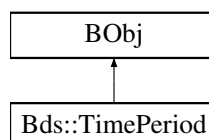
- [BdsD.h](#)
- [BdsD.cc](#)

7.91 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.

7.91.1 Detailed Description

This class defines a [TimePeriod](#).

It has [startTime](#) and [endTime](#) fields. [Note](#) the [endTime](#) is not included in the period.

7.91.2 Constructor & Destructor Documentation

7.91.2.1 TimePeriod()

```
Bds::TimePeriod::TimePeriod (
    BTimeStamp startTime = BTimeStamp(),
    BTimeStamp endTime = BTimeStamp() )
```

7.91.3 Member Function Documentation

7.91.3.1 getMember()

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

Reimplemented from [BObj](#).

7.91.3.2 getMembers()

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

Reimplemented from **BObj**.

7.91.3.3 getType()

```
BString Bds::TimePeriod::getType ( )
```

7.91.3.4 setMember()

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

Reimplemented from **BObj**.

7.91.3.5 setMembers()

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

Reimplemented from **BObj**.

7.91.4 Member Data Documentation

7.91.4.1 endTime

```
BTimeStamp Bds::TimePeriod::endTime
```

The End time to the nearest us.

7.91.4.2 startTime

BTimeStamp Bds::TimePeriod::startTime

The Start time to the nearest us.

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

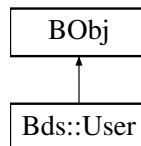
- [BdsD.h](#)
- [BdsD.cc](#)

7.92 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.

7.92.1 Detailed Description

This holds information on a user.

All information on a BDS users is stored along with the security groups they belong to.

7.92.2 Constructor & Destructor Documentation

7.92.2.1 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 >() )
```

7.92.3 Member Function Documentation

7.92.3.1 getMember()

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

Reimplemented from **BObj**.

7.92.3.2 getMembers()

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

Reimplemented from **BObj**.

7.92.3.3 getType()

```
BString Bds::User::getType ( )
```

7.92.3.4 setMember()

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

Reimplemented from **BObj**.

7.92.3.5 setMembers()

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

Reimplemented from **BObj**.

7.92.4 Member Data Documentation

7.92.4.1 address

```
BString Bds::User::address
```

The Users postal address.

7.92.4.2 email

```
BString Bds::User::email
```

The users email Address.

7.92.4.3 enabled

```
BInt32 Bds::User::enabled
```

Whether the users account is enabled.

7.92.4.4 groups

```
BList< BString > Bds::User::groups
```

The security groups the user belongs to.

7.92.4.5 id

```
BUInt32 Bds::User::id
```

The unique user ID.

7.92.4.6 name

```
BString Bds::User::name
```

The Users full name.

7.92.4.7 password

```
BString Bds::User::password
```

The Users password.

7.92.4.8 telephone

```
BString Bds::User::telephone
```

The Users telephone number.

7.92.4.9 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

- **Error** [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 converters to order blocks by time.
- class [Bds::DataFileOptions](#)
This defines a list of BDS data converter options.
- class [Bds::DataFile](#)
This class defines the interface for generic data file access that all of the BDS data converters 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 ntohll

```
#define ntohll(  
    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/BdsDataFileIdc.cpp File Reference

```
#include <BdsDataFileIdc.h>
#include <errno.h>
```

Namespaces

- [Bds](#)

Macros

- #define [LDEBUG](#) 0
- #define [dprintf](#)(fmt, a...)

8.34.1 Macro Definition Documentation

8.34.1.1 dprintf

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

8.34.1.2 LDEBUG

```
#define LDEBUG 0
```

8.35 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.d File Reference

8.36 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.h File Reference

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

Classes

- class [Bds::DataFileIdc](#)

This class defines the interface for IDC response data file access.

Namespaces

- [Bds](#)

8.37 /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)
- static **BString** [Bds::unitsCode](#) (Response &r)

Variables

- static char [Bds::cm6Table](#) [64]
- static **BUInt8** [Bds::cm6TableRev](#) [128]

8.38 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.d File Reference

8.39 /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](#)

8.40 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp File Reference

```
#include <BdsDataFileLac.h>
#include <BEndian.h>
#include <BDebug.h>
#include <errno.h>
```

Namespaces

- [Bds](#)

8.41 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.d File Reference

8.42 /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](#)

8.43 /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)

8.44 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.d File Reference

8.45 /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](#)

8.46 /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...)

8.46.1 Macro Definition Documentation

8.46.1.1 dprintf

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

8.46.1.2 LDEBUG

```
#define LDEBUG 0
```

8.47 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.d File Reference

8.48 /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](#)

8.49 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp File Reference

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

Namespaces

- [Bds](#)

Macros

- `#define` [BDEBUGL1](#) 0

8.49.1 Macro Definition Documentation

8.49.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.50 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.d File Reference

8.51 /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](#)

8.52 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.cpp File Reference

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

Namespaces

- [Bds](#)

Macros

- `#define` [BDEBUGL1](#) 0
- `#define` [BDEBUGL2](#) 0

Variables

- `const char *` [Bds::node_types](#) []

8.52.1 Macro Definition Documentation

8.52.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.52.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.53 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.d File Reference

8.54 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.h File Reference

```
#include <BdsDataFile.h>
#include <BdsD.h>
#include <pugixml.hpp>
```

Classes

- class [Bds::DataFileStationXml](#)

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

Namespaces

- [Bds](#)

8.55 /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

8.56 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.d File Reference

8.57 /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](#)

8.58 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp File Reference

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

Namespaces

- [Bds](#)

8.59 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.d File Reference

8.60 /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](#)

8.61 /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)

8.61.1 Function Documentation

8.61.1.1 [parseStringFixedFields\(\)](#)

```
static BList< BString> parseStringFixedFields (
    BString s,
    int * fieldWidths ) [static]
```

8.62 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.d File Reference

8.63 /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](#)

8.64 /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>
#include <BdsDataFileStationXml.h>
#include <BdsDataFileIdc.h>
```

Namespaces

- [Bds](#)

Functions

- **BString** [Bds::fixedWidthValue](#) (double v, int width)
This returns a double as a fixed width string truncating the data.

Variables

- DataFormatAll [Bds::dataFormatAll](#)

8.65 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.d File Reference

8.66 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h File Reference

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

Classes

- class [Bds::DataFormatAll](#)
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.

8.67 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp File Reference

```
#include <BdsDataFileSeed.h>
#include <BEndian.h>
#include <errno.h>
#include <BDebug.h>
#include <libmseed.h>
```

Namespaces

- [Bds](#)

Macros

- #define [BDEBUGL1](#) 0
- #define [BDEBUGL2](#) 0
- #define [BDEBUGL3](#) 0
- #define [DEBUG](#) 0
- #define [DEBUG_BLOCKETTE](#) 0
- #define [DEBUG_BLOCKS](#) 0
- #define [FILL_BLOCKS](#) 1
- #define [ROUND_TIMESTAMPS_US](#) 10

Functions

- static double [Bds::roundDigits](#) (double v, int nDigits)
- 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 **BTimeStamp** [Bds::fromSeedTimeString](#) (**BString** str)
- 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)

8.67.1 Macro Definition Documentation

8.67.1.1 BDEBUGL1

```
#define BDEBUGL1 0
```

8.67.1.2 BDEBUGL2

```
#define BDEBUGL2 0
```

8.67.1.3 BDEBUGL3

```
#define BDEBUGL3 0
```

8.67.1.4 DEBUG

```
#define DEBUG 0
```

8.67.1.5 DEBUG_BLOCKETTE

```
#define DEBUG_BLOCKETTE 0
```

8.67.1.6 DEBUG_BLOCKS

```
#define DEBUG_BLOCKS 0
```

8.67.1.7 FILL_BLOCKS

```
#define FILL_BLOCKS 1
```

8.67.1.8 ROUND_TIMESTAMPS_US

```
#define ROUND_TIMESTAMPS_US 10
```

8.68 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.d File Reference

8.69 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h File Reference

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

Classes

- class [Bds::DataFileSeed](#)
Data file convertor for SEED file formats.

Namespaces

- [Bds](#)

Typedefs

- typedef struct MSRecord_s [MSRecord](#)

8.69.1 Typedef Documentation

8.69.1.1 MSRecord

```
typedef struct MSRecord_s MSRecord
```

8.70 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp File Reference

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

Namespaces

- [Bds](#)

8.71 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.d File Reference

8.72 /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](#)

8.73 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.cpp File Reference

```
#include <BdsSeedTypes.h>
```

Namespaces

- [Bds](#)

8.74 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.d File Reference

8.75 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.idl File Reference

8.76 /src/blacknest/bds/bds/bdsDataLib/canada_compress.d File Reference

8.77 /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.

8.77.1 Macro Definition Documentation

8.77.1.1 CANCOMP_CORRUPT

```
#define CANCOMP_CORRUPT 2 /* corrupted call */
```

8.77.1.2 CANCOMP_ERR

```
#define CANCOMP_ERR -1 /* unrecoverable error (malloc fails) */
```

8.77.1.3 CANCOMP_EXCEED

```
#define CANCOMP_EXCEED
```

Value:

```
3      /* number of bytes available in compressed
data exceeded during decompression */
```

8.77.1.4 CANCOMP_NOT_20

```
#define CANCOMP_NOT_20 1 /* number of samples not divisible by 20 */
```

8.77.1.5 CANCOMP_SUCCESS

```
#define CANCOMP_SUCCESS 0 /* success */
```

8.77.2 Function Documentation

8.77.2.1 canada_compress()

```
int canada_compress (
    unsigned char * b,
    unsigned long * y,
    int * n,
    int m,
    unsigned long * v0 )
```

Compresses Canada format seismic data.

8.77.2.2 canada_uncompress()

```
int canada_uncompress (
    unsigned char * b,
    uint32_t * y,
    int * n,
    int m,
    uint32_t * v0 )
```

De-compresses Canada format seismic data.

8.78 BdsC.cc File Reference

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

Namespaces

- [Bds](#)

8.79 BdsC.d File Reference

8.80 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

8.81 BdsD.cc File Reference

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

Namespaces

- [Bds](#)

8.82 BdsD.d File Reference

8.83 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::PolynomialEntry](#)
This class defines an entry in a [Polynomial](#) coefficient table.
- class [Bds::Polynomial](#)
This class defines an [Polynomial](#) 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::ChannelInstrument](#)

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::ChannelInfo](#)

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

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::DataBlockChannel](#)

This class provides the actual seismic data values contained within a single data block along with the network↔:station:channel:source information.
 - 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.

- class [Bds::SpecialChannel](#)

- class [Bds::Event](#)

This class defines a seismic event.

Namespaces

- [Bds](#)

Typedefs

- typedef **BList**< DataFormat > [Bds::DataFormats](#)

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.

- enum [Bds::DataFormatSet](#) {
[Bds::DataFormatSetNone](#) = 0x00 , [Bds::DataFormatSetMetadataRead](#) = 0x01 , [Bds::DataFormatSetMetadataWrite](#) = 0x02 , [Bds::DataFormatSetSensordataRead](#) = 0x04 ,
[Bds::DataFormatSetSensordataWrite](#) = 0x08 }
Data format abilities bitset.
- enum [Bds::LocationSelect](#) { [Bds::LocationSelectAll](#) , [Bds::LocationSelectStation](#) , [Bds::LocationSelectChannel](#) }
Which Locations to select.

8.83.1 Detailed Description

BOAP data class definitions for: [Bds](#).

Date

2022-06-23T07:17:59

The classes in here have been defined by a BOAP *.bidl file and define classes able to be communicated across a BOAP link

8.84 BdsLib.cpp File Reference

```
#include <BdsLib.h>
#include <math.h>
#include <complex>
```

Namespaces

- [Bds](#)

Functions

- **Error** [Bds::bdsLibInit](#) (DataAccess &bds)
Initialise the bdsLib with settings from the BdsServer.
- **Error** [Bds::bdsLibInit](#) (DataAddAccess &bds)
Initialise the bdsLib with settings from the BdsServer.
- **Error** [Bds::bdsLibInit](#) (AdminAccess &bds)
Initialise the bdsLib with settings from the BdsServer.
- 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.
- **Error** [Bds::bdsDataInfoSetTimeRange](#) (DataInfo &dataInfo)
Restricts the time tange of all of the [DataInfo](#)'s channels to match the [DataInfo](#)'s startTime/endTime fields.
- **Error** [Bds::bdsDataInfoFromInfo](#) (**BDictString** info, DataInfo &dataInfo, **Bool** append)

- Convert info to [DataInfo](#).

 - **Error** [Bds::bdsInfoFromDataInfo](#) (const [DataInfo](#) &dataInfo, **BDictString** &info)
Converts a [DataInfo](#) object into a [BDictString](#) list of named strings.
 - **Error** [Bds::bdsDataInfoFlatten](#) ([DataInfo](#) &dataInfo)
Flattens a [DataInfo](#) to 1 segment per channel for use in [dataOpen\(\)](#) calls.
 - **Error** [Bds::bdsDataInfoMergeFlatten](#) ([DataInfo](#) &dataInfo, const [DataInfo](#) &dataInfoAdd)
Merges a [DataInfo](#) into another flattening the segments to 1 for use in [dataOpen\(\)](#) calls.
 - **String** [Bds::bdsUnitsConvert](#) (**String** units)
Tidy up units name to standard SI units format.
 - static int [Bds::responseSort](#) ([Response](#) &a, [Response](#) &b)
 - **Error** [Bds::bdsMetadataImportFix](#) ([ChannelInfos](#) &channelInfos, **Bool** stageRenum)
Fix up [ChannelInfos](#) from import. Mainly making sure response stages and their units are correct.
 - **Error** [Bds::bdsMetadataExportFix](#) ([ChannelInfos](#) &channelInfos, **Bool** singleResponse, **Bool** stage←Renum, **Bool** changeUnits, **Bool** stageGains, **Bool** decimation, **Bool** toDisplacement, **Bool** toNm)
Fix up [ChannelInfos](#) for export. Mainly making sure response stages and their units are correct.
 - **String** [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::bdsDumpSelectionInfo](#) ([SelectionInfo](#) sel)
Debug print out a [SelectionInfo](#) 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::bdsDumpLocation](#) ([Location](#) location)
Debug printout location.
 - **String** [Bds::bdsDataChannelInfo](#) (const [DataChannel](#) &dataChannel)
Returns a string representation of a [DataChannel](#) object.
 - **String** [Bds::bdsDataChannelRef](#) (const [DataChannel](#) &dataChannel)
Returns the string reference name of a [DataChannel](#) object.
 - **String** [Bds::bdsDataChannelRef](#) (const [ChannelInfo](#) &channelInfo)
Returns the string reference name of a [ChannelInfo](#) object.
 - **Error** [Bds::bdsDataChannelOverallResponse](#) (const [ChannelInfo](#) &channelInfo, [Response](#) &response)
Returns the overal response from the list of responses in a [ChannelInfo](#).
 - **String** [Bds::bdsSelectionChannelInfo](#) (const [Selection](#) &selection, **UInt** channel)
Returns a string describing the name and time period of a selection channel.
 - double [Bds::bdsPoleZeroGain](#) (const [PoleZero](#) &poleZero, double frequency)
Calculates the overall gain of the given [PoleZero](#) transfer function.
 - void [Bds::bdsPoleZeroGainPhase](#) (const [PoleZero](#) &poleZero, double frequency, double &gain, double &phase)
Calculates the overall gain and phase of the given [PoleZero](#) transfer function.
 - void [Bds::bdsPoleZeroToFap](#) (const [PoleZero](#) &poleZero, **UInt** nPoints, double calibrationFrequency, double sampleFrequency, **Array**< [Fap](#) > &fap)
Convert [PoleZero](#) to FAP.
 - static **String** [Bds::fileNameTime](#) (**TimeStamp** t)
 - **String** [Bds::bdsFileNameExpand](#) (**String** fileName, [ChannelInfo](#) &channelInfo)
Default filename from a [ChannelInfo](#).
 - **String** [Bds::bdsFileNameExpand](#) (**String** fileName, [ChannelInfos](#) &channelInfos)

- Default filename from a list of [ChannelInfo](#)'s.
 - **BString** [Bds::bdsFileNameExpand](#) (**BString** fileName, Selection &sel)
 - Default filename from a [Selection](#).
 - void [Bds::bdsSpecialChannelsSet](#) (const **BList**< SpecialChannel > specialChannels)
 - Set the special channels list.
 - **BList**< SpecialChannel > [Bds::bdsSpecialChannels](#) ()
 - Return list of special channels.
 - **Bool** [Bds::bdsSpecialChannelIgnore](#) (**BString** network, **BString** station, **BString** channel)
 - Check if channel should be ignored.
 - char [Bds::seedChannelInstrumentCode](#) (**BString** dataType)
 - Returns SEED instrument code from dataType.
 - **BString** [Bds::seedChannelDataType](#) (**BString** channel)
 - Returns dataType from channel name based on SEED channel name convention.
 - BStringList [Bds::bdsDataTypes](#) ()
 - Returns all known data types.
 - BStringList [Bds::bdsUnits](#) ()
 - Returns all known SI units.
 - **BString** [Bds::bdsUnitCase](#) (**BString** unit)
 - Converts character case of units.

Variables

- static **BList**< SpecialChannel > [Bds::bdsSpecialChannelsList](#)
- SeedICodeToDataType [Bds::seedICodeToDataTypes](#) []

8.85 BdsLib.d File Reference

8.86 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.

8.87 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

- **Error** [Bds::bdsLibInit](#) (DataAccess &bds)
Initialise the bdsLib with settings from the BdsServer.
- **Error** [Bds::bdsLibInit](#) (DataAddAccess &bds)
Initialise the bdsLib with settings from the BdsServer.
- **Error** [Bds::bdsLibInit](#) (AdminAccess &bds)
Initialise the bdsLib with settings from the BdsServer.
- 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.
- **Error** [Bds::bdsDataInfoSetTimeRange](#) (DataInfo &dataInfo)
Restricts the time range of all of the [DataInfo](#)'s channels to match the [DataInfo](#)'s startTime/endTime fields.
- **Error** [Bds::bdsDataInfoFromInfo](#) (**BDictString** info, DataInfo &dataInfo, **Bool** append)
Convert info to [DataInfo](#).
- **Error** [Bds::bdsInfoFromDataInfo](#) (const DataInfo &dataInfo, **BDictString** &info)
Converts a [DataInfo](#) object into a [BDictString](#) list of named strings.
- **Error** [Bds::bdsDataInfoFlatten](#) (DataInfo &dataInfo)
Flattens a [DataInfo](#) to 1 segment per channel for use in [dataOpen\(\)](#) calls.
- **Error** [Bds::bdsDataInfoMergeFlatten](#) (DataInfo &dataInfo, const DataInfo &dataInfoAdd)
Merges a [DataInfo](#) into another flattening the segments to 1 for use in [dataOpen\(\)](#) calls.
- **BString** [Bds::bdsUnitsConvert](#) (**BString** units)
Tidy up units name to standard SI units format.
- **Error** [Bds::bdsMetadataImportFix](#) (ChannelInfos &channelInfos, **Bool** stageRenum)
Fix up [ChannelInfos](#) from import. Mainly making sure response stages and their units are correct.
- **Error** [Bds::bdsMetadataExportFix](#) (ChannelInfos &channelInfos, **Bool** singleResponse, **Bool** stage↔Renum, **Bool** changeUnits, **Bool** stageGains, **Bool** decimation, **Bool** toDisplacement, **Bool** toNm)
Fix up [ChannelInfos](#) for export. Mainly making sure response stages and their units are correct.
- **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.
- **Error** [Bds::bdsDataChannelOverallResponse](#) (const ChannelInfo &channelInfo, Response &response)

- Returns the overall 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.
 - **BString** [Bds::bdsFileNameExpand](#) (**BString** fileName, Selection &sel)

Default filename from a [Selection](#).
 - void [Bds::bdsSpecialChannelsSet](#) (const **BList**< SpecialChannel > specialChannels)

Set the special channels list.
 - **BList**< SpecialChannel > [Bds::bdsSpecialChannels](#) ()

Return list of special channels.
 - **Bool** [Bds::bdsSpecialChannelIgnore](#) (**BString** network, **BString** station, **BString** channel)

Check if channel should be ignored.
 - char [Bds::seedChannelInstrumentCode](#) (**BString** dataType)

Returns SEED instrument code from dataType.
 - **BString** [Bds::seedChannelDataType](#) (**BString** channel)

Returns dataType from channel name based on SEED channel name convention.
 - BStringList [Bds::bdsDataTypes](#) ()

Returns all known data types.
 - BStringList [Bds::bdsUnits](#) ()

Returns all known SI units.
 - **BString** [Bds::bdsUnitCase](#) (**BString** unit)

Converts character case of units.
 - double [Bds::bdsPoleZeroGain](#) (const PoleZero &poleZero, double frequency)

Calculates the overall gain of the given [PoleZero](#) transfer function.
 - void [Bds::bdsPoleZeroGainPhase](#) (const PoleZero &poleZero, double frequency, double &gain, double &phase)

Calculates the overall 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::bdsDumpSelectionInfo](#) (SelectionInfo sel)

Debug print out a [SelectionInfo](#) 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.
 - void [Bds::bdsDumpLocation](#) (Location location)

Debug printout location.

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.

8.87.1 Detailed Description

General BdsLib API functions.

8.88 BdsS.cc File Reference

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

Namespaces

- [Bds](#)

8.89 BdsS.d File Reference

8.90 BdsT.cc File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <BdsT.h>
#include <Control.h>
```

8.91 /src/blacknest/bds/bds/doc/bdsApiOverview.dox File Reference

Index

/src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp, 363
/src/blacknest/bds/bds/bdsDataLib/BdsCompress.d, 363
/src/blacknest/bds/bds/bdsDataLib/BdsCompress.h, 363
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp, 364
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.d, 364
/src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h, 364
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp, 365
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.d, 365
/src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h, 365
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp, 365
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.d, 366
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h, 366
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp, 366
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.d, 367
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h, 367
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp, 367
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.d, 367
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h, 367
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp, 368
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.d, 370
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h, 370
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp, 371
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.d, 371
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h, 371
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.cpp, 372
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.d, 374
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h, 374
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp, 375
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.d, 375
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h, 375
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp, 375
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.d, 376
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h, 376
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.cpp, 377
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.d, 377
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileIdc.h, 377
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileImms.cpp, 378
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileImms.d, 378
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileImms.h, 378
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp, 379
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.d, 379
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h, 379
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.cpp, 379
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.d, 380
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.h, 380
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp, 380
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.d, 381
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h, 381
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp, 381
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.d, 382
/src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.h, 382

/src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.cpp, 382
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.d, 383
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileStationXml.h, 383
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp, 384
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.d, 384
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h, 384
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp, 385
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.d, 385
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h, 385
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp, 386
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.d, 386
 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h, 386
 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp, 387
 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.d, 387
 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h, 387
 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp, 388
 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.d, 390
 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h, 390
 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp, 390
 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.d, 391
 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.h, 391
 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.cpp, 391
 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.d, 391
 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.h, 391
 /src/blacknest/bds/bds/bdsDataLib/canada_compress.d, 391
 /src/blacknest/bds/bds/bdsDataLib/canada_compress.h, 391
 /src/blacknest/bds/bds/doc/bdsApiOverview.dox, 403
 ~BdsDataPacket
 Bds::BdsDataPacket, 90
 ~DataCollate
 Bds::DataCollate, 187
 ~DataFile
 Bds::DataFile, 197
 ~DataFileBds
 Bds::DataFileBds, 214
 Bds::DataFileCssData, 226
 Bds::DataFileSeed, 254
 Bds::DataFormatAll, 270
 Bds::ResponseObj, 330
 Bds::Fir, 287
 AccessGroup
 Bds::AccessGroup, 52
 accessGroupDelete
 Bds::AdminAccess, 60
 accessGroupGetList
 Bds::AdminAccess, 60
 accessGroupUpdate
 Bds::AdminAccess, 60
 address
 Bds::User, 361
 addSource
 Bds::DataCollate, 187
 AdminAccess
 Bds::AdminAccess, 59
 alias
 Bds::Source, 346
 Bds::Station, 355
 ALLOW_TIMESTAMP_JITTER
 BdsDataFileBds.cpp, 369
 BdsDataFileCd.cpp, 373
 amplitude
 Bds::Fap, 286
 apiVersion
 Bds, 46
 appendDouble
 Bds::BdsSeedType, 97
 appendExp
 Bds::BdsSeedType, 97
 appendInt
 Bds::BdsSeedType, 97
 appendString
 Bds::BdsSeedType, 98
 appendStringVariable
 Bds::BdsSeedType, 98
 approximationLowerBound
 Bds::Polynomial, 318
 approximationType
 Bds::Polynomial, 318
 approximationUpperBound
 Bds::Polynomial, 318
 array
 Bds::ChannelInfos, 130
 Bds::DataInfo, 274
 Bds::Selection, 332
 ArrayChannel
 Bds::ArrayChannel, 83
 arrayOffsetEast
 Bds::ArrayChannel, 84

- Bds::Location, 298
- arrayOffsetNorth
 - Bds::ArrayChannel, 84
 - Bds::Location, 298
- arrays
 - Bds::SelectionInfo, 338
- arraysAndStations
 - Bds::SelectionInfo, 338
- auth
 - Bds::CdChannel_1v0, 106
 - Bds::CdPacketData, 112
- authKey
 - Bds::CdPacketData, 112
- authSize
 - Bds::CdPacketData, 113
- AvailFull
 - Bds, 29
- AvailNone
 - Bds, 29
- AvailPartial
 - Bds, 29
- AvailType
 - Bds, 28
- availType
 - Bds::DataAvail, 172
- b
 - Bds::Fir, 287
- baseSamplingFrequency
 - Bds::Digitiser, 278
- BDEBUGL1
 - BdsDataFileSac.cpp, 382
 - BdsDataFileSeed.cpp, 389
 - BdsDataFileStationXml.cpp, 383
- BDEBUGL2
 - BdsDataFileSeed.cpp, 389
 - BdsDataFileStationXml.cpp, 383
- BDEBUGL3
 - BdsDataFileSeed.cpp, 389
- Bds, 21
 - apiVersion, 46
 - AvailFull, 29
 - AvailNone, 29
 - AvailPartial, 29
 - AvailType, 28
 - bdsChannelGetName, 33
 - bdsChannelGetTypeAux, 34
 - bdsDataChannelInfo, 34
 - bdsDataChannelOverallResponse, 34
 - bdsDataChannelRef, 34
 - bdsDataFileSeedLogError, 35
 - bdsDataFileSeedLogWarning, 35
 - BdsDataFileVersion, 46
 - bdsDataInfoFlatten, 35
 - bdsDataInfoFromInfo, 35
 - bdsDataInfoMergeFlatten, 35
 - bdsDataInfoSetTimeRange, 35
 - BdsDataType, 29
 - BdsDataTypeBlock, 29
 - BdsDataTypeData, 29
 - BdsDataTypeInfo, 29
 - BdsDataTypeInfoExtra, 29
 - bdsDataTypes, 36
 - bdsDumpChannelInfos, 36
 - bdsDumpData, 36
 - bdsDumpDataInfo, 36
 - bdsDumpLocation, 36
 - bdsDumpPoleZeros, 37
 - bdsDumpSelection, 37
 - bdsDumpSelectionInfo, 37
 - bdsFileNameExpand, 37, 38
 - bdsInfoFromDataInfo, 38
 - bdsLibInit, 38
 - bdsMetadataExportFix, 39
 - bdsMetadataImportFix, 39
 - bdsPoleZeroGain, 39
 - bdsPoleZeroGainPhase, 39
 - bdsPoleZeroToFap, 39
 - bdsSelectionChannelInfo, 40
 - bdsSpecialChannelIgnore, 40
 - bdsSpecialChannels, 40
 - bdsSpecialChannelsList, 47
 - bdsSpecialChannelsSet, 40
 - bdsStationAlias, 40
 - bdsUnCompressCm8, 41
 - bdsUnCompressSteim1, 41
 - bdsUnitCase, 41
 - bdsUnits, 41
 - bdsUnitsConvert, 41
 - ChannelAuxLen, 47
 - ChannelTypeLen, 47
 - cm6Table, 47
 - cm6TableRev, 47
 - crc, 42
 - crc64, 42
 - crclnit, 42
 - crclnitDone, 48
 - crcVec, 48
 - dataCalculateDifference, 42
 - dataCalculateUnDifference, 42
 - dataChecksum, 42
 - dataCompressCm6, 43
 - dataConvert, 43
 - dataDeCompressCm6, 43
 - DataFlagClipDataToChannels, 29
 - DataFlagClipDataToTime, 29
 - DataFlagMergeSegments, 29
 - DataFlagNoMetadata, 29
 - DataFlagNone, 29
 - DataFlags, 29
 - dataFormatAll, 48
 - DataFormats, 28
 - DataFormatSet, 29
 - DataFormatSetMetadataRead, 30
 - DataFormatSetMetadataWrite, 30
 - DataFormatSetNone, 30
 - DataFormatSetSensordataRead, 30

- DataFormatSetSensordataWrite, 30
- duplicateDump, 43
- ErrorDataQuality, 30
- ErrorNoMetaData, 30
- Errors, 30
- ErrorSlaveMode, 30
- ErrorTimeStamp, 30
- ErrorValidate, 30
- ErrorValidateBdsFudge, 30
- ErrorValidateDuplicate, 30
- ErrorValidateFilenameTime, 30
- ErrorValidateFix, 30
- ErrorValidateMetaData, 30
- ErrorValidateMissingBlocks, 30
- ErrorValidateReorder, 30
- ErrorValidateTimeBackwards, 30
- FileHeaderType, 30
- FileHeaderType_Standard, 30
- FileHeaderType_TapeDigitiser, 30
- fileNameTime, 44
- FileSampleType, 31
- FileSampleType_Float32, 31
- FileSampleType_Float64, 31
- FileSampleType_Int16, 31
- FileSampleType_Int32, 31
- FileSampleType_Unknown, 31
- fixedString, 44
- fixedWidthValue, 44
- fromSeedTimeString, 44
- getHexString, 44
- LocationSelect, 31
- LocationSelectAll, 31
- LocationSelectChannel, 31
- LocationSelectStation, 31
- Mode, 31
- ModeMaster, 31
- ModeSlave, 31
- NetworkNameLen, 48
- node_types, 48
- nullString, 44
- Priority, 31
- PriorityHigh, 33
- PriorityLow, 33
- PriorityNormal, 33
- record_handler, 45
- removeCR, 45
- responseSort, 45
- roundDigits, 45
- SampleFormat, 33
- SampleFormatFloat32, 33
- SampleFormatFloat64, 33
- SampleFormatInt16, 33
- SampleFormatInt24, 33
- SampleFormatInt32, 33
- SampleFormatUnknown, 33
- Scale, 49
- seedChannelDataType, 45
- seedChannelInstrumentCode, 45
- seedCodeToDataTypes, 49
- seedTime, 46
- seedTimeString, 46
- SelectionGroup, 33
- SelectionGroupData, 33
- SelectionGroupDataWithCount, 33
- SelectionGroupMetaData, 33
- SourceLen, 49
- StationNameLen, 49
- stringFormat, 46
- unitsCode, 46
- Bds::AccessGroup, 51
 - AccessGroup, 52
 - endTime, 53
 - getMember, 52
 - getMembers, 52
 - getType, 52
 - group, 53
 - id, 53
 - network, 53
 - setMember, 52
 - setMembers, 53
 - startTime, 54
 - station, 54
- Bds::AdminAccess, 54
 - accessGroupDelete, 60
 - accessGroupGetList, 60
 - accessGroupUpdate, 60
 - AdminAccess, 59
 - calibrationDelete, 60
 - calibrationGetList, 60
 - calibrationUpdate, 61
 - changeDelete, 61
 - changeGetList, 61
 - changeGetListNumber, 61
 - changeGroupDelete, 61
 - changeGroupEnd, 62
 - changeGroupGetList, 62
 - changeGroupStart, 62
 - channelDelete, 62
 - channelGet, 62
 - channelGetList, 63
 - channelInstrumentDelete, 63
 - channelInstrumentGetList, 63
 - channelInstrumentUpdate, 63
 - channelUpdate, 63
 - clean, 64
 - connect, 64
 - dataAvailability, 64
 - databaseBackup, 64
 - databaseRestore, 64
 - dataChannelDelete, 65
 - dataChannelGetList, 65
 - dataChannelUpdate, 65
 - dataClose, 65
 - dataFileDelete, 65
 - dataFileGetList, 66
 - dataFileUpdate, 66

- dataFormatGetList, 66
- dataFormattedGetLength, 66
- dataFormattedRead, 66
- dataGetBlock, 67
- dataGetChannelInfo, 67
- dataGetInfo, 67
- dataGetNotes, 67
- dataGetWarnings, 67
- dataOpen, 68
- dataPutBlock, 68
- dataRealtimeConfig, 68
- dataRealtimeGet, 68
- dataSearch, 69
- dataSeekBlock, 69
- dataSetInfo, 69
- digitiserDelete, 69
- digitiserGet, 69
- digitiserGetList, 70
- digitiserUpdate, 70
- eventDelete, 70
- eventGetList, 70
- eventUpdate, 70
- extraCall, 71
- getSelectionInfo, 71
- getSelections, 71
- getVersion, 71
- groupDelete, 71
- groupGetList, 72
- groupUpdate, 72
- locationDelete, 72
- locationGetList, 72
- locationUpdate, 72
- logAppend, 73
- logDelete, 73
- logGetList, 73
- logUpdate, 73
- metadataGetChannelInfo, 73
- metadataGetFormatted, 74
- modeSet, 74
- modeSnapshotPause, 74
- networkDelete, 74
- networkGetList, 74
- networkUpdate, 75
- noteDelete, 75
- noteGetList, 75
- noteReadDocument, 75
- noteUpdate, 75
- noteWriteDocument, 76
- responseDelete, 76
- responseGetList, 76
- responseUpdate, 76
- sensorDelete, 76
- sensorGet, 77
- sensorGetList, 77
- sensorUpdate, 77
- serverConfigurationGet, 77
- setUser, 77
- setUserReal, 78
- sourceDelete, 78
- sourceGetList, 78
- sourcePriorityDelete, 78
- sourcePriorityGetList, 78
- sourcePriorityUpdate, 78
- sourceUpdate, 79
- specialChannelDelete, 79
- specialChannelGetList, 79
- specialChannelUpdate, 79
- sqlQuery, 79
- stationDelete, 80
- stationGetList, 80
- stationUpdate, 80
- statisticsGet, 80
- transactionEnd, 80
- transactionStart, 81
- userDelete, 81
- userGet, 81
- userGetFromId, 81
- userGetGroups, 81
- userGetList, 81
- userGetOptions, 82
- userSet, 82
- userSetOptions, 82
- userUpdate, 82
- validateUser, 82
- Bds::ArrayChannel, 83
 - ArrayChannel, 83
 - arrayOffsetEast, 84
 - arrayOffsetNorth, 84
 - channel, 84
 - network, 84
 - station, 84
- Bds::BdsDataBlock, 85
 - data, 85
 - header, 85
- Bds::BdsDataBlockHeader, 86
 - length, 86
 - packetOffset, 86
 - type, 86
- Bds::BdsDataBlockPos, 87
 - BdsDataBlockPos, 87
 - channel, 88
 - endTime, 88
 - numChannels, 88
 - numSamples, 88
 - operator<, 87
 - position, 88
 - segment, 88
 - startTime, 88
- Bds::BdsDataPacket, 89
 - ~BdsDataPacket, 90
 - BdsDataPacket, 89
 - clear, 90
 - dump, 90
 - getHeader, 90
 - reset, 90
 - setChecksumAndLength, 90

- setHeader, 90
 - validateChecksum, 91
- Bds::BdsDataPacketHeader, 91
 - checksum, 91
 - endTime, 92
 - length, 92
 - sequence, 92
 - startTime, 92
 - streamlet, 92
 - type, 92
- Bds::BdsDataSegment, 93
 - BdsDataSegment, 93
 - blocks, 94
 - endTime, 94
 - numBlocks, 94
 - numSamples, 94
 - operator<, 93
 - sampleRate, 94
 - startTime, 94
- Bds::BdsDataStreamlet, 95
 - BdsDataStreamlet, 95
 - blocks, 95
 - channel, 95
 - numChannels, 96
 - packetNumber, 96
 - position, 96
 - segments, 96
- Bds::BdsSeedType, 96
 - appendDouble, 97
 - appendExp, 97
 - appendInt, 97
 - appendString, 98
 - appendStringVariable, 98
 - BdsSeedType, 97
 - getDouble, 98
 - getInt, 98
 - getString, 98
 - getStringVariable, 98
 - getUInt, 99
- Bds::Calibration, 99
 - Calibration, 101
 - calibrationFactor, 102
 - calibrationFrequency, 102
 - calibrationUnits, 102
 - calibrationUnitsDesc, 103
 - channel, 103
 - depth, 103
 - endTime, 103
 - getMember, 101
 - getMembers, 101
 - getType, 101
 - horizontalAngle, 103
 - id, 103
 - name, 104
 - network, 104
 - rawCalibrationFactor, 104
 - rawCalibrationFrequency, 104
 - rawCalibrationUnits, 104
 - samplingFrequency, 104
 - setMember, 102
 - setMembers, 102
 - source, 105
 - startTime, 105
 - station, 105
 - verticalAngle, 105
 - waterLevel, 105
- Bds::CdChannel_1v0, 106
 - auth, 106
 - calibrationFactor, 106
 - calibrationPeriod, 106
 - channel, 106
 - channelName, 107
 - compress, 107
 - name, 107
 - spare0, 107
 - spare1, 107
 - stationName, 107
- Bds::CdDataChannel, 108
 - channel, 108
 - data, 108
 - dataSize, 108
 - mode, 108
 - numSamples, 109
 - period, 109
 - startTime, 109
 - station, 109
 - status, 109
- Bds::CdDataFormatFrame_1v0, 109
 - channels, 110
 - frameLength, 110
 - frameType, 110
 - maxFrameLength, 110
 - numChannels, 110
 - period, 110
- Bds::CdFlag, 111
 - CdFlag, 111
 - dead, 111
 - zeroed, 111
- Bds::CdPacketData, 112
 - auth, 112
 - authKey, 112
 - authSize, 113
 - channels, 113
 - crc, 113
 - creator, 113
 - destination, 113
 - frameType, 113
 - numChannels, 113
 - period, 113
 - sequenceNum, 114
 - series, 114
 - startTime, 114
 - trailerOffset, 114
- Bds::Change, 114
 - Change, 115
 - changeGroupId, 117

- getMember, 116
- getMembers, 116
- getType, 116
- id, 117
- rowId, 117
- setMember, 116
- setMembers, 116
- table, 117
- time, 117
- type, 117
- Bds::ChangeGroup, 118
 - ChangeGroup, 119
 - description, 120
 - getMember, 119
 - getMembers, 119
 - getType, 119
 - id, 120
 - setMember, 119
 - setMembers, 120
 - time, 120
 - title, 120
 - type, 121
 - user, 121
- Bds::Channel, 121
 - Channel, 122
 - channel, 124
 - channelAux, 124
 - channelType, 124
 - dataType, 124
 - description, 124
 - endTime, 124
 - getMember, 123
 - getMembers, 123
 - getType, 123
 - id, 125
 - network, 125
 - setMember, 123
 - setMembers, 123
 - startTime, 125
 - station, 125
- Bds::ChannelInfo, 125
 - calibration, 127
 - channel, 127
 - ChannelInfo, 126
 - channelLocation, 127
 - digitiser, 127
 - endTime, 127
 - responses, 128
 - sensor, 128
 - source, 128
 - startTime, 128
 - station, 128
 - stationLocation, 128
- Bds::ChannelInfos, 129
 - array, 130
 - ChannelInfos, 129
 - channels, 130
- Bds::ChannelInstrument, 130
 - channelId, 132
 - ChannelInstrument, 131
 - digitiserId, 133
 - endTime, 133
 - getMember, 131
 - getMembers, 132
 - getType, 132
 - id, 133
 - sensorId, 133
 - setMember, 132
 - setMembers, 132
 - source, 133
 - startTime, 133
- Bds::ChannelName, 134
 - channel, 135
 - ChannelName, 134
 - network, 135
 - source, 135
 - station, 135
- Bds::CleanOptions, 136
 - changes, 136
 - CleanOptions, 136
 - deletedFiles, 136
 - logs, 137
- Bds::CompressSteim1, 137
 - clear, 138
 - CompressSteim1, 137
 - setByteOrder, 138
 - unCompress, 138
- Bds::DataAccess, 138
 - calibrationGetList, 142
 - channelGetList, 142
 - channelInstrumentGetList, 142
 - clean, 142
 - connect, 142
 - DataAccess, 141
 - dataAvailability, 142
 - databaseBackup, 143
 - dataChannelGetList, 143
 - dataClose, 143
 - dataFileGetList, 143
 - dataFormatGetList, 143
 - dataFormattedGetLength, 144
 - dataFormattedRead, 144
 - dataGetBlock, 144
 - dataGetChannelInfo, 144
 - dataGetInfo, 144
 - dataGetNotes, 145
 - dataGetWarnings, 145
 - dataOpen, 145
 - dataRealtimeConfig, 146
 - dataRealtimeGet, 146
 - dataSearch, 146
 - dataSeekBlock, 146
 - digitiserGet, 146
 - digitiserGetList, 147
 - eventGetList, 147
 - getSelectionInfo, 147

- getSelections, 147
- getVersion, 147
- groupGetList, 148
- locationGetList, 148
- logAppend, 148
- logUpdate, 148
- metadataGetChannelInfo, 148
- metadataGetFormatted, 149
- modeSet, 149
- modeSnapshotPause, 149
- networkGetList, 149
- noteGetList, 149
- noteReadDocument, 150
- noteUpdate, 150
- noteWriteDocument, 150
- responseGetList, 150
- sensorGet, 150
- sensorGetList, 151
- serverConfigurationGet, 151
- setUser, 151
- setUserReal, 151
- sourceGetList, 151
- sourcePriorityGetList, 152
- specialChannelGetList, 152
- stationGetList, 152
- statisticsGet, 152
- userGet, 152
- userGetFromId, 153
- userGetGroups, 153
- userGetOptions, 153
- userSet, 153
- userSetOptions, 153
- validateUser, 154
- Bds::DataAddAccess, 154
 - calibrationGetList, 158
 - channelGetList, 158
 - channelInstrumentGetList, 158
 - clean, 158
 - connect, 158
 - DataAddAccess, 157
 - dataAvailability, 159
 - databaseBackup, 159
 - dataChannelGetList, 159
 - dataClose, 159
 - dataFileGetList, 159
 - dataFormatGetList, 160
 - dataFormattedGetLength, 160
 - dataFormattedRead, 160
 - dataGetBlock, 160
 - dataGetChannelInfo, 160
 - dataGetInfo, 161
 - dataGetNotes, 161
 - dataGetWarnings, 161
 - dataOpen, 161
 - dataPutBlock, 162
 - dataRealtimeConfig, 162
 - dataRealtimeGet, 162
 - dataSearch, 162
 - dataSeekBlock, 162
 - dataSetInfo, 163
 - digitiserGet, 163
 - digitiserGetList, 163
 - eventDelete, 163
 - eventGetList, 163
 - eventUpdate, 164
 - getSelectionInfo, 164
 - getSelections, 164
 - getVersion, 164
 - groupGetList, 164
 - locationGetList, 165
 - logAppend, 165
 - logUpdate, 165
 - metadataGetChannelInfo, 165
 - metadataGetFormatted, 165
 - modeSet, 166
 - modeSnapshotPause, 166
 - networkGetList, 166
 - noteGetList, 166
 - noteReadDocument, 166
 - noteUpdate, 167
 - noteWriteDocument, 167
 - responseGetList, 167
 - sensorGet, 167
 - sensorGetList, 167
 - serverConfigurationGet, 168
 - setUser, 168
 - setUserReal, 168
 - sourceGetList, 168
 - sourcePriorityGetList, 168
 - specialChannelGetList, 169
 - stationGetList, 169
 - statisticsGet, 169
 - userGet, 169
 - userGetFromId, 169
 - userGetGroups, 170
 - userGetOptions, 170
 - userSet, 170
 - userSetOptions, 170
 - validateUser, 170
- Bds::DataAvail, 171
 - availType, 172
 - DataAvail, 171
 - endTime, 172
 - startTime, 172
- Bds::DataAvailChan, 172
 - channel, 173
 - DataAvailChan, 173
 - endTime, 174
 - network, 174
 - segments, 174
 - source, 174
 - startTime, 174
 - station, 174
- Bds::DataBlock, 175
 - channelData, 176
 - channelNumber, 176

- DataBlock, 176
- endTime, 176
- info, 176
- segmentNumber, 176
- startTime, 177
- Bds::DataBlockChannel, 177
 - channel, 178
 - DataBlockChannel, 178
 - network, 178
 - source, 178
 - station, 178
- Bds::DataBlockPos, 179
 - DataBlockPos, 179
 - endTime, 180
 - numSamples, 180
 - operator<, 179
 - order, 180
 - position, 180
 - ref, 180
 - startTime, 180
- Bds::DataChannel, 181
 - channel, 184
 - DataChannel, 182
 - dataFileChannel, 184
 - dataFileId, 184
 - endTime, 184
 - getMember, 183
 - getMembers, 183
 - getType, 183
 - id, 184
 - importFilename, 184
 - importFormat, 185
 - importStartTime, 185
 - info, 185
 - network, 185
 - numBlocks, 185
 - numSamples, 185
 - sampleFormat, 186
 - sampleRate, 186
 - setMember, 183
 - setMembers, 183
 - source, 186
 - startTime, 186
 - station, 186
- Bds::DataCollate, 187
 - ~DataCollate, 187
 - addSource, 187
 - DataCollate, 187
 - readData, 187
- Bds::DataError, 188
 - DataError, 189
 - getErrorNumber, 190
 - getString, 190
 - getTitle, 190
 - mergeDataInfo, 190
 - num, 190
 - ochannel, 192
 - odescription, 192
 - oendTime, 192
 - oerrorNumber, 192
 - ofilename, 192
 - onetwork, 192
 - operator int, 190
 - osource, 193
 - ostartTime, 193
 - ostation, 193
 - otitle, 193
 - ouser, 193
 - set, 191
 - setString, 191
 - setStringUser, 191
 - str, 191
- Bds::DataFile, 194
 - ~DataFile, 197
 - close, 198
 - dataErrorFixup, 198
 - DataFile, 197
 - DataOrder, 196
 - DataOrderAll, 196
 - DataOrderChannel, 196
 - DataOrderSample, 196
 - DataOrderUnknown, 196
 - duplicateCheck, 198
 - end, 198
 - FeatureCanRead, 196
 - FeatureCanWrite, 196
 - FeatureNone, 196
 - Features, 196
 - fileNameProcess, 198
 - flush, 198
 - getDataOrder, 199
 - getFeatures, 199
 - getFileName, 199
 - getFilePosition, 199
 - getFixesInfo, 199
 - getFormat, 200
 - getFormats, 200
 - getInfo, 200
 - getMetaData, 200
 - init, 200
 - ofile, 203
 - ofilename, 203
 - ofilenameTime, 203
 - oformat, 203
 - omode, 203
 - open, 201
 - readData, 201
 - ReadOptionDeleteDuplicates, 197
 - ReadOptionFileNameProcess, 197
 - ReadOptionFixCorruptions, 197
 - ReadOptionFixSampleRate, 197
 - ReadOptionIgnoreSamplerate, 197
 - ReadOptionInfoExtra, 197
 - ReadOptionNone, 197
 - ReadOptionPrintBlocks, 197
 - ReadOptionReorder, 197

- ReadOptionsList, 196
- ReadOptionValidate, 197
- seekBlock, 201
- setFormat, 201
- setInfo, 202
- start, 202
- timeCompare, 202
- writeData, 202
- WriteOptionNoMetadata, 197
- WriteOptionNone, 197
- WriteOptionSensorData, 197
- WriteOptionsList, 197
- Bds::DataFileAd22, 204
 - DataFileAd22, 204
 - getDataOrder, 205
 - getFeatures, 205
 - getFixesInfo, 205
 - getFormats, 205
 - getInfo, 205
 - readData, 205
- Bds::DataFileAscii, 206
 - DataFileAscii, 207
 - end, 207
 - getDataOrder, 207
 - getFeatures, 207
 - getFormats, 208
 - open, 208
 - setFormat, 208
 - setInfo, 208
 - start, 208
 - writeData, 209
- Bds::DataFileBdrs, 209
 - DataFileBdrs, 210
 - getDataOrder, 210
 - getFeatures, 210
 - getFixesInfo, 210
 - getFormats, 211
 - getInfo, 211
 - readData, 211
- Bds::DataFileBds, 212
 - ~DataFileBds, 214
 - close, 214
 - DataFileBds, 214
 - DefaultBlockSize, 213
 - flush, 214
 - getDataOrder, 215
 - getDiskBlockSize, 215
 - getFormats, 215
 - getInfo, 215
 - open, 215
 - packetRead, 216
 - packetWrite, 216
 - PackFormat, 214
 - PackFormat_CM, 214
 - PackFormat_SM, 214
 - PackFormat_SM_CC, 214
 - PackFormat_Unknown, 214
 - readData, 216
 - seekBlock, 216
 - setDiskBlockSize, 217
 - setFormat, 217
 - setInfo, 217
 - setReadPositionToStart, 217
 - setWritePositionForAppend, 218
 - streamletToChannel, 218
 - StreamsMax, 213
 - writeData, 218
- Bds::DataFileBknas, 218
 - DataFileBknas, 219
 - getFormats, 219
 - open, 219
 - setInfo, 220
 - writeData, 220
- Bds::DataFileCd, 220
 - DataFileCd, 221
 - getDataOrder, 221
 - getFeatures, 221
 - getFixesInfo, 222
 - getFormats, 222
 - getInfo, 222
 - readData, 222
- Bds::DataFileCss, 223
 - DataFileCss, 223
 - getDataOrder, 224
 - getFeatures, 224
 - getFormats, 224
 - getInfo, 224
 - readData, 224
- Bds::DataFileCssData, 225
 - ~DataFileCssData, 226
 - calibrationFactor, 226
 - calibrationFreq, 227
 - chan, 227
 - chanid, 227
 - clip, 227
 - commId, 227
 - DataFileCssData, 226
 - datatype, 227
 - dirName, 227
 - endTime, 227
 - file, 228
 - fileName, 228
 - fileOffset, 228
 - instType, 228
 - jdate, 228
 - loadDate, 228
 - nsamp, 228
 - sampleBigEndian, 228
 - sampleFormat, 229
 - sampleRate, 229
 - sampleSize, 229
 - segtype, 229
 - set, 226
 - sta, 229
 - startTime, 229
 - wfid, 229

- Bds::DataFileGcf, 230
 - DataFileGcf, 231
 - getDataOrder, 231
 - getFeatures, 231
 - getFixesInfo, 231
 - getFormats, 231
 - getInfo, 231
 - readData, 232
- Bds::DataFileIdc, 232
 - DataFileIdc, 233
 - getFeatures, 233
 - getFormats, 233
 - getMetaData, 233
 - setInfo, 233
- Bds::DataFileIms, 234
 - close, 235
 - DataFileIms, 235
 - end, 235
 - getDataOrder, 235
 - getFeatures, 236
 - getFormats, 236
 - getMetaData, 236
 - open, 236
 - setInfo, 236
 - start, 237
 - writeData, 237
- Bds::DataFileInfo, 237
 - comment, 240
 - DataFileInfo, 238
 - endTime, 240
 - format, 240
 - getMember, 239
 - getMembers, 239
 - getType, 239
 - id, 240
 - importTime, 240
 - importUserId, 240
 - location, 241
 - setMember, 239
 - setMembers, 239
 - startTime, 241
 - state, 241
 - stream, 241
 - url, 241
- Bds::DataFileLac, 242
 - DataFileLac, 242
 - getDataOrder, 243
 - getFeatures, 243
 - getFixesInfo, 243
 - getFormats, 243
 - getInfo, 243
 - readData, 243
- Bds::DataFileLog, 244
 - DataFileLog, 245
 - end, 245
 - getDataOrder, 245
 - getFeatures, 245
 - getFormats, 246
 - getInfo, 246
 - open, 246
 - readData, 246
 - setFormat, 246
 - setInfo, 247
 - start, 247
 - writeData, 247
- Bds::DataFileOptions, 248
 - DataFileOptions, 248
 - oignoreBlockList, 249
 - ooptionList, 249
 - operator int, 248
 - operator | =, 248
- Bds::DataFileResponse, 249
 - DataFileResponse, 250
 - getFeatures, 250
 - getFormats, 250
 - getMetaData, 250
 - setInfo, 250
- Bds::DataFileSac, 251
 - DataFileSac, 252
 - getFeatures, 252
 - getFormats, 252
 - getMetaData, 252
 - setInfo, 252
- Bds::DataFileSeed, 253
 - ~DataFileSeed, 254
 - close, 254
 - DataFileSeed, 254
 - end, 255
 - getDataOrder, 255
 - getFeatures, 255
 - getFixesInfo, 255
 - getFormats, 255
 - getInfo, 256
 - getMetaData, 256
 - msrFileWrite, 256
 - omsrErr, 257
 - onoLock, 258
 - readData, 256
 - setFormat, 256
 - setInfo, 257
 - start, 257
 - writeData, 257
- Bds::DataFileStationXml, 258
 - DataFileStationXml, 259
 - getFeatures, 259
 - getFormats, 259
 - getMetaData, 259
 - setInfo, 259
- Bds::DataFileTapeDigitiser, 260
 - DataFileTapeDigitiser, 261
 - getFormats, 261
 - getInfo, 261
 - open, 261
 - readData, 261
- Bds::DataFileWra, 262
 - DataFileWra, 263

- getDataOrder, [263](#)
 - getFeatures, [263](#)
 - getFixesInfo, [263](#)
 - getFormats, [263](#)
 - getInfo, [264](#)
 - readData, [264](#)
 - setFormat, [264](#)
- Bds::DataFileWraAgso, [265](#)
 - DataFileWraAgso, [265](#)
 - getDataOrder, [266](#)
 - getFeatures, [266](#)
 - getFormats, [266](#)
 - getInfo, [266](#)
 - readData, [266](#)
- Bds::DataFormat, [267](#)
 - DataFormat, [268](#)
 - dataRead, [268](#)
 - dataWrite, [268](#)
 - description, [268](#)
 - extension, [268](#)
 - metadataRead, [269](#)
 - metadataWrite, [269](#)
 - names, [269](#)
- Bds::DataFormatAll, [269](#)
 - ~DataFormatAll, [270](#)
 - DataFormatAll, [270](#)
 - findFormat, [270](#)
 - formatGet, [270](#)
 - formatGetExtension, [271](#)
 - formatList, [271](#)
- Bds::DataHandle, [271](#)
 - dataFileId, [272](#)
 - DataHandle, [272](#)
 - handle, [272](#)
- Bds::DataInfo, [272](#)
 - array, [274](#)
 - channels, [274](#)
 - DataInfo, [273](#)
 - description, [274](#)
 - endTime, [274](#)
 - info, [274](#)
 - infoExtra, [275](#)
 - startTime, [275](#)
 - synchronous, [275](#)
 - warnings, [275](#)
- Bds::Digitiser, [276](#)
 - baseSamplingFrequency, [278](#)
 - Digitiser, [277](#)
 - endTime, [278](#)
 - gain, [278](#)
 - getMember, [277](#)
 - getMembers, [277](#)
 - getType, [277](#)
 - id, [278](#)
 - initialSamplingFrequency, [279](#)
 - name, [279](#)
 - numberChannels, [279](#)
 - serialNumber, [279](#)
 - setMember, [278](#)
 - setMembers, [278](#)
 - shared, [279](#)
 - startTime, [279](#)
 - type, [280](#)
- Bds::Event, [280](#)
 - dataChannels, [282](#)
 - description, [282](#)
 - elevation, [282](#)
 - endTime, [282](#)
 - Event, [281](#)
 - eventTime, [282](#)
 - extra, [283](#)
 - id, [283](#)
 - latitude, [283](#)
 - longitude, [283](#)
 - magnitude, [283](#)
 - magnitudeUnits, [283](#)
 - network, [284](#)
 - notes, [284](#)
 - source, [284](#)
 - startTime, [284](#)
 - title, [284](#)
 - type, [284](#)
 - userId, [285](#)
 - waterDepth, [285](#)
- Bds::Fap, [285](#)
 - amplitude, [286](#)
 - Fap, [286](#)
 - frequency, [286](#)
 - phase, [286](#)
- Bds::Fir, [287](#)
 - a, [287](#)
 - b, [287](#)
 - Fir, [287](#)
- Bds::FirEntry, [288](#)
 - coefficient, [288](#)
 - error, [289](#)
 - FirEntry, [288](#)
- Bds::GcfChannel, [289](#)
 - channel, [289](#)
 - format, [289](#)
 - sampleRate, [290](#)
 - streamId, [290](#)
 - systemId, [290](#)
 - type, [290](#)
- Bds::Group, [290](#)
 - description, [292](#)
 - getMember, [291](#)
 - getMembers, [291](#)
 - getType, [292](#)
 - Group, [291](#)
 - group, [292](#)
 - id, [293](#)
 - setMember, [292](#)
 - setMembers, [292](#)
- Bds::ListRange, [293](#)
 - getMember, [294](#)

- getMembers, 294
- getType, 294
- ListRange, 294
- number, 295
- reverse, 295
- setMember, 294
- setMembers, 295
- start, 295
- Bds::Location, 296
 - arrayOffsetEast, 298
 - arrayOffsetNorth, 298
 - channel, 298
 - datum, 298
 - elevation, 299
 - endTime, 299
 - getMember, 297
 - getMembers, 297
 - getType, 297
 - id, 299
 - latitude, 299
 - Location, 297
 - longitude, 299
 - network, 299
 - setMember, 298
 - setMembers, 298
 - startTime, 300
 - station, 300
- Bds::Log, 300
 - description, 302
 - getMember, 301
 - getMembers, 301
 - getType, 302
 - id, 302
 - Log, 301
 - priority, 303
 - setMember, 302
 - setMembers, 302
 - subSystem, 303
 - time, 303
 - title, 303
 - type, 303
- Bds::LogSelect, 304
 - LogSelect, 304
 - priority, 304
 - startTime, 305
 - subSystem, 305
 - type, 305
- Bds::Network, 305
 - description, 307
 - getMember, 306
 - getMembers, 307
 - getType, 307
 - id, 308
 - Network, 306
 - network, 308
 - setMember, 307
 - setMembers, 307
 - stations, 308
- Bds::Note, 308
 - channel, 311
 - dataFileId, 311
 - description, 312
 - docFormat, 312
 - docUrl, 312
 - endTime, 312
 - errorNumber, 312
 - eventId, 312
 - getMember, 310
 - getMembers, 310
 - getType, 311
 - id, 313
 - importFilename, 313
 - network, 313
 - Note, 310
 - setMember, 311
 - setMembers, 311
 - source, 313
 - startTime, 313
 - station, 313
 - timeAdded, 314
 - title, 314
 - type, 314
 - user, 314
- Bds::Point, 314
 - Point, 315
 - x, 315
 - y, 315
- Bds::PoleZero, 316
 - poles, 316
 - PoleZero, 316
 - zeros, 316
- Bds::Polynomial, 317
 - approximationLowerBound, 318
 - approximationType, 318
 - approximationUpperBound, 318
 - coefficients, 318
 - frequencyLowerBound, 319
 - frequencyUpperBound, 319
 - maximumError, 319
 - Polynomial, 318
 - transferType, 319
 - validFrequencyUnits, 319
- Bds::PolynomialEntry, 320
 - coefficient, 320
 - measurementMethod, 321
 - minusError, 321
 - plusError, 321
 - PolynomialEntry, 320
- Bds::Response, 321
 - channel, 324
 - decimation, 324
 - decimationCorr, 324
 - decimationDelay, 324
 - decimationOffset, 325
 - description, 325
 - endTime, 325

- faps, [325](#)
- fir, [325](#)
- gain, [325](#)
- gainFrequency, [326](#)
- id, [326](#)
- inputUnits, [326](#)
- inputUnitsDesc, [326](#)
- measured, [326](#)
- name, [326](#)
- network, [327](#)
- outputUnits, [327](#)
- outputUnitsDesc, [327](#)
- poleZeros, [327](#)
- polynomial, [327](#)
- Response, [323](#)
- sampleRate, [327](#)
- source, [328](#)
- stage, [328](#)
- stageType, [328](#)
- startTime, [328](#)
- station, [328](#)
- symmetry, [328](#)
- type, [329](#)
- Bds::ResponseObj, [329](#)
 - ~ResponseObj, [330](#)
 - getString, [330](#)
 - ResponseObj, [330](#)
 - setString, [330](#)
- Bds::Selection, [330](#)
 - array, [332](#)
 - calibrationName, [332](#)
 - channelId, [333](#)
 - channels, [333](#)
 - completeSegments, [333](#)
 - dataTypes, [333](#)
 - digitiserId, [333](#)
 - endTime, [333](#)
 - eventId, [334](#)
 - excludeChannels, [334](#)
 - id, [334](#)
 - locationSelect, [334](#)
 - name, [334](#)
 - range, [334](#)
 - Selection, [332](#)
 - sensorId, [335](#)
 - sensorOldId, [335](#)
 - startTime, [335](#)
- Bds::SelectionChannel, [335](#)
 - channel, [336](#)
 - network, [336](#)
 - SelectionChannel, [336](#)
 - source, [336](#)
 - station, [336](#)
- Bds::SelectionInfo, [337](#)
 - arrays, [338](#)
 - arraysAndStations, [338](#)
 - channels, [338](#)
 - endTime, [338](#)
 - networks, [339](#)
 - numDataChannels, [339](#)
 - SelectionInfo, [338](#)
 - sources, [339](#)
 - startTime, [339](#)
 - stations, [339](#)
- Bds::Sensor, [340](#)
 - endTime, [342](#)
 - gain, [342](#)
 - gainUnits, [342](#)
 - getMember, [341](#)
 - getMembers, [341](#)
 - getType, [341](#)
 - id, [342](#)
 - name, [343](#)
 - numberChannels, [343](#)
 - oldId, [343](#)
 - Sensor, [341](#)
 - serialNumber, [343](#)
 - setMember, [342](#)
 - setMembers, [342](#)
 - shared, [343](#)
 - startTime, [343](#)
 - type, [344](#)
- Bds::Source, [344](#)
 - alias, [346](#)
 - description, [346](#)
 - getMember, [345](#)
 - getMembers, [345](#)
 - getType, [346](#)
 - id, [347](#)
 - setMember, [346](#)
 - setMembers, [346](#)
 - Source, [345](#)
 - source, [347](#)
 - sourceMeta, [347](#)
- Bds::SourcePriority, [347](#)
 - endTime, [349](#)
 - getMember, [348](#)
 - getMembers, [349](#)
 - getType, [349](#)
 - id, [350](#)
 - priority, [350](#)
 - setMember, [349](#)
 - setMembers, [349](#)
 - source, [350](#)
 - SourcePriority, [348](#)
 - startTime, [350](#)
- Bds::SpecialChannel, [351](#)
 - channel, [353](#)
 - dataType, [353](#)
 - description, [353](#)
 - endTime, [353](#)
 - getMember, [352](#)
 - getMembers, [352](#)
 - getType, [352](#)
 - id, [353](#)
 - network, [353](#)

- setMember, [352](#)
 - setMembers, [352](#)
 - SpecialChannel, [351](#)
 - startTime, [354](#)
 - station, [354](#)
- Bds::Station, [354](#)
 - alias, [355](#)
 - channels, [355](#)
 - description, [355](#)
 - id, [355](#)
 - name, [356](#)
 - network, [356](#)
 - Station, [355](#)
 - type, [356](#)
- Bds::TimePeriod, [356](#)
 - endTime, [358](#)
 - getMember, [357](#)
 - getMembers, [357](#)
 - getType, [358](#)
 - setMember, [358](#)
 - setMembers, [358](#)
 - startTime, [358](#)
 - TimePeriod, [357](#)
- Bds::User, [359](#)
 - address, [361](#)
 - email, [361](#)
 - enabled, [361](#)
 - getMember, [360](#)
 - getMembers, [360](#)
 - getType, [360](#)
 - groups, [361](#)
 - id, [362](#)
 - name, [362](#)
 - password, [362](#)
 - setMember, [361](#)
 - setMembers, [361](#)
 - telephone, [362](#)
 - User, [360](#)
 - user, [362](#)
- BdsC.cc, [393](#)
- BdsC.d, [394](#)
- BdsC.h, [394](#)
- bdsChannelGetName
 - Bds, [33](#)
- bdsChannelGetTypeAux
 - Bds, [34](#)
- BdsD.cc, [394](#)
- BdsD.d, [395](#)
- BdsD.h, [395](#)
- BdsDataBlockPos
 - Bds::BdsDataBlockPos, [87](#)
- bdsDataChannelInfo
 - Bds, [34](#)
- bdsDataChannelOverallResponse
 - Bds, [34](#)
- bdsDataChannelRef
 - Bds, [34](#)
- BdsDataFileAd22.cpp
 - DEBUG_VELATRACK, [366](#)
- BdsDataFileBds.cpp
 - ALLOW_TIMESTAMP_JITTER, [369](#)
 - dl2printf, [369](#)
 - dl3printf, [369](#)
 - dlprintf, [369](#)
 - LDEBUG, [369](#)
 - LDEBUG2, [369](#)
 - LDEBUG3, [369](#)
 - TIMESTAMP_JITTER, [370](#)
- BdsDataFileBknas.cpp
 - clip, [371](#)
- BdsDataFileCd.cpp
 - ALLOW_TIMESTAMP_JITTER, [373](#)
 - dprintf, [373](#)
 - ErrorFormatNoDataFormat, [374](#)
 - htonll, [373](#)
 - INCLUDE_CHANNEL_AUTH, [373](#)
 - LDEBUG, [373](#)
 - MULTIPLE_SEGMENT, [373](#)
 - ntohl, [373](#)
 - SEGMENT_GAP, [374](#)
 - TIMESTAMP_JITTER, [374](#)
- BdsDataFileGcf.cpp
 - DEBUG, [376](#)
 - TEST_REORDER, [376](#)
- BdsDataFileIdc.cpp
 - dprintf, [377](#)
 - LDEBUG, [377](#)
- BdsDataFileResponse.cpp
 - dprintf, [381](#)
 - LDEBUG, [381](#)
- BdsDataFileSac.cpp
 - BDEBUGL1, [382](#)
- BdsDataFileSeed.cpp
 - BDEBUGL1, [389](#)
 - BDEBUGL2, [389](#)
 - BDEBUGL3, [389](#)
 - DEBUG, [389](#)
 - DEBUG_BLOCKETTE, [389](#)
 - DEBUG_BLOCKS, [389](#)
 - FILL_BLOCKS, [389](#)
 - ROUND_TIMESTAMPS_US, [389](#)
- BdsDataFileSeed.h
 - MSRecord, [390](#)
- bdsDataFileSeedLogError
 - Bds, [35](#)
- bdsDataFileSeedLogWarning
 - Bds, [35](#)
- BdsDataFileStationXml.cpp
 - BDEBUGL1, [383](#)
 - BDEBUGL2, [383](#)
- BdsDataFileVersion
 - Bds, [46](#)
- BdsDataFileWraAgso.cpp
 - parseStringFixedFields, [386](#)
- bdsDataInfoFlatten
 - Bds, [35](#)

- bdsDataInfoFromInfo
 - Bds, [35](#)
- bdsDataInfoMergeFlatten
 - Bds, [35](#)
- bdsDataInfoSetTimeRange
 - Bds, [35](#)
- BdsDataPacket
 - Bds::BdsDataPacket, [89](#)
- BdsDataSegment
 - Bds::BdsDataSegment, [93](#)
- BdsDataStreamlet
 - Bds::BdsDataStreamlet, [95](#)
- BdsDataType
 - Bds, [29](#)
- BdsDataTypeBlock
 - Bds, [29](#)
- BdsDataTypeData
 - Bds, [29](#)
- BdsDataTypeInfo
 - Bds, [29](#)
- BdsDataTypeInfoExtra
 - Bds, [29](#)
- bdsDataTypes
 - Bds, [36](#)
- bdsDumpChannelInfos
 - Bds, [36](#)
- bdsDumpData
 - Bds, [36](#)
- bdsDumpDataInfo
 - Bds, [36](#)
- bdsDumpLocation
 - Bds, [36](#)
- bdsDumpPoleZeros
 - Bds, [37](#)
- bdsDumpSelection
 - Bds, [37](#)
- bdsDumpSelectionInfo
 - Bds, [37](#)
- bdsFileNameExpand
 - Bds, [37](#), [38](#)
- bdsInfoFromDataInfo
 - Bds, [38](#)
- BdsLib.cpp, [398](#)
- BdsLib.d, [400](#)
- BdsLib.dox, [400](#)
- BdsLib.h, [400](#)
- bdsLibInit
 - Bds, [38](#)
- bdsMetadataExportFix
 - Bds, [39](#)
- bdsMetadataImportFix
 - Bds, [39](#)
- bdsPoleZeroGain
 - Bds, [39](#)
- bdsPoleZeroGainPhase
 - Bds, [39](#)
- bdsPoleZeroToFap
 - Bds, [39](#)
- BdsS.cc, [403](#)
- BdsS.d, [403](#)
- BdsSeedType
 - Bds::BdsSeedType, [97](#)
- bdsSelectionChannelInfo
 - Bds, [40](#)
- bdsSpecialChannelIgnore
 - Bds, [40](#)
- bdsSpecialChannels
 - Bds, [40](#)
- bdsSpecialChannelsList
 - Bds, [47](#)
- bdsSpecialChannelsSet
 - Bds, [40](#)
- bdsStationAlias
 - Bds, [40](#)
- BdsT.cc, [403](#)
- bdsUnCompressCm8
 - Bds, [41](#)
- bdsUnCompressSteim1
 - Bds, [41](#)
- bdsUnitCase
 - Bds, [41](#)
- bdsUnits
 - Bds, [41](#)
- bdsUnitsConvert
 - Bds, [41](#)
- blocks
 - Bds::BdsDataSegment, [94](#)
 - Bds::BdsDataStreamlet, [95](#)
- Calibration
 - Bds::Calibration, [101](#)
- calibration
 - Bds::ChannelInfo, [127](#)
- calibrationDelete
 - Bds::AdminAccess, [60](#)
- calibrationFactor
 - Bds::Calibration, [102](#)
 - Bds::CdChannel_1v0, [106](#)
 - Bds::DataFileCssData, [226](#)
- calibrationFreq
 - Bds::DataFileCssData, [227](#)
- calibrationFrequency
 - Bds::Calibration, [102](#)
- calibrationGetList
 - Bds::AdminAccess, [60](#)
 - Bds::DataAccess, [142](#)
 - Bds::DataAddAccess, [158](#)
- calibrationName
 - Bds::Selection, [332](#)
- calibrationPeriod
 - Bds::CdChannel_1v0, [106](#)
- calibrationUnits
 - Bds::Calibration, [102](#)
- calibrationUnitsDesc
 - Bds::Calibration, [103](#)
- calibrationUpdate
 - Bds::AdminAccess, [61](#)

- canada_compress
 - canada_compress.h, [393](#)
- canada_compress.h
 - canada_compress, [393](#)
 - canada_uncompress, [393](#)
 - CANCOMP_CORRUPT, [392](#)
 - CANCOMP_ERR, [392](#)
 - CANCOMP_EXCEED, [392](#)
 - CANCOMP_NOT_20, [392](#)
 - CANCOMP_SUCCESS, [392](#)
- canada_uncompress
 - canada_compress.h, [393](#)
- CANCOMP_CORRUPT
 - canada_compress.h, [392](#)
- CANCOMP_ERR
 - canada_compress.h, [392](#)
- CANCOMP_EXCEED
 - canada_compress.h, [392](#)
- CANCOMP_NOT_20
 - canada_compress.h, [392](#)
- CANCOMP_SUCCESS
 - canada_compress.h, [392](#)
- CdFlag
 - Bds::CdFlag, [111](#)
- chan
 - Bds::DataFileCssData, [227](#)
- Change
 - Bds::Change, [115](#)
- changeDelete
 - Bds::AdminAccess, [61](#)
- changeGetList
 - Bds::AdminAccess, [61](#)
- changeGetListNumber
 - Bds::AdminAccess, [61](#)
- ChangeGroup
 - Bds::ChangeGroup, [119](#)
- changeGroupDelete
 - Bds::AdminAccess, [61](#)
- changeGroupEnd
 - Bds::AdminAccess, [62](#)
- changeGroupGetList
 - Bds::AdminAccess, [62](#)
- changeGroupId
 - Bds::Change, [117](#)
- changeGroupStart
 - Bds::AdminAccess, [62](#)
- changes
 - Bds::CleanOptions, [136](#)
- chanid
 - Bds::DataFileCssData, [227](#)
- Channel
 - Bds::Channel, [122](#)
- channel
 - Bds::ArrayChannel, [84](#)
 - Bds::BdsDataBlockPos, [88](#)
 - Bds::BdsDataStreamlet, [95](#)
 - Bds::Calibration, [103](#)
 - Bds::CdChannel_1v0, [106](#)
 - Bds::CdDataChannel, [108](#)
 - Bds::Channel, [124](#)
 - Bds::ChannelInfo, [127](#)
 - Bds::ChannelName, [135](#)
 - Bds::DataAvailChan, [173](#)
 - Bds::DataBlockChannel, [178](#)
 - Bds::DataChannel, [184](#)
 - Bds::GcfChannel, [289](#)
 - Bds::Location, [298](#)
 - Bds::Note, [311](#)
 - Bds::Response, [324](#)
 - Bds::SelectionChannel, [336](#)
 - Bds::SpecialChannel, [353](#)
- channelAux
 - Bds::Channel, [124](#)
- ChannelAuxLen
 - Bds, [47](#)
- channelData
 - Bds::DataBlock, [176](#)
- channelDelete
 - Bds::AdminAccess, [62](#)
- channelGet
 - Bds::AdminAccess, [62](#)
- channelGetList
 - Bds::AdminAccess, [63](#)
 - Bds::DataAccess, [142](#)
 - Bds::DataAddAccess, [158](#)
- channelId
 - Bds::ChannelInstrument, [132](#)
 - Bds::Selection, [333](#)
- ChannelInfo
 - Bds::ChannelInfo, [126](#)
- ChannelInfos
 - Bds::ChannelInfos, [129](#)
- ChannelInstrument
 - Bds::ChannelInstrument, [131](#)
- channelInstrumentDelete
 - Bds::AdminAccess, [63](#)
- channelInstrumentGetList
 - Bds::AdminAccess, [63](#)
 - Bds::DataAccess, [142](#)
 - Bds::DataAddAccess, [158](#)
- channelInstrumentUpdate
 - Bds::AdminAccess, [63](#)
- channelLocation
 - Bds::ChannelInfo, [127](#)
- ChannelName
 - Bds::ChannelName, [134](#)
- channelName
 - Bds::CdChannel_1v0, [107](#)
- channelNumber
 - Bds::DataBlock, [176](#)
- channels
 - Bds::CdDataFormatFrame_1v0, [110](#)
 - Bds::CdPacketData, [113](#)
 - Bds::ChannelInfos, [130](#)
 - Bds::DataInfo, [274](#)
 - Bds::Selection, [333](#)

- Bds::SelectionInfo, 338
- Bds::Station, 355
- channelType
 - Bds::Channel, 124
- ChannelTypeLen
 - Bds, 47
- channelUpdate
 - Bds::AdminAccess, 63
- checksum
 - Bds::BdsDataPacketHeader, 91
- clean
 - Bds::AdminAccess, 64
 - Bds::DataAccess, 142
 - Bds::DataAddAccess, 158
- CleanOptions
 - Bds::CleanOptions, 136
- clear
 - Bds::BdsDataPacket, 90
 - Bds::CompressSteim1, 138
- clip
 - Bds::DataFileCssData, 227
 - BdsDataFileBknas.cpp, 371
- close
 - Bds::DataFile, 198
 - Bds::DataFileBds, 214
 - Bds::DataFileIms, 235
 - Bds::DataFileSeed, 254
- cm6Table
 - Bds, 47
- cm6TableRev
 - Bds, 47
- coefficient
 - Bds::FirEntry, 288
 - Bds::PolynomialEntry, 320
- coefficients
 - Bds::Polynomial, 318
- comment
 - Bds::DataFileInfo, 240
- commlId
 - Bds::DataFileCssData, 227
- completeSegments
 - Bds::Selection, 333
- compress
 - Bds::CdChannel_1v0, 107
- CompressSteim1
 - Bds::CompressSteim1, 137
- connect
 - Bds::AdminAccess, 64
 - Bds::DataAccess, 142
 - Bds::DataAddAccess, 158
- crc
 - Bds, 42
 - Bds::CdPacketData, 113
- crc64
 - Bds, 42
- crclnit
 - Bds, 42
- crclnitDone
 - Bds, 48
- crcVec
 - Bds, 48
- creator
 - Bds::CdPacketData, 113
- data
 - Bds::BdsDataBlock, 85
 - Bds::CdDataChannel, 108
- DataAccess
 - Bds::DataAccess, 141
- DataAddAccess
 - Bds::DataAddAccess, 157
- DataAvail
 - Bds::DataAvail, 171
- dataAvailability
 - Bds::AdminAccess, 64
 - Bds::DataAccess, 142
 - Bds::DataAddAccess, 159
- DataAvailChan
 - Bds::DataAvailChan, 173
- databaseBackup
 - Bds::AdminAccess, 64
 - Bds::DataAccess, 143
 - Bds::DataAddAccess, 159
- databaseRestore
 - Bds::AdminAccess, 64
- DataBlock
 - Bds::DataBlock, 176
- DataBlockChannel
 - Bds::DataBlockChannel, 178
- DataBlockPos
 - Bds::DataBlockPos, 179
- dataCalculateDifference
 - Bds, 42
- dataCalculateUnDifference
 - Bds, 42
- DataChannel
 - Bds::DataChannel, 182
- dataChannelDelete
 - Bds::AdminAccess, 65
- dataChannelGetList
 - Bds::AdminAccess, 65
 - Bds::DataAccess, 143
 - Bds::DataAddAccess, 159
- dataChannels
 - Bds::Event, 282
- dataChannelUpdate
 - Bds::AdminAccess, 65
- dataChecksum
 - Bds, 42
- dataClose
 - Bds::AdminAccess, 65
 - Bds::DataAccess, 143
 - Bds::DataAddAccess, 159
- DataCollate
 - Bds::DataCollate, 187
- dataCompressCm6
 - Bds, 43

- dataConvert
 - Bds, [43](#)
- dataDeCompressCm6
 - Bds, [43](#)
- DataError
 - Bds::DataError, [189](#)
- dataErrorFixup
 - Bds::DataFile, [198](#)
- DataFile
 - Bds::DataFile, [197](#)
- DataFileAd22
 - Bds::DataFileAd22, [204](#)
- DataFileAscii
 - Bds::DataFileAscii, [207](#)
- DataFileBdrs
 - Bds::DataFileBdrs, [210](#)
- DataFileBds
 - Bds::DataFileBds, [214](#)
- DataFileBknas
 - Bds::DataFileBknas, [219](#)
- DataFileCd
 - Bds::DataFileCd, [221](#)
- dataFileChannel
 - Bds::DataChannel, [184](#)
- DataFileCss
 - Bds::DataFileCss, [223](#)
- DataFileCssData
 - Bds::DataFileCssData, [226](#)
- dataFileDelete
 - Bds::AdminAccess, [65](#)
- DataFileGcf
 - Bds::DataFileGcf, [231](#)
- dataFileGetList
 - Bds::AdminAccess, [66](#)
 - Bds::DataAccess, [143](#)
 - Bds::DataAddAccess, [159](#)
- dataFileId
 - Bds::DataChannel, [184](#)
 - Bds::DataHandle, [272](#)
 - Bds::Note, [311](#)
- DataFileIdc
 - Bds::DataFileIdc, [233](#)
- DataFileIms
 - Bds::DataFileIms, [235](#)
- DataFileInfo
 - Bds::DataFileInfo, [238](#)
- DataFileLac
 - Bds::DataFileLac, [242](#)
- DataFileLog
 - Bds::DataFileLog, [245](#)
- DataFileOptions
 - Bds::DataFileOptions, [248](#)
- DataFileResponse
 - Bds::DataFileResponse, [250](#)
- DataFileSac
 - Bds::DataFileSac, [252](#)
- DataFileSeed
 - Bds::DataFileSeed, [254](#)
- DataFileStationXml
 - Bds::DataFileStationXml, [259](#)
- DataFileTapeDigitiser
 - Bds::DataFileTapeDigitiser, [261](#)
- dataFileUpdate
 - Bds::AdminAccess, [66](#)
- DataFileWra
 - Bds::DataFileWra, [263](#)
- DataFileWraAgso
 - Bds::DataFileWraAgso, [265](#)
- DataFlagClipDataToChannels
 - Bds, [29](#)
- DataFlagClipDataToTime
 - Bds, [29](#)
- DataFlagMergeSegments
 - Bds, [29](#)
- DataFlagNoMetadata
 - Bds, [29](#)
- DataFlagNone
 - Bds, [29](#)
- DataFlags
 - Bds, [29](#)
- DataFormat
 - Bds::DataFormat, [268](#)
- DataFormatAll
 - Bds::DataFormatAll, [270](#)
- dataFormatAll
 - Bds, [48](#)
- dataFormatGetList
 - Bds::AdminAccess, [66](#)
 - Bds::DataAccess, [143](#)
 - Bds::DataAddAccess, [160](#)
- DataFormats
 - Bds, [28](#)
- DataFormatSet
 - Bds, [29](#)
- DataFormatSetMetadataRead
 - Bds, [30](#)
- DataFormatSetMetadataWrite
 - Bds, [30](#)
- DataFormatSetNone
 - Bds, [30](#)
- DataFormatSetSensordataRead
 - Bds, [30](#)
- DataFormatSetSensordataWrite
 - Bds, [30](#)
- dataFormattedGetLength
 - Bds::AdminAccess, [66](#)
 - Bds::DataAccess, [144](#)
 - Bds::DataAddAccess, [160](#)
- dataFormattedRead
 - Bds::AdminAccess, [66](#)
 - Bds::DataAccess, [144](#)
 - Bds::DataAddAccess, [160](#)
- dataGetBlock
 - Bds::AdminAccess, [67](#)
 - Bds::DataAccess, [144](#)
 - Bds::DataAddAccess, [160](#)

- dataGetChannelInfo
 - Bds::AdminAccess, [67](#)
 - Bds::DataAccess, [144](#)
 - Bds::DataAddAccess, [160](#)
- dataGetInfo
 - Bds::AdminAccess, [67](#)
 - Bds::DataAccess, [144](#)
 - Bds::DataAddAccess, [161](#)
- dataGetNotes
 - Bds::AdminAccess, [67](#)
 - Bds::DataAccess, [145](#)
 - Bds::DataAddAccess, [161](#)
- dataGetWarnings
 - Bds::AdminAccess, [67](#)
 - Bds::DataAccess, [145](#)
 - Bds::DataAddAccess, [161](#)
- DataHandle
 - Bds::DataHandle, [272](#)
- DataInfo
 - Bds::DataInfo, [273](#)
- dataOpen
 - Bds::AdminAccess, [68](#)
 - Bds::DataAccess, [145](#)
 - Bds::DataAddAccess, [161](#)
- DataOrder
 - Bds::DataFile, [196](#)
- DataOrderAll
 - Bds::DataFile, [196](#)
- DataOrderChannel
 - Bds::DataFile, [196](#)
- DataOrderSample
 - Bds::DataFile, [196](#)
- DataOrderUnknown
 - Bds::DataFile, [196](#)
- dataPutBlock
 - Bds::AdminAccess, [68](#)
 - Bds::DataAddAccess, [162](#)
- dataRead
 - Bds::DataFormat, [268](#)
- dataRealtimeConfig
 - Bds::AdminAccess, [68](#)
 - Bds::DataAccess, [146](#)
 - Bds::DataAddAccess, [162](#)
- dataRealtimeGet
 - Bds::AdminAccess, [68](#)
 - Bds::DataAccess, [146](#)
 - Bds::DataAddAccess, [162](#)
- dataSearch
 - Bds::AdminAccess, [69](#)
 - Bds::DataAccess, [146](#)
 - Bds::DataAddAccess, [162](#)
- dataSeekBlock
 - Bds::AdminAccess, [69](#)
 - Bds::DataAccess, [146](#)
 - Bds::DataAddAccess, [162](#)
- dataSetInfo
 - Bds::AdminAccess, [69](#)
 - Bds::DataAddAccess, [163](#)
- dataSize
 - Bds::CdDataChannel, [108](#)
- dataType
 - Bds::Channel, [124](#)
 - Bds::SpecialChannel, [353](#)
- datatype
 - Bds::DataFileCssData, [227](#)
- dataTypes
 - Bds::Selection, [333](#)
- dataWrite
 - Bds::DataFormat, [268](#)
- datum
 - Bds::Location, [298](#)
- dead
 - Bds::CdFlag, [111](#)
- DEBUG
 - BdsDataFileGcf.cpp, [376](#)
 - BdsDataFileSeed.cpp, [389](#)
- DEBUG_BLOCKETTE
 - BdsDataFileSeed.cpp, [389](#)
- DEBUG_BLOCKS
 - BdsDataFileSeed.cpp, [389](#)
- DEBUG_VELATRACK
 - BdsDataFileAd22.cpp, [366](#)
- decimation
 - Bds::Response, [324](#)
- decimationCorr
 - Bds::Response, [324](#)
- decimationDelay
 - Bds::Response, [324](#)
- decimationOffset
 - Bds::Response, [325](#)
- DefaultBlockSize
 - Bds::DataFileBds, [213](#)
- deletedFiles
 - Bds::CleanOptions, [136](#)
- depth
 - Bds::Calibration, [103](#)
- description
 - Bds::ChangeGroup, [120](#)
 - Bds::Channel, [124](#)
 - Bds::DataFormat, [268](#)
 - Bds::DataInfo, [274](#)
 - Bds::Event, [282](#)
 - Bds::Group, [292](#)
 - Bds::Log, [302](#)
 - Bds::Network, [307](#)
 - Bds::Note, [312](#)
 - Bds::Response, [325](#)
 - Bds::Source, [346](#)
 - Bds::SpecialChannel, [353](#)
 - Bds::Station, [355](#)
- destination
 - Bds::CdPacketData, [113](#)
- Digitiser
 - Bds::Digitiser, [277](#)
- digitiser
 - Bds::ChannelInfo, [127](#)

- digitiserDelete
 - Bds::AdminAccess, [69](#)
- digitiserGet
 - Bds::AdminAccess, [69](#)
 - Bds::DataAccess, [146](#)
 - Bds::DataAddAccess, [163](#)
- digitiserGetList
 - Bds::AdminAccess, [70](#)
 - Bds::DataAccess, [147](#)
 - Bds::DataAddAccess, [163](#)
- digitiserId
 - Bds::ChannelInstrument, [133](#)
 - Bds::Selection, [333](#)
- digitiserUpdate
 - Bds::AdminAccess, [70](#)
- dirName
 - Bds::DataFileCssData, [227](#)
- dl2printf
 - BdsDataFileBds.cpp, [369](#)
- dl3printf
 - BdsDataFileBds.cpp, [369](#)
- dlprintf
 - BdsDataFileBds.cpp, [369](#)
- docFormat
 - Bds::Note, [312](#)
- docUrl
 - Bds::Note, [312](#)
- dprintf
 - BdsDataFileCd.cpp, [373](#)
 - BdsDataFileIdc.cpp, [377](#)
 - BdsDataFileResponse.cpp, [381](#)
- dump
 - Bds::BdsDataPacket, [90](#)
- duplicateCheck
 - Bds::DataFile, [198](#)
- duplicateDump
 - Bds, [43](#)
- elevation
 - Bds::Event, [282](#)
 - Bds::Location, [299](#)
- email
 - Bds::User, [361](#)
- enabled
 - Bds::User, [361](#)
- end
 - Bds::DataFile, [198](#)
 - Bds::DataFileAscii, [207](#)
 - Bds::DataFileIms, [235](#)
 - Bds::DataFileLog, [245](#)
 - Bds::DataFileSeed, [255](#)
- endTime
 - Bds::AccessGroup, [53](#)
 - Bds::BdsDataBlockPos, [88](#)
 - Bds::BdsDataPacketHeader, [92](#)
 - Bds::BdsDataSegment, [94](#)
 - Bds::Calibration, [103](#)
 - Bds::Channel, [124](#)
 - Bds::ChannelInfo, [127](#)
 - Bds::ChannelInstrument, [133](#)
 - Bds::DataAvail, [172](#)
 - Bds::DataAvailChan, [174](#)
 - Bds::DataBlock, [176](#)
 - Bds::DataBlockPos, [180](#)
 - Bds::DataChannel, [184](#)
 - Bds::DataFileCssData, [227](#)
 - Bds::DataFileInfo, [240](#)
 - Bds::DataInfo, [274](#)
 - Bds::Digitiser, [278](#)
 - Bds::Event, [282](#)
 - Bds::Location, [299](#)
 - Bds::Note, [312](#)
 - Bds::Response, [325](#)
 - Bds::Selection, [333](#)
 - Bds::SelectionInfo, [338](#)
 - Bds::Sensor, [342](#)
 - Bds::SourcePriority, [349](#)
 - Bds::SpecialChannel, [353](#)
 - Bds::TimePeriod, [358](#)
- error
 - Bds::FirEntry, [289](#)
- ErrorDataQuality
 - Bds, [30](#)
- ErrorFormatNoDataFormat
 - BdsDataFileCd.cpp, [374](#)
- ErrorNoMetaData
 - Bds, [30](#)
- errorNumber
 - Bds::Note, [312](#)
- Errors
 - Bds, [30](#)
- ErrorSlaveMode
 - Bds, [30](#)
- ErrorTimeStamp
 - Bds, [30](#)
- ErrorValidate
 - Bds, [30](#)
- ErrorValidateBdsFudge
 - Bds, [30](#)
- ErrorValidateDuplicate
 - Bds, [30](#)
- ErrorValidateFilenameTime
 - Bds, [30](#)
- ErrorValidateFix
 - Bds, [30](#)
- ErrorValidateMetaData
 - Bds, [30](#)
- ErrorValidateMissingBlocks
 - Bds, [30](#)
- ErrorValidateReorder
 - Bds, [30](#)
- ErrorValidateTimeBackwards
 - Bds, [30](#)
- Event
 - Bds::Event, [281](#)
- eventDelete
 - Bds::AdminAccess, [70](#)

- Bds::DataAddAccess, 163
- eventGetList
 - Bds::AdminAccess, 70
 - Bds::DataAccess, 147
 - Bds::DataAddAccess, 163
- eventId
 - Bds::Note, 312
 - Bds::Selection, 334
- eventTime
 - Bds::Event, 282
- eventUpdate
 - Bds::AdminAccess, 70
 - Bds::DataAddAccess, 164
- excludeChannels
 - Bds::Selection, 334
- extension
 - Bds::DataFormat, 268
- extra
 - Bds::Event, 283
- extraCall
 - Bds::AdminAccess, 71
- Fap
 - Bds::Fap, 286
- faps
 - Bds::Response, 325
- FeatureCanRead
 - Bds::DataFile, 196
- FeatureCanWrite
 - Bds::DataFile, 196
- FeatureNone
 - Bds::DataFile, 196
- Features
 - Bds::DataFile, 196
- file
 - Bds::DataFileCssData, 228
- FileHeaderType
 - Bds, 30
- FileHeaderType_Standard
 - Bds, 30
- FileHeaderType_TapeDigitiser
 - Bds, 30
- fileName
 - Bds::DataFileCssData, 228
- fileNameProcess
 - Bds::DataFile, 198
- fileNameTime
 - Bds, 44
- fileOffset
 - Bds::DataFileCssData, 228
- FileSampleType
 - Bds, 31
- FileSampleType_Float32
 - Bds, 31
- FileSampleType_Float64
 - Bds, 31
- FileSampleType_Int16
 - Bds, 31
- FileSampleType_Int32
 - Bds, 31
- FileSampleType_Unknown
 - Bds, 31
- FILL_BLOCKS
 - BdsDataFileSeed.cpp, 389
- findFormat
 - Bds::DataFormatAll, 270
- Fir
 - Bds::Fir, 287
- fir
 - Bds::Response, 325
- FirEntry
 - Bds::FirEntry, 288
- fixedString
 - Bds, 44
- fixedWidthValue
 - Bds, 44
- flush
 - Bds::DataFile, 198
 - Bds::DataFileBds, 214
- format
 - Bds::DataFileInfo, 240
 - Bds::GcfChannel, 289
- formatGet
 - Bds::DataFormatAll, 270
- formatGetExtension
 - Bds::DataFormatAll, 271
- formatList
 - Bds::DataFormatAll, 271
- frameLength
 - Bds::CdDataFormatFrame_1v0, 110
- frameType
 - Bds::CdDataFormatFrame_1v0, 110
 - Bds::CdPacketData, 113
- frequency
 - Bds::Fap, 286
- frequencyLowerBound
 - Bds::Polynomial, 319
- frequencyUpperBound
 - Bds::Polynomial, 319
- fromSeedTimeString
 - Bds, 44
- gain
 - Bds::Digitiser, 278
 - Bds::Response, 325
 - Bds::Sensor, 342
- gainFrequency
 - Bds::Response, 326
- gainUnits
 - Bds::Sensor, 342
- getDataOrder
 - Bds::DataFile, 199
 - Bds::DataFileAd22, 205
 - Bds::DataFileAscii, 207
 - Bds::DataFileBdrs, 210
 - Bds::DataFileBds, 215
 - Bds::DataFileCd, 221
 - Bds::DataFileCss, 224

- Bds::DataFileGcf, 231
- Bds::DataFileImgs, 235
- Bds::DataFileLac, 243
- Bds::DataFileLog, 245
- Bds::DataFileSeed, 255
- Bds::DataFileWra, 263
- Bds::DataFileWraAgso, 266
- getDiskBlockSize
 - Bds::DataFileBds, 215
- getDouble
 - Bds::BdsSeedType, 98
- getErrorNumber
 - Bds::DataError, 190
- getFeatures
 - Bds::DataFile, 199
 - Bds::DataFileAd22, 205
 - Bds::DataFileAscii, 207
 - Bds::DataFileBdrs, 210
 - Bds::DataFileCd, 221
 - Bds::DataFileCss, 224
 - Bds::DataFileGcf, 231
 - Bds::DataFileIdc, 233
 - Bds::DataFileImgs, 236
 - Bds::DataFileLac, 243
 - Bds::DataFileLog, 245
 - Bds::DataFileResponse, 250
 - Bds::DataFileSac, 252
 - Bds::DataFileSeed, 255
 - Bds::DataFileStationXml, 259
 - Bds::DataFileWra, 263
 - Bds::DataFileWraAgso, 266
- getFileName
 - Bds::DataFile, 199
- getFilePosition
 - Bds::DataFile, 199
- getFixesInfo
 - Bds::DataFile, 199
 - Bds::DataFileAd22, 205
 - Bds::DataFileBdrs, 210
 - Bds::DataFileCd, 222
 - Bds::DataFileGcf, 231
 - Bds::DataFileLac, 243
 - Bds::DataFileSeed, 255
 - Bds::DataFileWra, 263
- getFormat
 - Bds::DataFile, 200
- getFormats
 - Bds::DataFile, 200
 - Bds::DataFileAd22, 205
 - Bds::DataFileAscii, 208
 - Bds::DataFileBdrs, 211
 - Bds::DataFileBds, 215
 - Bds::DataFileBknas, 219
 - Bds::DataFileCd, 222
 - Bds::DataFileCss, 224
 - Bds::DataFileGcf, 231
 - Bds::DataFileIdc, 233
 - Bds::DataFileImgs, 236
 - Bds::DataFileLac, 243
 - Bds::DataFileLog, 246
 - Bds::DataFileSeed, 255
 - Bds::DataFileStationXml, 259
 - Bds::DataFileTapeDigitiser, 261
 - Bds::DataFileWra, 263
 - Bds::DataFileWraAgso, 266
- getHeader
 - Bds::BdsDataPacket, 90
- getHexString
 - Bds, 44
- getInfo
 - Bds::DataFile, 200
 - Bds::DataFileAd22, 205
 - Bds::DataFileBdrs, 211
 - Bds::DataFileBds, 215
 - Bds::DataFileCd, 222
 - Bds::DataFileCss, 224
 - Bds::DataFileGcf, 231
 - Bds::DataFileLac, 243
 - Bds::DataFileLog, 246
 - Bds::DataFileSeed, 256
 - Bds::DataFileTapeDigitiser, 261
 - Bds::DataFileWra, 264
 - Bds::DataFileWraAgso, 266
- getInt
 - Bds::BdsSeedType, 98
- getMember
 - Bds::AccessGroup, 52
 - Bds::Calibration, 101
 - Bds::Change, 116
 - Bds::ChangeGroup, 119
 - Bds::Channel, 123
 - Bds::ChannelInstrument, 131
 - Bds::DataChannel, 183
 - Bds::DataFileInfo, 239
 - Bds::Digitiser, 277
 - Bds::Group, 291
 - Bds::ListRange, 294
 - Bds::Location, 297
 - Bds::Log, 301
 - Bds::Network, 306
 - Bds::Note, 310
 - Bds::Sensor, 341
 - Bds::Source, 345
 - Bds::SourcePriority, 348
 - Bds::SpecialChannel, 352
 - Bds::TimePeriod, 357
 - Bds::User, 360
- getMembers
 - Bds::AccessGroup, 52
 - Bds::Calibration, 101
 - Bds::Change, 116
 - Bds::ChangeGroup, 119
 - Bds::Channel, 123
 - Bds::ChannelInstrument, 132

- Bds::DataChannel, 183
- Bds::DataFileInfo, 239
- Bds::Digitiser, 277
- Bds::Group, 291
- Bds::ListRange, 294
- Bds::Location, 297
- Bds::Log, 301
- Bds::Network, 307
- Bds::Note, 310
- Bds::Sensor, 341
- Bds::Source, 345
- Bds::SourcePriority, 349
- Bds::SpecialChannel, 352
- Bds::TimePeriod, 357
- Bds::User, 360
- getMetaData
 - Bds::DataFile, 200
 - Bds::DataFileIdc, 233
 - Bds::DataFileIms, 236
 - Bds::DataFileResponse, 250
 - Bds::DataFileSac, 252
 - Bds::DataFileSeed, 256
 - Bds::DataFileStationXml, 259
- getSelectionInfo
 - Bds::AdminAccess, 71
 - Bds::DataAccess, 147
 - Bds::DataAddAccess, 164
- getSelections
 - Bds::AdminAccess, 71
 - Bds::DataAccess, 147
 - Bds::DataAddAccess, 164
- getString
 - Bds::BdsSeedType, 98
 - Bds::DataError, 190
 - Bds::ResponseObj, 330
- getStringVariable
 - Bds::BdsSeedType, 98
- getTitle
 - Bds::DataError, 190
- getType
 - Bds::AccessGroup, 52
 - Bds::Calibration, 101
 - Bds::Change, 116
 - Bds::ChangeGroup, 119
 - Bds::Channel, 123
 - Bds::ChannelInstrument, 132
 - Bds::DataChannel, 183
 - Bds::DataFileInfo, 239
 - Bds::Digitiser, 277
 - Bds::Group, 292
 - Bds::ListRange, 294
 - Bds::Location, 297
 - Bds::Log, 302
 - Bds::Network, 307
 - Bds::Note, 311
 - Bds::Sensor, 341
 - Bds::Source, 346
 - Bds::SourcePriority, 349
 - Bds::SpecialChannel, 352
 - Bds::TimePeriod, 358
 - Bds::User, 360
- getUInt
 - Bds::BdsSeedType, 99
- getVersion
 - Bds::AdminAccess, 71
 - Bds::DataAccess, 147
 - Bds::DataAddAccess, 164
- Group
 - Bds::Group, 291
- group
 - Bds::AccessGroup, 53
 - Bds::Group, 292
- groupDelete
 - Bds::AdminAccess, 71
- groupGetList
 - Bds::AdminAccess, 72
 - Bds::DataAccess, 148
 - Bds::DataAddAccess, 164
- groups
 - Bds::User, 361
- groupUpdate
 - Bds::AdminAccess, 72
- handle
 - Bds::DataHandle, 272
- header
 - Bds::BdsDataBlock, 85
- horizontalAngle
 - Bds::Calibration, 103
- htonll
 - BdsDataFileCd.cpp, 373
- id
 - Bds::AccessGroup, 53
 - Bds::Calibration, 103
 - Bds::Change, 117
 - Bds::ChangeGroup, 120
 - Bds::Channel, 125
 - Bds::ChannelInstrument, 133
 - Bds::DataChannel, 184
 - Bds::DataFileInfo, 240
 - Bds::Digitiser, 278
 - Bds::Event, 283
 - Bds::Group, 293
 - Bds::Location, 299
 - Bds::Log, 302
 - Bds::Network, 308
 - Bds::Note, 313
 - Bds::Response, 326
 - Bds::Selection, 334
 - Bds::Sensor, 342
 - Bds::Source, 347
 - Bds::SourcePriority, 350
 - Bds::SpecialChannel, 353
 - Bds::Station, 355
 - Bds::User, 362
- importFilename

- Bds::DataChannel, 184
- Bds::Note, 313
- importFormat
 - Bds::DataChannel, 185
- importStartTime
 - Bds::DataChannel, 185
- importTime
 - Bds::DataFileInfo, 240
- importUserId
 - Bds::DataFileInfo, 240
- INCLUDE_CHANNEL_AUTH
 - BdsDataFileCd.cpp, 373
- info
 - Bds::DataBlock, 176
 - Bds::DataChannel, 185
 - Bds::DataInfo, 274
- infoExtra
 - Bds::DataInfo, 275
- init
 - Bds::DataFile, 200
- initialSamplingFrequency
 - Bds::Digitiser, 279
- inputUnits
 - Bds::Response, 326
- inputUnitsDesc
 - Bds::Response, 326
- instType
 - Bds::DataFileCssData, 228
- jdate
 - Bds::DataFileCssData, 228
- latitude
 - Bds::Event, 283
 - Bds::Location, 299
- LDEBUG
 - BdsDataFileBds.cpp, 369
 - BdsDataFileCd.cpp, 373
 - BdsDataFileIdc.cpp, 377
 - BdsDataFileResponse.cpp, 381
- LDEBUG2
 - BdsDataFileBds.cpp, 369
- LDEBUG3
 - BdsDataFileBds.cpp, 369
- length
 - Bds::BdsDataBlockHeader, 86
 - Bds::BdsDataPacketHeader, 92
- ListRange
 - Bds::ListRange, 294
- loadDate
 - Bds::DataFileCssData, 228
- Location
 - Bds::Location, 297
- location
 - Bds::DataFileInfo, 241
- locationDelete
 - Bds::AdminAccess, 72
- locationGetList
 - Bds::AdminAccess, 72
- Bds::DataAccess, 148
- Bds::DataAddAccess, 165
- LocationSelect
 - Bds, 31
- locationSelect
 - Bds::Selection, 334
- LocationSelectAll
 - Bds, 31
- LocationSelectChannel
 - Bds, 31
- LocationSelectStation
 - Bds, 31
- locationUpdate
 - Bds::AdminAccess, 72
- Log
 - Bds::Log, 301
- logAppend
 - Bds::AdminAccess, 73
 - Bds::DataAccess, 148
 - Bds::DataAddAccess, 165
- logDelete
 - Bds::AdminAccess, 73
- logGetList
 - Bds::AdminAccess, 73
- logs
 - Bds::CleanOptions, 137
- LogSelect
 - Bds::LogSelect, 304
- logUpdate
 - Bds::AdminAccess, 73
 - Bds::DataAccess, 148
 - Bds::DataAddAccess, 165
- longitude
 - Bds::Event, 283
 - Bds::Location, 299
- magnitude
 - Bds::Event, 283
- magnitudeUnits
 - Bds::Event, 283
- maxFrameLength
 - Bds::CdDataFormatFrame_1v0, 110
- maximumError
 - Bds::Polynomial, 319
- measured
 - Bds::Response, 326
- measurementMethod
 - Bds::PolynomialEntry, 321
- mergeDataInfo
 - Bds::DataError, 190
- metadataGetChannelInfo
 - Bds::AdminAccess, 73
 - Bds::DataAccess, 148
 - Bds::DataAddAccess, 165
- metadataGetFormatted
 - Bds::AdminAccess, 74
 - Bds::DataAccess, 149
 - Bds::DataAddAccess, 165
- metadataRead

- Bds::DataFormat, 269
- metadataWrite
 - Bds::DataFormat, 269
- minusError
 - Bds::PolynomialEntry, 321
- Mode
 - Bds, 31
- mode
 - Bds::CdDataChannel, 108
- ModeMaster
 - Bds, 31
- modeSet
 - Bds::AdminAccess, 74
 - Bds::DataAccess, 149
 - Bds::DataAddAccess, 166
- ModeSlave
 - Bds, 31
- modeSnapshotPause
 - Bds::AdminAccess, 74
 - Bds::DataAccess, 149
 - Bds::DataAddAccess, 166
- MSRecord
 - BdsDataFileSeed.h, 390
- msrFileWrite
 - Bds::DataFileSeed, 256
- MULTIPLE_SEGMENT
 - BdsDataFileCd.cpp, 373
- name
 - Bds::Calibration, 104
 - Bds::CdChannel_1v0, 107
 - Bds::Digitiser, 279
 - Bds::Response, 326
 - Bds::Selection, 334
 - Bds::Sensor, 343
 - Bds::Station, 356
 - Bds::User, 362
- names
 - Bds::DataFormat, 269
- Network
 - Bds::Network, 306
- network
 - Bds::AccessGroup, 53
 - Bds::ArrayChannel, 84
 - Bds::Calibration, 104
 - Bds::Channel, 125
 - Bds::ChannelName, 135
 - Bds::DataAvailChan, 174
 - Bds::DataBlockChannel, 178
 - Bds::DataChannel, 185
 - Bds::Event, 284
 - Bds::Location, 299
 - Bds::Network, 308
 - Bds::Note, 313
 - Bds::Response, 327
 - Bds::SelectionChannel, 336
 - Bds::SpecialChannel, 353
 - Bds::Station, 356
- networkDelete
 - Bds::AdminAccess, 74
- networkGetList
 - Bds::AdminAccess, 74
 - Bds::DataAccess, 149
 - Bds::DataAddAccess, 166
- NetworkNameLen
 - Bds, 48
- networks
 - Bds::SelectionInfo, 339
- networkUpdate
 - Bds::AdminAccess, 75
- node_types
 - Bds, 48
- Note
 - Bds::Note, 310
- noteDelete
 - Bds::AdminAccess, 75
- noteGetList
 - Bds::AdminAccess, 75
 - Bds::DataAccess, 149
 - Bds::DataAddAccess, 166
- noteReadDocument
 - Bds::AdminAccess, 75
 - Bds::DataAccess, 150
 - Bds::DataAddAccess, 166
- notes
 - Bds::Event, 284
- noteUpdate
 - Bds::AdminAccess, 75
 - Bds::DataAccess, 150
 - Bds::DataAddAccess, 167
- noteWriteDocument
 - Bds::AdminAccess, 76
 - Bds::DataAccess, 150
 - Bds::DataAddAccess, 167
- nsamp
 - Bds::DataFileCssData, 228
- ntohl
 - BdsDataFileCd.cpp, 373
- nullString
 - Bds, 44
- num
 - Bds::DataError, 190
- number
 - Bds::ListRange, 295
- numberChannels
 - Bds::Digitiser, 279
 - Bds::Sensor, 343
- numBlocks
 - Bds::BdsDataSegment, 94
 - Bds::DataChannel, 185
- numChannels
 - Bds::BdsDataBlockPos, 88
 - Bds::BdsDataStreamlet, 96
 - Bds::CdDataFormatFrame_1v0, 110
 - Bds::CdPacketData, 113
- numDataChannels
 - Bds::SelectionInfo, 339

- numSamples
 - Bds::BdsDataBlockPos, 88
 - Bds::BdsDataSegment, 94
 - Bds::CdDataChannel, 109
 - Bds::DataBlockPos, 180
 - Bds::DataChannel, 185
- ochannel
 - Bds::DataError, 192
- odescription
 - Bds::DataError, 192
- oendTime
 - Bds::DataError, 192
- oerrorNumber
 - Bds::DataError, 192
- ofile
 - Bds::DataFile, 203
- ofilename
 - Bds::DataFile, 203
- ofilename
 - Bds::DataError, 192
- ofilenameTime
 - Bds::DataFile, 203
- oformat
 - Bds::DataFile, 203
- oignoreBlockList
 - Bds::DataFileOptions, 249
- oldId
 - Bds::Sensor, 343
- omode
 - Bds::DataFile, 203
- omsrErr
 - Bds::DataFileSeed, 257
- onetwork
 - Bds::DataError, 192
- onoLock
 - Bds::DataFileSeed, 258
- ooptionList
 - Bds::DataFileOptions, 249
- open
 - Bds::DataFile, 201
 - Bds::DataFileAscii, 208
 - Bds::DataFileBds, 215
 - Bds::DataFileBknas, 219
 - Bds::DataFileIms, 236
 - Bds::DataFileLog, 246
 - Bds::DataFileTapeDigitiser, 261
- operator int
 - Bds::DataError, 190
 - Bds::DataFileOptions, 248
- operator<
 - Bds::BdsDataBlockPos, 87
 - Bds::BdsDataSegment, 93
 - Bds::DataBlockPos, 179
- operator |=
 - Bds::DataFileOptions, 248
- order
 - Bds::DataBlockPos, 180
- osource
 - Bds::DataError, 193
- ostartTime
 - Bds::DataError, 193
- ostation
 - Bds::DataError, 193
- otitle
 - Bds::DataError, 193
- ouser
 - Bds::DataError, 193
- outputUnits
 - Bds::Response, 327
- outputUnitsDesc
 - Bds::Response, 327
- packetNumber
 - Bds::BdsDataStreamlet, 96
- packetOffset
 - Bds::BdsDataBlockHeader, 86
- packetRead
 - Bds::DataFileBds, 216
- packetWrite
 - Bds::DataFileBds, 216
- PackFormat
 - Bds::DataFileBds, 214
- PackFormat_CM
 - Bds::DataFileBds, 214
- PackFormat_SM
 - Bds::DataFileBds, 214
- PackFormat_SM_CC
 - Bds::DataFileBds, 214
- PackFormat_Unknown
 - Bds::DataFileBds, 214
- parseStringFixedFields
 - BdsDataFileWraAgso.cpp, 386
- password
 - Bds::User, 362
- period
 - Bds::CdDataChannel, 109
 - Bds::CdDataFormatFrame_1v0, 110
 - Bds::CdPacketData, 113
- phase
 - Bds::Fap, 286
- plusError
 - Bds::PolynomialEntry, 321
- Point
 - Bds::Point, 315
- poles
 - Bds::PoleZero, 316
- PoleZero
 - Bds::PoleZero, 316
- poleZeros
 - Bds::Response, 327
- Polynomial
 - Bds::Polynomial, 318
- polynomial
 - Bds::Response, 327
- PolynomialEntry
 - Bds::PolynomialEntry, 320
- position

- Bds::BdsDataBlockPos, 88
- Bds::BdsDataStreamlet, 96
- Bds::DataBlockPos, 180
- Priority
 - Bds, 31
- priority
 - Bds::Log, 303
 - Bds::LogSelect, 304
 - Bds::SourcePriority, 350
- PriorityHigh
 - Bds, 33
- PriorityLow
 - Bds, 33
- PriorityNormal
 - Bds, 33
- range
 - Bds::Selection, 334
- rawCalibrationFactor
 - Bds::Calibration, 104
- rawCalibrationFrequency
 - Bds::Calibration, 104
- rawCalibrationUnits
 - Bds::Calibration, 104
- readData
 - Bds::DataCollate, 187
 - Bds::DataFile, 201
 - Bds::DataFileAd22, 205
 - Bds::DataFileBdrs, 211
 - Bds::DataFileBds, 216
 - Bds::DataFileCd, 222
 - Bds::DataFileCss, 224
 - Bds::DataFileGcf, 232
 - Bds::DataFileLac, 243
 - Bds::DataFileLog, 246
 - Bds::DataFileSeed, 256
 - Bds::DataFileTapeDigitiser, 261
 - Bds::DataFileWra, 264
 - Bds::DataFileWraAgso, 266
- ReadOptionDeleteDuplicates
 - Bds::DataFile, 197
- ReadOptionFileNameProcess
 - Bds::DataFile, 197
- ReadOptionFixCorruptions
 - Bds::DataFile, 197
- ReadOptionFixSampleRate
 - Bds::DataFile, 197
- ReadOptionIgnoreSamplerate
 - Bds::DataFile, 197
- ReadOptionInfoExtra
 - Bds::DataFile, 197
- ReadOptionNone
 - Bds::DataFile, 197
- ReadOptionPrintBlocks
 - Bds::DataFile, 197
- ReadOptionReorder
 - Bds::DataFile, 197
- ReadOptionsList
 - Bds::DataFile, 196
- ReadOptionValidate
 - Bds::DataFile, 197
- record_handler
 - Bds, 45
- ref
 - Bds::DataBlockPos, 180
- removeCR
 - Bds, 45
- reset
 - Bds::BdsDataPacket, 90
- Response
 - Bds::Response, 323
- responseDelete
 - Bds::AdminAccess, 76
- responseGetList
 - Bds::AdminAccess, 76
 - Bds::DataAccess, 150
 - Bds::DataAddAccess, 167
- ResponseObj
 - Bds::ResponseObj, 330
- responses
 - Bds::ChannelInfo, 128
- responseSort
 - Bds, 45
- responseUpdate
 - Bds::AdminAccess, 76
- reverse
 - Bds::ListRange, 295
- ROUND_TIMESTAMPS_US
 - BdsDataFileSeed.cpp, 389
- roundDigits
 - Bds, 45
- rowId
 - Bds::Change, 117
- sampleBigEndian
 - Bds::DataFileCssData, 228
- SampleFormat
 - Bds, 33
- sampleFormat
 - Bds::DataChannel, 186
 - Bds::DataFileCssData, 229
- SampleFormatFloat32
 - Bds, 33
- SampleFormatFloat64
 - Bds, 33
- SampleFormatInt16
 - Bds, 33
- SampleFormatInt24
 - Bds, 33
- SampleFormatInt32
 - Bds, 33
- SampleFormatUnknown
 - Bds, 33
- sampleRate
 - Bds::BdsDataSegment, 94
 - Bds::DataChannel, 186
 - Bds::DataFileCssData, 229
 - Bds::GcfChannel, 290

- Bds::Response, [327](#)
- sampleSize
 - Bds::DataFileCssData, [229](#)
- samplingFrequency
 - Bds::Calibration, [104](#)
- Scale
 - Bds, [49](#)
- seedChannelDataType
 - Bds, [45](#)
- seedChannelInstrumentCode
 - Bds, [45](#)
- seedIcodeToDataTypes
 - Bds, [49](#)
- seedTime
 - Bds, [46](#)
- seedTimeString
 - Bds, [46](#)
- seekBlock
 - Bds::DataFile, [201](#)
 - Bds::DataFileBds, [216](#)
- segment
 - Bds::BdsDataBlockPos, [88](#)
- SEGMENT_GAP
 - BdsDataFileCd.cpp, [374](#)
- segmentNumber
 - Bds::DataBlock, [176](#)
- segments
 - Bds::BdsDataStreamlet, [96](#)
 - Bds::DataAvailChan, [174](#)
- segtype
 - Bds::DataFileCssData, [229](#)
- Selection
 - Bds::Selection, [332](#)
- SelectionChannel
 - Bds::SelectionChannel, [336](#)
- SelectionGroup
 - Bds, [33](#)
- SelectionGroupData
 - Bds, [33](#)
- SelectionGroupDataWithCount
 - Bds, [33](#)
- SelectionGroupMetaData
 - Bds, [33](#)
- SelectionInfo
 - Bds::SelectionInfo, [338](#)
- Sensor
 - Bds::Sensor, [341](#)
- sensor
 - Bds::ChannelInfo, [128](#)
- sensorDelete
 - Bds::AdminAccess, [76](#)
- sensorGet
 - Bds::AdminAccess, [77](#)
 - Bds::DataAccess, [150](#)
 - Bds::DataAddAccess, [167](#)
- sensorGetList
 - Bds::AdminAccess, [77](#)
 - Bds::DataAccess, [151](#)
- Bds::DataAddAccess, [167](#)
- sensorId
 - Bds::ChannelInstrument, [133](#)
 - Bds::Selection, [335](#)
- sensorOldId
 - Bds::Selection, [335](#)
- sensorUpdate
 - Bds::AdminAccess, [77](#)
- sequence
 - Bds::BdsDataPacketHeader, [92](#)
- sequenceNum
 - Bds::CdPacketData, [114](#)
- serialNumber
 - Bds::Digitiser, [279](#)
 - Bds::Sensor, [343](#)
- series
 - Bds::CdPacketData, [114](#)
- serverConfigurationGet
 - Bds::AdminAccess, [77](#)
 - Bds::DataAccess, [151](#)
 - Bds::DataAddAccess, [168](#)
- set
 - Bds::DataError, [191](#)
 - Bds::DataFileCssData, [226](#)
- setByteOrder
 - Bds::CompressSteim1, [138](#)
- setChecksumAndLength
 - Bds::BdsDataPacket, [90](#)
- setDiskBlockSize
 - Bds::DataFileBds, [217](#)
- setFormat
 - Bds::DataFile, [201](#)
 - Bds::DataFileAscii, [208](#)
 - Bds::DataFileBds, [217](#)
 - Bds::DataFileLog, [246](#)
 - Bds::DataFileSeed, [256](#)
 - Bds::DataFileWra, [264](#)
- setHeader
 - Bds::BdsDataPacket, [90](#)
- setInfo
 - Bds::DataFile, [202](#)
 - Bds::DataFileAscii, [208](#)
 - Bds::DataFileBds, [217](#)
 - Bds::DataFileBknas, [220](#)
 - Bds::DataFileIdc, [233](#)
 - Bds::DataFileImms, [236](#)
 - Bds::DataFileLog, [247](#)
 - Bds::DataFileResponse, [250](#)
 - Bds::DataFileSac, [252](#)
 - Bds::DataFileSeed, [257](#)
 - Bds::DataFileStationXml, [259](#)
- setMember
 - Bds::AccessGroup, [52](#)
 - Bds::Calibration, [102](#)
 - Bds::Change, [116](#)
 - Bds::ChangeGroup, [119](#)
 - Bds::Channel, [123](#)
 - Bds::ChannelInstrument, [132](#)

- Bds::DataChannel, 183
- Bds::DataFileInfo, 239
- Bds::Digitiser, 278
- Bds::Group, 292
- Bds::ListRange, 294
- Bds::Location, 298
- Bds::Log, 302
- Bds::Network, 307
- Bds::Note, 311
- Bds::Sensor, 342
- Bds::Source, 346
- Bds::SourcePriority, 349
- Bds::SpecialChannel, 352
- Bds::TimePeriod, 358
- Bds::User, 361
- setMembers
 - Bds::AccessGroup, 53
 - Bds::Calibration, 102
 - Bds::Change, 116
 - Bds::ChangeGroup, 120
 - Bds::Channel, 123
 - Bds::ChannelInstrument, 132
 - Bds::DataChannel, 183
 - Bds::DataFileInfo, 239
 - Bds::Digitiser, 278
 - Bds::Group, 292
 - Bds::ListRange, 295
 - Bds::Location, 298
 - Bds::Log, 302
 - Bds::Network, 307
 - Bds::Note, 311
 - Bds::Sensor, 342
 - Bds::Source, 346
 - Bds::SourcePriority, 349
 - Bds::SpecialChannel, 352
 - Bds::TimePeriod, 358
 - Bds::User, 361
- setReadPositionToStart
 - Bds::DataFileBds, 217
- setString
 - Bds::DataError, 191
 - Bds::ResponseObj, 330
- setStringUser
 - Bds::DataError, 191
- setUser
 - Bds::AdminAccess, 77
 - Bds::DataAccess, 151
 - Bds::DataAddAccess, 168
- setUserReal
 - Bds::AdminAccess, 78
 - Bds::DataAccess, 151
 - Bds::DataAddAccess, 168
- setWritePositionForAppend
 - Bds::DataFileBds, 218
- shared
 - Bds::Digitiser, 279
 - Bds::Sensor, 343
- Source
 - Bds::Source, 345
- source
 - Bds::Calibration, 105
 - Bds::ChannelInfo, 128
 - Bds::ChannelInstrument, 133
 - Bds::ChannelName, 135
 - Bds::DataAvailChan, 174
 - Bds::DataBlockChannel, 178
 - Bds::DataChannel, 186
 - Bds::Event, 284
 - Bds::Note, 313
 - Bds::Response, 328
 - Bds::SelectionChannel, 336
 - Bds::Source, 347
 - Bds::SourcePriority, 350
- sourceDelete
 - Bds::AdminAccess, 78
- sourceGetList
 - Bds::AdminAccess, 78
 - Bds::DataAccess, 151
 - Bds::DataAddAccess, 168
- SourceLen
 - Bds, 49
- sourceMeta
 - Bds::Source, 347
- SourcePriority
 - Bds::SourcePriority, 348
- sourcePriorityDelete
 - Bds::AdminAccess, 78
- sourcePriorityGetList
 - Bds::AdminAccess, 78
 - Bds::DataAccess, 152
 - Bds::DataAddAccess, 168
- sourcePriorityUpdate
 - Bds::AdminAccess, 78
- sources
 - Bds::SelectionInfo, 339
- sourceUpdate
 - Bds::AdminAccess, 79
- spare0
 - Bds::CdChannel_1v0, 107
- spare1
 - Bds::CdChannel_1v0, 107
- SpecialChannel
 - Bds::SpecialChannel, 351
- specialChannelDelete
 - Bds::AdminAccess, 79
- specialChannelGetList
 - Bds::AdminAccess, 79
 - Bds::DataAccess, 152
 - Bds::DataAddAccess, 169
- specialChannelUpdate
 - Bds::AdminAccess, 79
- sqlQuery
 - Bds::AdminAccess, 79
- sta
 - Bds::DataFileCssData, 229
- stage

- Bds::Response, 328
- stageType
 - Bds::Response, 328
- start
 - Bds::DataFile, 202
 - Bds::DataFileAscii, 208
 - Bds::DataFileIms, 237
 - Bds::DataFileLog, 247
 - Bds::DataFileSeed, 257
 - Bds::ListRange, 295
- startTime
 - Bds::AccessGroup, 54
 - Bds::BdsDataBlockPos, 88
 - Bds::BdsDataPacketHeader, 92
 - Bds::BdsDataSegment, 94
 - Bds::Calibration, 105
 - Bds::CdDataChannel, 109
 - Bds::CdPacketData, 114
 - Bds::Channel, 125
 - Bds::ChannelInfo, 128
 - Bds::ChannelInstrument, 133
 - Bds::DataAvail, 172
 - Bds::DataAvailChan, 174
 - Bds::DataBlock, 177
 - Bds::DataBlockPos, 180
 - Bds::DataChannel, 186
 - Bds::DataFileCssData, 229
 - Bds::DataFileInfo, 241
 - Bds::DataInfo, 275
 - Bds::Digitiser, 279
 - Bds::Event, 284
 - Bds::Location, 300
 - Bds::LogSelect, 305
 - Bds::Note, 313
 - Bds::Response, 328
 - Bds::Selection, 335
 - Bds::SelectionInfo, 339
 - Bds::Sensor, 343
 - Bds::SourcePriority, 350
 - Bds::SpecialChannel, 354
 - Bds::TimePeriod, 358
- state
 - Bds::DataFileInfo, 241
- Station
 - Bds::Station, 355
- station
 - Bds::AccessGroup, 54
 - Bds::ArrayChannel, 84
 - Bds::Calibration, 105
 - Bds::CdDataChannel, 109
 - Bds::Channel, 125
 - Bds::ChannelInfo, 128
 - Bds::ChannelName, 135
 - Bds::DataAvailChan, 174
 - Bds::DataBlockChannel, 178
 - Bds::DataChannel, 186
 - Bds::Location, 300
 - Bds::Note, 313
 - Bds::Response, 328
 - Bds::SelectionChannel, 336
 - Bds::SpecialChannel, 354
- stationDelete
 - Bds::AdminAccess, 80
- stationGetList
 - Bds::AdminAccess, 80
 - Bds::DataAccess, 152
 - Bds::DataAddAccess, 169
- stationLocation
 - Bds::ChannelInfo, 128
- stationName
 - Bds::CdChannel_1v0, 107
- StationNameLen
 - Bds, 49
- stations
 - Bds::Network, 308
 - Bds::SelectionInfo, 339
- stationUpdate
 - Bds::AdminAccess, 80
- statisticsGet
 - Bds::AdminAccess, 80
 - Bds::DataAccess, 152
 - Bds::DataAddAccess, 169
- status
 - Bds::CdDataChannel, 109
- str
 - Bds::DataError, 191
- stream
 - Bds::DataFileInfo, 241
- streamId
 - Bds::GcfChannel, 290
- streamlet
 - Bds::BdsDataPacketHeader, 92
- streamletToChannel
 - Bds::DataFileBds, 218
- StreamsMax
 - Bds::DataFileBds, 213
- stringFormat
 - Bds, 46
- subSystem
 - Bds::Log, 303
 - Bds::LogSelect, 305
- symmetry
 - Bds::Response, 328
- synchronous
 - Bds::DataInfo, 275
- systemId
 - Bds::GcfChannel, 290
- table
 - Bds::Change, 117
- telephone
 - Bds::User, 362
- TEST_REORDER
 - BdsDataFileGcf.cpp, 376
- time
 - Bds::Change, 117
 - Bds::ChangeGroup, 120

- Bds::Log, 303
- timeAdded
 - Bds::Note, 314
- timeCompare
 - Bds::DataFile, 202
- TimePeriod
 - Bds::TimePeriod, 357
- TIMESTAMP_JITTER
 - BdsDataFileBds.cpp, 370
 - BdsDataFileCd.cpp, 374
- title
 - Bds::ChangeGroup, 120
 - Bds::Event, 284
 - Bds::Log, 303
 - Bds::Note, 314
- trailerOffset
 - Bds::CdPacketData, 114
- transactionEnd
 - Bds::AdminAccess, 80
- transactionStart
 - Bds::AdminAccess, 81
- transferType
 - Bds::Polynomial, 319
- type
 - Bds::BdsDataBlockHeader, 86
 - Bds::BdsDataPacketHeader, 92
 - Bds::Change, 117
 - Bds::ChangeGroup, 121
 - Bds::Digitiser, 280
 - Bds::Event, 284
 - Bds::GcfChannel, 290
 - Bds::Log, 303
 - Bds::LogSelect, 305
 - Bds::Note, 314
 - Bds::Response, 329
 - Bds::Sensor, 344
 - Bds::Station, 356
- unCompress
 - Bds::CompressSteim1, 138
- unitsCode
 - Bds, 46
- url
 - Bds::DataFileInfo, 241
- User
 - Bds::User, 360
- user
 - Bds::ChangeGroup, 121
 - Bds::Note, 314
 - Bds::User, 362
- userDelete
 - Bds::AdminAccess, 81
- userGet
 - Bds::AdminAccess, 81
 - Bds::DataAccess, 152
 - Bds::DataAddAccess, 169
- userGetFromId
 - Bds::AdminAccess, 81
 - Bds::DataAccess, 153
- Bds::DataAddAccess, 169
- userGetGroups
 - Bds::AdminAccess, 81
 - Bds::DataAccess, 153
 - Bds::DataAddAccess, 170
- userGetList
 - Bds::AdminAccess, 81
- userGetOptions
 - Bds::AdminAccess, 82
 - Bds::DataAccess, 153
 - Bds::DataAddAccess, 170
- userId
 - Bds::Event, 285
- userSet
 - Bds::AdminAccess, 82
 - Bds::DataAccess, 153
 - Bds::DataAddAccess, 170
- userSetOptions
 - Bds::AdminAccess, 82
 - Bds::DataAccess, 153
 - Bds::DataAddAccess, 170
- userUpdate
 - Bds::AdminAccess, 82
- validateChecksum
 - Bds::BdsDataPacket, 91
- validateUser
 - Bds::AdminAccess, 82
 - Bds::DataAccess, 154
 - Bds::DataAddAccess, 170
- validFrequencyUnits
 - Bds::Polynomial, 319
- verticalAngle
 - Bds::Calibration, 105
- warnings
 - Bds::DataInfo, 275
- waterDepth
 - Bds::Event, 285
- waterLevel
 - Bds::Calibration, 105
- wfid
 - Bds::DataFileCssData, 229
- writeData
 - Bds::DataFile, 202
 - Bds::DataFileAscii, 209
 - Bds::DataFileBds, 218
 - Bds::DataFileBknas, 220
 - Bds::DataFileIms, 237
 - Bds::DataFileLog, 247
 - Bds::DataFileSeed, 257
- WriteOptionNoMetadata
 - Bds::DataFile, 197
- WriteOptionNone
 - Bds::DataFile, 197
- WriteOptionSensorData
 - Bds::DataFile, 197
- WriteOptionsList
 - Bds::DataFile, 197

x

Bds::Point, [315](#)

y

Bds::Point, [315](#)

zeroed

Bds::CdFlag, [111](#)

zeros

Bds::PoleZero, [316](#)