BdsApi

2.2.0

Generated by Doxygen 1.8.15

Chapter 1

Main Page

Author

Dr Terry Barnaby

Version

2.2.0

Date

2020-06-23

1.1 Introduction

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

The API operates over a network type interface using an RPC type mechanism implemented by BEAM's BOAP RPC system. The BdsApi API makes use of the BEAM 'C++' class library for lower level and system independent functionality. The BEAM 'C++' class library provides a small set of low level 'C++' classes for strings, lists and system interface functions. It also implements the BOAP RPC mechanism used to implement the BdsApi. There is some brief information on the BEAM class library later on in this page and a full API definition is available in the beam-lib documentation.

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

This is the reference documentation for the BdsApi. An overall API description and programming manual is provided separately in: `BdsDevelopment.pdf`

1.2 Overview

The BdsApi has been developed using the BOAP (BEAM Object Access Protocol). This provides a simple but powerful Object Orientated RPC mechanism. The BdsApi is written in a high level interface definition language (IDL). The bidl tool generates the client and server side 'C++' interface and implementation files for the API. These are then provided as a set of 'C++' header files and a binary library file for the clients to link to. The BOAP system employs a simple BOAP name server process that provides a translation between object names and network I↩ PAddress/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
#ifndef 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;
}
/*****************************************************************************
 *  BdsMetaData1.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:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

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

Chapter 4

Class Index

4.1 Class List

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

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

Chapter 6

Namespace Documentation

6.1 Bds Namespace Reference

Classes

- class AccessGroup

  This holds information on data access groups.
- class AdminAccess

  This is the AdminAccess Access API interface.
- class ArrayChannel

  This class defines an arrays channel.
- struct BdsDataBlock

  BdsDataFileBds: internal fixed size BDS Data Block.
- struct BdsDataBlockHeader

  BdsDataFileBds: internal fixed size BDS Data Block header.
- class BdsDataBlockPos

  BdsDataFileBds: internal file storage data block position.
- class BdsDataPacket

  BdsDataFileBds: internal file storage packet.
- struct BdsDataPacketHeader

  BdsDataFileBds internal file storage packet header.
- class BdsDataSegment

  BdsDataFileBds: internal file storage data segment.
- class BdsDataStreamlet

  BdsDataFileBds: internal file storage data streamlet.
- class BdsSeedType

  BdsDataFileSeed internal parent for all SEED types.
- class Calibration

  This class defines a calibration setting.
- struct CdChannel_1v0

  BdsDataFile: Internal CD1.0 channel information.
- class CdDataChannel

  BdsDataFile: Internal CD channel information.
- struct CdDataFormatFrame_1v0

  BdsDataFile: Internal CD1.0 frame information.
- class CdFlag

BdsDataFile: Internal CD flag.

- class CdPacketData

  BdsDataFile: Internal CD data packet.
- class Change

  This holds information on a medatdata or sensor data change.
- class ChangeGroup

  This holds information on a set of Changes.
- class Channel

  This class defines a seismic data Channel.
- class ChannelInfo

  This class provides information on a channel.
- class ChannelInfos

  This class provides metadata information on a set of channels.
- class ChannelInstrument

  This class defines a Channel's instrument.
- class ChannelName

  This class defines a full channel name.
- class CleanOptions

  This defines the set of clean options used in the clean() function.
- class CompressSteim1

  Steim1 un-compress class.
- class DataAccess

  This is the Data Access API interface.
- class DataAddAccess

  This is the DataAdd Access API interface.
- class DataAvail

  This class provides availability information on a particular period of data.
- class DataAvailChan

  This class defines avaiilablity information on a set of data.
- class DataBlock

  This class provides the acual seismic data values contained within a single data block.
- class DataBlockPos

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

  This class defines information on a single channels set of data stored in a file.
- class DataCollate

  Not sure if this is used or what it does.
- class DataError

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

  This class defines the interface for generic data file access that all of the BDS data conterors share.
- class DataFileAd22

  Data file convertor for AD22 format files.
- class DataFileAscii

  Data file convertor for ASCII format files.
- class DataFileBdrs

  Data file convertor for BDRS format files.
- class DataFileBds

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

Data file convertor for BKNAS format files.

- class DataFileCd

  Data file convertor for CD1.0 and CD1.1 file formats.
- class DataFileCss

  Data file convertor for CSS format files.
- class DataFileCssData

  DataFileCss internal CSS data type.
- class DataFileGcf

  Data file convertor for GCF format files.
- class DataFileIms

  Data file convertor for IMS format files.
- class DataFileInfo

  This class defines information on a sensor data file.
- class DataFileLac

  Data file convertor for LAC format files.
- class DataFileLog

  Data file convertor for LOG format files.
- class DataFileOptions

  This defines a list of BDS data convtor options.
- class DataFileResponse

  This class defines the interface for generic response data file access.
- class DataFileSac

  Data file convertor for SAC format files.
- class DataFileSeed

  Data file convertor for SEED file formats.
- class DataFileTapeDigitiser

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

  Data file convertor for WRA format files.
- class DataFileWraAgso

  Data file convertor for WRA AGSO format files.
- class DataFormat

  This holds information on a seismic data format.
- class DataFormats

  This class defines the interface for generic data file access.
- class DataHandle

  This defines a handle to a sensor data stream/file when opened for read or write.
- class DataInfo

  This class defines information on a set of data.
- class Digitiser

  This class defines a seismic Digitiser.
- class Fap

  This class defines an entry in an Amplitude/Phase Response table.
- class Fir

  This class defines an FIR response table.
- class FirEntry

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

  DataFileGcf internal GCF channel information.
- class Group

  This holds information on a user security group.

- class ListRange

  This class defines an integer based range.
- class Location

  This class defines the physical location of a Station.
- class Log

  This holds information on a Log entry.
- class LogSelect

  This defines the selection cirteria when requesting a set of log entries.
- class Network

  This class defines a seismic Network organisation.
- class Note

  This holds information on a Note for general information.
- class Point

  This class defines an X,Y location.
- class PoleZero

  This class defines a Pole/Zero Response.
- class Response

  This class defines a seismic Response characteristic.
- class ResponseObj

  Response object adding string conversion.
- class Selection

  This class defines a generic metadata or seismic data selection.
- class SelectionChannel

  This class defines a channel for selection.
- class SelectionInfo

  This class defines the set of metadata or siesmic data selected when getSelectionInfo() is use.
- class Sensor

  This class defines a seismic Sensor.
- class Source

  This class defines a seismic data Source.
- class SourcePriority

  This class defines a Source Priority entry.
- class Station

  This class defines a seismic station.
- class TimePeriod

  This class defines a TimePeriod.
- class User

  This holds information on a user.

Enumerations

- enum Errors {
  ErrorNoMetaData = 64, ErrorDataQuality = 65, ErrorSlaveMode = 66, ErrorTimeStamp = 67,
  ErrorValidate = 80, ErrorValidateMissingBlocks = 81, ErrorValidateTimeBackwards = 82, ErrorValidateFilenameTime = 83,
  ErrorValidateMetaData = 84, ErrorValidateFix = 85, ErrorValidateDuplicate = 86, ErrorValidateReorder = 87,
  ErrorValidateBdsFudge = 88 }

  The System Error number list in addition to standard system error numbers.
- enum Priority { PriorityLow, PriorityNormal, PriorityHigh }

  Priority levels.
- enum Mode { ModeMaster, ModeSlave }

BdsServer mode.

- enum DataFlags { DataFlagNone = 0x00, DataFlagClipDataToTime = 0x01, DataFlagClipDataToChannels = 0x02, DataFlagMergeSegments = 0x04, DataFlagNoMetadata = 0x08 }

  Flags when opening data files.
- enum SelectionGroup { SelectionGroupData, SelectionGroupMetaData, SelectionGroupDataWithCount }

  The Selection group when making selections.
- enum SampleFormat { SampleFormatUnknown, SampleFormatInt16, SampleFormatInt32, SampleFormatFloat32, SampleFormatFloat64, SampleFormatInt24 }

  The actual format of a data sample.
- enum AvailType { AvailNone, AvailPartial, AvailFull }

  A flag defining the data availability state.
- enum BdsDataType { BdsDataTypeBlock = 0x42534442, BdsDataTypeInfo = 0x30534442, BdsDataTypeData = 0x31534442, BdsDataTypeInfoExtra = 0x32534442 }

  BdsDataFileBds: internal file block type field.
- enum FileHeaderType { FileHeaderType_Standard = 1, FileHeaderType_TapeDigitiser = 10 }
- enum FileSampleType { FileSampleType_Unknown, FileSampleType_Float32, FileSampleType_Float64, FileSampleType_Int16, FileSampleType_Int32 }

Functions

- void bdsDumpPoleZeros (PoleZero poleZeros)

  Debug print out a PoleZeros object.
- void bdsChannelGetTypeAux (**BString** name, **BString** &type, **BString** &aux)

  Get the channel type and aux fields from a generic channel name.
- **BString** bdsChannelGetName (**BString** type, **BString** aux)

  Create a full channel name from a channels type and aux fields.
- **BError** bdsDataInfoSetTimeRange (DataInfo &dataInfo)

  Restricts the time 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** bdsDataInfoMergeFlatten (DataInfo &dataInfo, const DataInfo &dataInfoAdd)

  Merges a DataInfo into another flattening the segments to 1 for use in dataOpen() calls.
- **BString** bdsStationAlias (Station station)

  Returns the station alias if set else its name.
- void bdsDumpSelection (Selection sel)

  Debug print out a Selection object.
- void bdsDumpDataInfo (DataInfo dataInfo, int includeInfo=0)

  Debug print out a DataInfo object.
- void bdsDumpChannelInfos (const ChannelInfos &channelInfos)

  Debug print out a ChannelInfos object.
- void bdsDumpData (const DataBlock &dataBlock, int nSamples=0)

  Debug print out a DataBlock object.
- **BString** bdsDataChannelInfo (const DataChannel &dataChannel)

  Returns a string representation of a DataChannel object.
- **BString** bdsDataChannelRef (const DataChannel &dataChannel)

Returns the string reference name of a DataChannel object.

- **BString** bdsDataChannelRef (const ChannelInfo &channelInfo)

  Returns the string reference name of a ChannelInfo object.

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

  Returns the overal response from the list of responses in a ChannelInfo.

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

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

- double bdsPoleZeroGain (const PoleZero &poleZero, double frequency)

  Calculates the overal gain of the given PoleZero transfer function.

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

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

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

  Convert PoleZero to FAP.

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

  Default filename from a ChannelInfo.

- **BString** bdsFileNameExpand (**BString** fileName, ChannelInfos &channelInfos)

  Default filename from a list of ChannelInfo's.

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

  Uncompress CM8 formatted data.

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

  Uncompress STEIM1 formatted data.

- static **BString** nullString (**BString** s)
- **BUInt32** crc (**BUInt32** crc, void * **data**, int numBytes)
- static void crcInit ()
- static uint64_t crc64 (const void *buffer, const uint32_t len)
- **BString** getHexString (char * **data**, int len)
- int duplicateDump (DataBlock &data1, DataBlock &data2, int channel)
- static **BError** fixedString (double v, int fieldWidth, int numDecimal, **BString** &str)
- void dataCalculateDifference (**BInt32** &prevValue, **BArray**< **BInt32** > & **data**)
- void dataCalculateUnDifference (**BInt32** &prevValue, **BArray**< **BInt32** > & **data**)
- **BInt32** dataChecksum (**BInt32** **checksum**, **BArray**< **BInt32** > & **data**)
- **BError** dataCompressCm6 (int &prevValue1, int &prevValue2, **BArray**< **BInt32** > & **data**, **BString** &d)
- **BError** dataDeCompressCm6 (int &prevValue1, int &prevValue2, **BString** &d, **BArray**< **BInt32** > & **data**)
- static void dataConvert (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BInt32** > &dataOut)
- static **BString** stringFormat (**BTimeStamp** t)
- static **BString** removeCR (**BString** str)
- **BString** fixedWidthValue (double v, int width)

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

- static void bdsDataFileSeedLogWarning (char *str)
- static void bdsDataFileSeedLogError (char *str)
- static hptime_t seedTime (**BTimeStamp** t)
- static **BString** seedTimeString (**BTimeStamp** t)
- static void dataConvert (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BInt32** > &dataOut)
- static void dataConvert (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BFloat32** > &dataOut)
- static void record_handler (char *record, int reclen, void *info)

Variables

- const **BUInt32** apiVersion = 0
- const int NetworkNameLen = 3

  Maximum Network name length.
- const int StationNameLen = 5

  Maximum Station name length.
- const int ChannelTypeLen = 3

  Maximum Channel type name length.
- const int ChannelAuxLen = 2

  Maximum Channel Aux length.
- const int SourceLen = 16

  Maximum Source length.
- const **BString** BdsDataFileVersion = "1.2.0"
- static uint64_t crcVec [256]
- static int crcInitDone
- static char cm6Table [64]
- static **BUInt8** cm6TableRev [128]
- const double Scale = 16777216.0
- DataFormats dataFormats

6.1.1 Enumeration Type Documentation

6.1.1.1 AvailType

`enum Bds::AvailType`

A flag defining the data availability state.

Enumerator

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

6.1.1.2 BdsDataType

`enum Bds::BdsDataType`

BdsDataFileBds: internal file block type field.

Enumerator

BdsDataTypeBlock	

Enumerator

BdsDataTypeInfo	
BdsDataTypeData	
BdsDataTypeInfoExtra	

6.1.1.3 DataFlags

`enum Bds::DataFlags`

Flags when opening data files.

Enumerator

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

6.1.1.4 Errors

`enum Bds::Errors`

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

Enumerator

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

Enumerator

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

6.1.1.5 FileHeaderType

`enum Bds::FileHeaderType`

Enumerator

FileHeaderType_Standard	
FileHeaderType_TapeDigitiser	

6.1.1.6 FileSampleType

`enum Bds::FileSampleType`

Enumerator

FileSampleType_Unknown	
FileSampleType_Float32	
FileSampleType_Float64	
FileSampleType_Int16	
FileSampleType_Int32	

6.1.1.7 Mode

`enum Bds::Mode`

BdsServer mode.

Enumerator

ModeMaster	BdsServer is a master
ModeSlave	BdsServer is a slave

6.1.1.8 Priority

`enum Bds::Priority`

Priority levels.

Enumerator

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

6.1.1.9 SampleFormat

`enum Bds::SampleFormat`

The actual format of a data sample.

Enumerator

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

6.1.1.10 SelectionGroup

`enum Bds::SelectionGroup`

The Selection group when making selections.

Enumerator

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

6.1.2 Function Documentation

6.1.2.1 bdsChannelGetName()

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

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

6.1.2.2 bdsChannelGetTypeAux()

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

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

Parameters

in	*name*	The channels full name
out	*type*	Returns the type component of the channel's name
out	*aux*	Returns the aux component of the channel's name

6.1.2.3 bdsDataChannelInfo()

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

Returns a string representation of a DataChannel object.

6.1.2.4 bdsDataChannelOverallResponse()

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

Returns the overal response from the list of responses in a ChannelInfo.

6.1.2.5 bdsDataChannelRef() [1/2]

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

Returns the string reference name of a DataChannel object.

6.1.2.6 bdsDataChannelRef() [2/2]

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

Returns the string reference name of a ChannelInfo object.

6.1.2.7 bdsDataFileSeedLogError()

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

6.1.2.8 bdsDataFileSeedLogWarning()

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

6.1.2.9 bdsDataInfoFromInfo()

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

Convert info to DataInfo.

Sets up a DataInfo object from a BDictString list of named strings.

6.1.2.10 bdsDataInfoMergeFlatten()

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

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

6.1.2.11 bdsDataInfoSetTimeRange()

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

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

6.1.2.12 bdsDumpChannelInfos()

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

Debug print out a ChannelInfos object.

6.1.2.13 bdsDumpData()

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

Debug print out a DataBlock object.

6.1.2.14 bdsDumpDataInfo()

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

Debug print out a DataInfo object.

6.1.2.15 bdsDumpPoleZeros()

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

Debug print out a PoleZeros object.

6.1.2.16 bdsDumpSelection()

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

Debug print out a Selection object.

6.1.2.17 bdsFileNameExpand() [1/2]

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

Default filename from a ChannelInfo.

6.1.2.18 bdsFileNameExpand() [2/2]

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

Default filename from a list of ChannelInfo's.

6.1.2.19 bdsInfoFromDataInfo()

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

Converts a DataInfo object into a BDictString list of named strings.

6.1.2.20 bdsPoleZeroGain()

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

Calculates the overal gain of the given PoleZero transfer function.

6.1.2.21 bdsPoleZeroGainPhase()

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

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

6.1.2.22 bdsPoleZeroToFap()

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

Convert PoleZero to FAP.

6.1.2.23 bdsSelectionChannelInfo()

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

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

6.1.2.24 bdsStationAlias()

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

Returns the station alias if set else its name.

6.1.2.25 bdsUnCompressCm8()

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

Uncompress CM8 formatted data.

6.1.2.26 bdsUnCompressSteim1()

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

Uncompress STEIM1 formatted data.

6.1.2.27 crc()

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

6.1.2.28 crc64()

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

6.1.2.29 crcInit()

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

6.1.2.30 dataCalculateDifference()

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

6.1.2.31 dataCalculateUnDifference()

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

6.1.2.32 dataChecksum()

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

6.1.2.33 dataCompressCm6()

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

6.1.2.34 dataConvert() [1/3]

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

6.1.2.35 dataConvert() [2/3]

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

6.1.2.36 dataConvert() [3/3]

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

6.1.2.37 dataDeCompressCm6()

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

6.1.2.38 duplicateDump()

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

6.1.2.39 fileNameTime()

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

6.1.2.40 fixedString()

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

6.1.2.41 fixedWidthValue()

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

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

6.1.2.42 getHexString()

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

6.1.2.43 nullString()

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

6.1.2.44 record_handler()

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

6.1.2.45 removeCR()

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

6.1.2.46 seedTime()

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

6.1.2.47 seedTimeString()

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

6.1.2.48 stringFormat()

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

6.1.3 Variable Documentation

6.1.3.1 apiVersion

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

6.1.3.2 BdsDataFileVersion

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

6.1.3.3 ChannelAuxLen

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

Maximum Channel Aux length.

6.1.3.4 ChannelTypeLen

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

Maximum Channel type name length.

6.1.3.5 cm6Table

```
char Bds::cm6Table[64] [static]
```

Initial value:

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

6.1.3.6 cm6TableRev

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

Initial value:

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

6.1.3.7 crcInitDone

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

6.1.3.8 crcVec

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

6.1.3.9 dataFormats

```
DataFormats Bds::dataFormats
```

6.1.3.10 NetworkNameLen

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

Maximum Network name length.

6.1.3.11 Scale

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

6.1.3.12 SourceLen

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

Maximum Source length.

6.1.3.13 StationNameLen

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

Maximum Station name length.

Chapter 7

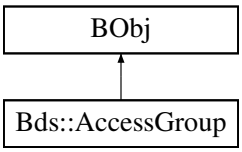
Class Documentation

7.1 Bds::AccessGroup Class Reference

This holds information on data access groups.

`#include <BdsD.h>`

Inheritance diagram for Bds::AccessGroup:



Public Member Functions

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

Public Attributes

- **BUInt32** id

  The unique id.
- **BString** group

  The Group name.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time.
- **BString** network

  The Network Name.
- **BString** station

  The Station name.

7.1.1 Detailed Description

This holds information on data access groups.

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

7.1.2 Constructor & Destructor Documentation

7.1.2.1 AccessGroup()

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

7.1.3 Member Function Documentation

7.1.3.1 getMember()

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

Reimplemented from **BObj**.

7.1.3.2 getMembers()

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

Reimplemented from **BObj**.

7.1.3.3 getType()

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

7.1.3.4 setMember()

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

Reimplemented from **BObj**.

7.1.3.5 setMembers()

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

Reimplemented from **BObj**.

7.1.4 Member Data Documentation

7.1.4.1 endTime

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

The End Time.

7.1.4.2 group

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

The Group name.

7.1.4.3 id

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

The unique id.

7.1.4.4 network

```
BString Bds::AccessGroup::network
```

The Network Name.

7.1.4.5 startTime

```
BTimeStamp Bds::AccessGroup::startTime
```

The Start Time.

7.1.4.6 station

```
BString Bds::AccessGroup::station
```

The Station name.

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

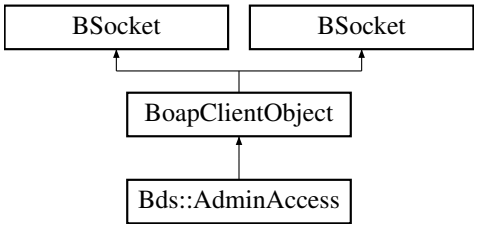
- BdsD.h
- BdsD.cc

7.2 Bds::AdminAccess Class Reference

This is the AdminAccess Access API interface.

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

Inheritance diagram for Bds::AdminAccess:



Public Member Functions

- AdminAccess (**BString** name="")
- **BError** connect (**BString** user, **BString** password)

  Provides user/password information.
- **BError** validateUser (**BString** user, **BString** email)

  Checks the user given name or email.
- **BError** setUser (**BString** user, **BString** email)

  Sets user to given name or email.
- **BError** setUserReal ()

  Sets user back to real user.
- **BError** getVersion (**BString** &version, **BString** &name)

  Gets the software version and server name.
- **BError** userGetList (**BList**< User > &users)

  Get list of Users.
- **BError** userUpdate (**BInt32** append, User user, **BUInt32** &id)

  Update or append a user entry.
- **BError** userDelete (**BUInt32** id)

  Delete a user entry.
- **BError** userGetFromId (**BUInt32** id, User &user)

  Get user info given user ID.
- **BError** userGet (User &user)

  Get user info.
- **BError** userSet (User user)

  Set user info.
- **BError** userGetGroups (**BList**< **BString** > &groups)

  Get list of groups the user belongs to.
- **BError** groupGetList (**BList**< Group > &groups)

  Get list of Groups.
- **BError** groupUpdate (**BInt32** append, Group group, **BUInt32** &id)

  Update or append a group entry.
- **BError** groupDelete (**BUInt32** id)

  Delete a group entry.
- **BError** accessGroupGetList (**BList**< AccessGroup > &accessGroups)

  Get list of AccessGroups.
- **BError** accessGroupUpdate (**BInt32** append, AccessGroup group, **BUInt32** &id)

  Update or append an AccessGroup entry.
- **BError** accessGroupDelete (**BUInt32** id)

  Delete an AccessGroup entry.
- **BError** getSelectionInfo (SelectionGroup group, Selection selectionIn, SelectionInfo &selectionInfo)

  Get information on possible selections. Use in GUI programs to list options available.
- **BError** getSelections (SelectionGroup group, Selection selectionIn, Selection &selectionOut)

  Get selection list.
- **BError** networkGetList (**BList**< Network > &networks)

  Get list of Networks.
- **BError** networkUpdate (**BInt32** append, Network network, **BUInt32** &id)

  Add or update a Network entry.
- **BError** networkDelete (**BUInt32** id)

  Delete a Network entry.
- **BError** stationGetList (Selection sel, **BList**< Station > &stations)

  Get list of Stations.

- **BError** stationUpdate (**BInt32** append, Station station, **BUInt32** &id)

  Add or update a Station entry.
- **BError** stationDelete (**BUInt32** id)

  Delete a Station entry.
- **BError** locationGetList (Selection sel, **BList**< Location > &locations)

  Get list of Station Locations.
- **BError** locationUpdate (**BInt32** append, Location location, **BUInt32** &id)

  Add or update a Station Location entry.
- **BError** locationDelete (**BUInt32** id)

  Delete a Station Location entry.
- **BError** channelGetList (Selection sel, **BList**< Channel > &channels)

  Get list of Channels.
- **BError** channelGet (**BUInt32** id, Channel &channel)

  Get a channel.
- **BError** channelUpdate (**BInt32** append, Channel channel, **BUInt32** &id)

  Add or update a Channel entry.
- **BError** channelDelete (**BUInt32** id)

  Delete a Channel entry.
- **BError** sourceGetList (**BList**< Source > &sources)

  Get list of Sources.
- **BError** sourceUpdate (**BInt32** append, Source source, **BUInt32** &id)

  Add or update a Source entry.
- **BError** sourceDelete (**BUInt32** id)

  Delete a Source entry.
- **BError** sourcePriorityGetList (**BList**< SourcePriority > &sourcePrioritys)

  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** dataFileGetList (Selection sel, **BList**< DataFileInfo > &dataFile)

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

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

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

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

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

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

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

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

  Return the channel MetaData in structured form.
- **BError** dataOpen (DataInfo dataInfo, **BString** mode, **BString** format, **BUInt32** flags, DataHandle &data↩Handle)

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

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

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

  Get information on the data file.
- **BError** dataGetBlock (DataHandle dataHandle, **BUInt32** channel, **BUInt32** segment, **BUInt32** block↩Number, DataBlock & **data**)

  Return a block of data.
- **BError** dataSeekBlock (DataHandle dataHandle, **BUInt32** channel, **BUInt32** segment, **BTimeStamp** time, **BUInt32** &blockNumber)

  Searches for a data block matching the time given.
- **BError** dataSetInfo (DataHandle dataHandle, DataInfo dataInfo)

  Set the info when writing to a file.
- **BError** dataPutBlock (DataHandle dataHandle, DataBlock **data**)

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

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

Read the raw data from the file.

- **BError** dataFormattedGetLength (DataHandle dataHandle, **BUInt64** & **length**)

  Read the raw data from the file.
- **BError** changeGroupStart (ChangeGroup changeGroup)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

  Backup the database.

- **BError** databaseRestore (**BString** ref, **BString** type)

  Restore the database.
- **BError** sqlQuery (**BString** query, **BList**< **BDict**< **BString** > > &result)

  A low level SQL access function.

Additional Inherited Members

7.2.1 Detailed Description

This is the AdminAccess Access API interface.

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

7.2.2 Constructor & Destructor Documentation

7.2.2.1 AdminAccess()

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

7.2.3 Member Function Documentation

7.2.3.1 accessGroupDelete()

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

Delete an AccessGroup entry.

7.2.3.2 accessGroupGetList()

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

Get list of AccessGroups.

7.2.3.3 accessGroupUpdate()

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

Update or append an AccessGroup entry.

7.2.3.4 calibrationDelete()

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

Delete a Calibration entry.

7.2.3.5 calibrationGetList()

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

Get list of Calibrations.

7.2.3.6 calibrationUpdate()

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

Add or update a Calibration entry.

7.2.3.7 changeDelete()

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

Delete a Change.

7.2.3.8 changeGetList()

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

Get a list of Changes.

7.2.3.9 changeGetListNumber()

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

Get the number of changes in a ChangeGroup.

7.2.3.10 changeGroupDelete()

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

Delete a ChangeGroup.

7.2.3.11 changeGroupEnd()

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

End a ChangeGroup.

7.2.3.12 changeGroupGetList()

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

Return a list of ChangeGroups.

7.2.3.13 changeGroupStart()

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

Start a new ChangeGroup when making a set of changes to the BDS's database.

7.2.3.14 channelDelete()

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

Delete a Channel entry.

7.2.3.15 channelGet()

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

Get a channel.

7.2.3.16 channelGetList()

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

Get list of Channels.

7.2.3.17 channelInstrumentDelete()

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

Delete an Instrument entry.

7.2.3.18 channelInstrumentGetList()

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

Get list of Channel Instruments.

7.2.3.19 channelInstrumentUpdate()

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

Add or update a Instrument entry.

7.2.3.20 channelUpdate()

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

Add or update a Channel entry.

7.2.3.21 clean()

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

Cleans the system logs and Changes information.

7.2.3.22 connect()

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

Provides user/password information.

7.2.3.23 dataAvailability()

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

Return availability for data matching the given selection parameters.

7.2.3.24 databaseBackup()

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

Backup the database.

7.2.3.25 databaseRestore()

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

Restore the database.

7.2.3.26 dataChannelDelete()

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

Delete a DataChannel entry.

7.2.3.27 dataChannelGetList()

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

Get list of DataChannels.

7.2.3.28 dataChannelUpdate()

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

Add or update a DataChannel entry.

7.2.3.29 dataClose()

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

Close a file.

7.2.3.30 dataFileDelete()

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

Delete a DataFile entry.

7.2.3.31 dataFileGetList()

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

Get list of DataFiles.

7.2.3.32 dataFileUpdate()

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

Add or update a DataFile entry.

7.2.3.33 dataFormatGetList()

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

Get list of supported data formats.

7.2.3.34 dataFormattedGetLength()

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

Read the raw data from the file.

7.2.3.35 dataFormattedRead()

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

Read the raw data from the file.

7.2.3.36 dataGetBlock()

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

Return a block of data.

7.2.3.37 dataGetChannelInfo()

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

Return the channel MetaData in structured form.

7.2.3.38 dataGetInfo()

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

Get information on the data file.

7.2.3.39 dataGetNotes()

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

Get notes on the data file.

7.2.3.40 dataGetWarnings()

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

Get information on the data file.

7.2.3.41 dataOpen()

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

Open a data file.

Parameters

in	*dataInfo*	The sensor data to open selection
in	*mode*	The open format. The mode should be set to "w" for writing data, "a" when appending data and "r" for reading data.
in	*format*	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	*flags*	A bitset of flags as defined by Bds::DataFlags.
out	*dataHandle*	The handle for the open data set

7.2.3.42 dataPutBlock()

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

Send a block of data.

7.2.3.43 dataSearch()

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

Search for data matching the given selection parameters.

7.2.3.44 dataSeekBlock()

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

Searches for a data block matching the time given.

7.2.3.45 dataSetInfo()

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

Set the info when writing to a file.

7.2.3.46 digitiserDelete()

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

Delete a Digitiser entry.

7.2.3.47 digitiserGet()

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

Get a Digitiser object given its ID.

7.2.3.48 digitiserGetList()

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

Get list of Digitisers.

7.2.3.49 digitiserUpdate()

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

Add or update a Digitiser entry.

7.2.3.50 getSelectionInfo()

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

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

7.2.3.51 getSelections()

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

Get selection list.

7.2.3.52 getVersion()

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

Gets the software version and server name.

7.2.3.53 groupDelete()

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

Delete a group entry.

7.2.3.54 groupGetList()

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

Get list of Groups.

7.2.3.55 groupUpdate()

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

Update or append a group entry.

7.2.3.56 locationDelete()

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

Delete a Station Location entry.

7.2.3.57 locationGetList()

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

Get list of Station Locations.

7.2.3.58 locationUpdate()

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

Add or update a Station Location entry.

7.2.3.59 logAppend()

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

Append a log item.

7.2.3.60 logDelete()

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

Delete a Log item.

7.2.3.61 logGetList()

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

Get list of log entries.

7.2.3.62 logUpdate()

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

Add or Update a Log item.

7.2.3.63 modeSet()

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

Changes the system mode from Master to slave.

7.2.3.64 modeSnapshotPause()

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

Enables/disables backup synchronisation pause.

7.2.3.65 networkDelete()

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

Delete a Network entry.

7.2.3.66 networkGetList()

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

Get list of Networks.

7.2.3.67 networkUpdate()

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

Add or update a Network entry.

7.2.3.68 noteDelete()

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

Delete a Note.

7.2.3.69 noteGetList()

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

Get a list of Notes.

7.2.3.70 noteReadDocument()

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

Read a document associated with a Note.

7.2.3.71 noteUpdate()

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

Add or update a Note.

7.2.3.72 noteWriteDocument()

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

Given a Note write a document associated with it.

7.2.3.73 responseDelete()

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

Delete a Response entry.

7.2.3.74 responseGetList()

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

Get list of Responses.

7.2.3.75 responseUpdate()

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

Add or update a Response entry.

7.2.3.76 sensorDelete()

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

Delete a Sensor entry.

7.2.3.77 sensorGet()

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

Get a Sensor object given its ID.

7.2.3.78 sensorGetList()

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

Get list of Sensors.

7.2.3.79 sensorUpdate()

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

Add or update a Sensor entry.

7.2.3.80 setUser()

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

Sets user to given name or email.

7.2.3.81 setUserReal()

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

Sets user back to real user.

7.2.3.82 sourceDelete()

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

Delete a Source entry.

7.2.3.83 sourceGetList()

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

Get list of Sources.

7.2.3.84 sourcePriorityDelete()

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

7.2.3.85 sourcePriorityGetList()

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

Get list of SourcePriorities.

7.2.3.86 sourcePriorityUpdate()

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

7.2.3.87 sourceUpdate()

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

Add or update a Source entry.

7.2.3.88 sqlQuery()

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

A low level SQL access function.

7.2.3.89 stationDelete()

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

Delete a Station entry.

7.2.3.90 stationGetList()

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

Get list of Stations.

7.2.3.91 stationUpdate()

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

Add or update a Station entry.

7.2.3.92 statisticsGet()

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

Get a list of system statistics.

7.2.3.93 transactionEnd()

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

Ends a set of transactions.

7.2.3.94 transactionStart()

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

Starts a set of transactions.

7.2.3.95 userDelete()

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

Delete a user entry.

7.2.3.96 userGet()

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

Get user info.

7.2.3.97 userGetFromId()

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

Get user info given user ID.

7.2.3.98 userGetGroups()

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

Get list of groups the user belongs to.

7.2.3.99 userGetList()

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

Get list of Users.

7.2.3.100 userSet()

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

Set user info.

7.2.3.101 userUpdate()

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

Update or append a user entry.

7.2.3.102 validateUser()

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

Checks the user given name or email.

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

- BdsC.h
- BdsC.cc
- BdsLib.dox

7.3 Bds::ArrayChannel Class Reference

This class defines an arrays channel.

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

Public Member Functions

- ArrayChannel (**BString** station= **BString**(), **BString** channel= **BString**())

Public Attributes

- **BString** station

  The Stations name.
- **BString** channel

  The Channels name.

7.3.1 Detailed Description

This class defines an arrays channel.

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

7.3.2 Constructor & Destructor Documentation

7.3.2.1 ArrayChannel()

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

7.3.3 Member Data Documentation

7.3.3.1 channel

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

The Channels name.

7.3.3.2 station

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

The Stations name.

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

- BdsD.h
- BdsD.cc

7.4 Bds::BdsDataBlock Struct Reference

BdsDataFileBds: internal fixed size BDS Data Block.

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

Public Attributes

- BdsDataBlockHeader header
- char data [4]

  The packet data.

7.4.1 Detailed Description

BdsDataFileBds: internal fixed size BDS Data Block.

7.4.2 Member Data Documentation

7.4.2.1 data

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

The packet data.

7.4.2.2 header

```
BdsDataBlockHeader Bds::BdsDataBlock::header
```

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

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

7.5 Bds::BdsDataBlockHeader Struct Reference

BdsDataFileBds: internal fixed size BDS Data Block header.

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

Public Attributes

- **BUInt32** type

  Blocks type, contains magic number for synchronisation.
- **BUInt32** length

  Packet length.
- **BUInt32** packetOffset

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

7.5.1 Detailed Description

BdsDataFileBds: internal fixed size BDS Data Block header.

7.5.2 Member Data Documentation

7.5.2.1 length

```
BUInt32 Bds::BdsDataBlockHeader::length
```

Packet length.

7.5.2.2 packetOffset

```
BUInt32 Bds::BdsDataBlockHeader::packetOffset
```

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

7.5.2.3 type

```
BUInt32 Bds::BdsDataBlockHeader::type
```

Blocks type, contains magic number for synchronisation.

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

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

7.6 Bds::BdsDataBlockPos Class Reference

BdsDataFileBds: internal file storage data block position.

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

Public Member Functions

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

Public Attributes

- **BTimeStamp** startTime
- **BTimeStamp** endTime
- **BUInt32** channel
- **BUInt32** numChannels
- **BUInt32** segment
- **BUInt64** position
- **BUInt64** numSamples

7.6.1 Detailed Description

BdsDataFileBds: internal file storage data block position.

7.6.2 Constructor & Destructor Documentation

7.6.2.1 BdsDataBlockPos()

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

7.6.3 Member Function Documentation

7.6.3.1 operator<()

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

7.6.4 Member Data Documentation

7.6.4.1 channel

```
BUInt32 Bds::BdsDataBlockPos::channel
```

7.6.4.2 endTime

```
BTimeStamp Bds::BdsDataBlockPos::endTime
```

7.6.4.3 numChannels

```
BUInt32 Bds::BdsDataBlockPos::numChannels
```

7.6.4.4 numSamples

```
BUInt64 Bds::BdsDataBlockPos::numSamples
```

7.6.4.5 position

```
BUInt64 Bds::BdsDataBlockPos::position
```

7.6.4.6 segment

```
BUInt32 Bds::BdsDataBlockPos::segment
```

7.6.4.7 startTime

```
BTimeStamp Bds::BdsDataBlockPos::startTime
```

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

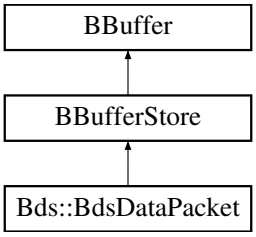
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h

7.7 Bds::BdsDataPacket Class Reference

BdsDataFileBds: internal file storage packet.

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

Inheritance diagram for Bds::BdsDataPacket:



Public Member Functions

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

Additional Inherited Members

7.7.1 Detailed Description

BdsDataFileBds: internal file storage packet.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 BdsDataPacket()

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

7.7.2.2 ~BdsDataPacket()

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

7.7.3 Member Function Documentation

7.7.3.1 clear()

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

7.7.3.2 dump()

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

7.7.3.3 getHeader()

```
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 ()
- int 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:

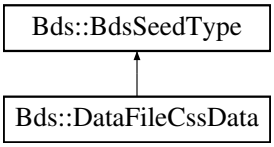
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h

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:

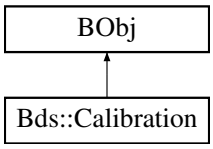
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.h
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp

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= **B**↩ **TimeStamp**(), **BString** network= **BString**(), **BString** station= **BString**(), **BString** channel= **BString**(), **BString** source= **BString**(), **BString** name= **BString**(), **BFloat64** samplingFrequency=0, **BFloat64** calibrationFrequency=0, **BFloat64** calibrationFactor=0, **BString** calibrationUnits= **BString**(), **BFloat64** depth=0, **BFloat64** horizontalAngle=0, **BFloat64** verticalAngle=0)
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  The ID.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time.
- **BString** network

  The Network/Organisation Name.
- **BString** station

  The station.
- **BString** channel

  The channel.
- **BString** source

  The source.
- **BString** name

  The Calibrations name, "Main", "Measured".
- **BFloat64** samplingFrequency

  The sample rate used.
- **BFloat64** calibrationFrequency

  The frequency that the CalibrationFactor value is valid for.
- **BFloat64** calibrationFactor

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

  The measurment units.
- **BFloat64** depth

  The depth of the sensor below ground level in meters.
- **BFloat64** horizontalAngle

  The Sensors channel placement horizontal angle in degrees clockwise from north.
- **BFloat64** verticalAngle

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

7.12.1 Detailed Description

This class defines a calibration setting.

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

7.12.2 Constructor & Destructor Documentation

7.12.2.1 Calibration()

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

7.12.3 Member Function Documentation

7.12.3.1 getMember()

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

Reimplemented from **BObj**.

7.12.3.2 getMembers()

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

Reimplemented from **BObj**.

7.12.3.3 getType()

```
BString Bds::Calibration::getType ( )
```

7.12.3.4 setMember()

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

Reimplemented from **BObj**.

7.12.3.5 setMembers()

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

Reimplemented from **BObj**.

7.12.4 Member Data Documentation

7.12.4.1 calibrationFactor

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

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

7.12.4.2 calibrationFrequency

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

The frequency that the CalibrationFactor value is valid for.

7.12.4.3 calibrationUnits

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

The measurment units.

7.12.4.4 channel

```
BString Bds::Calibration::channel
```

The channel.

7.12.4.5 depth

```
BFloat64 Bds::Calibration::depth
```

The depth of the sensor below ground level in meters.

7.12.4.6 endTime

```
BTimeStamp Bds::Calibration::endTime
```

The End Time.

7.12.4.7 horizontalAngle

```
BFloat64 Bds::Calibration::horizontalAngle
```

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

7.12.4.8 id

```
BUInt32 Bds::Calibration::id
```

The ID.

7.12.4.9 name

```
BString Bds::Calibration::name
```

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

7.12.4.10 network

```
BString Bds::Calibration::network
```

The Network/Organisation Name.

7.12.4.11 samplingFrequency

```
BFloat64 Bds::Calibration::samplingFrequency
```

The sample rate used.

7.12.4.12 source

```
BString Bds::Calibration::source
```

The source.

7.12.4.13 startTime

```
BTimeStamp Bds::Calibration::startTime
```

The Start Time.

7.12.4.14 station

```
BString Bds::Calibration::station
```

The station.

7.12.4.15 verticalAngle

```
BFloat64 Bds::Calibration::verticalAngle
```

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

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

- BdsD.h
- BdsD.cc

7.13 Bds::CdChannel_1v0 Struct Reference

BdsDataFile: Internal CD1.0 channel information.

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

Public Attributes

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

7.13.1 Detailed Description

BdsDataFile: Internal CD1.0 channel information.

7.13.2 Member Data Documentation

7.13.2.1 auth

```
BUInt8 Bds::CdChannel_1v0::auth
```

7.13.2.2 calibrationFactor

```
BFloat32 Bds::CdChannel_1v0::calibrationFactor
```

7.13.2.3 calibrationPeriod

```
BFloat32 Bds::CdChannel_1v0::calibrationPeriod
```

7.13.2.4 channel

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

7.13.2.5 compress

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

7.13.2.6 name

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

7.13.2.7 spare0

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

7.13.2.8 spare1

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

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

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

7.14 Bds::CdDataChannel Class Reference

BdsDataFile: Internal CD channel information.

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

Public Attributes

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

7.14.1 Detailed Description

BdsDataFile: Internal CD channel information.

7.14.2 Member Data Documentation

7.14.2.1 channel

```
BString Bds::CdDataChannel::channel
```

7.14.2.2 data

```
BUInt8* Bds::CdDataChannel::data
```

7.14.2.3 dataSize

```
BUInt32 Bds::CdDataChannel::dataSize
```

7.14.2.4 mode

```
char Bds::CdDataChannel::mode[24]
```

7.14.2.5 numSamples

```
BUInt32 Bds::CdDataChannel::numSamples
```

7.14.2.6 period

```
BUInt32 Bds::CdDataChannel::period
```

7.14.2.7 startTime

```
BTimeStamp Bds::CdDataChannel::startTime
```

7.14.2.8 station

```
BString Bds::CdDataChannel::station
```

7.14.2.9 status

```
char Bds::CdDataChannel::status[32]
```

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

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

7.15 Bds::CdDataFormatFrame_1v0 Struct Reference

BdsDataFile: Internal CD1.0 frame information.

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

Public Attributes

- **BUInt32** frameType
- **BUInt32** frameLength
- **BUInt32** maxFrameLength
- **BUInt32** numChannels
- **BUInt32** period
- CdChannel_1v0 channels [100]

7.15.1 Detailed Description

BdsDataFile: Internal CD1.0 frame information.

7.15.2 Member Data Documentation

7.15.2.1 channels

```
CdChannel_1v0 Bds::CdDataFormatFrame_1v0::channels[100]
```

7.15.2.2 frameLength

```
BUInt32 Bds::CdDataFormatFrame_1v0::frameLength
```

7.15.2.3 frameType

```
BUInt32 Bds::CdDataFormatFrame_1v0::frameType
```

7.15.2.4 maxFrameLength

```
BUInt32 Bds::CdDataFormatFrame_1v0::maxFrameLength
```

7.15.2.5 numChannels

```
BUInt32 Bds::CdDataFormatFrame_1v0::numChannels
```

7.15.2.6 period

```
BUInt32 Bds::CdDataFormatFrame_1v0::period
```

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

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

7.16 Bds::CdFlag Class Reference

BdsDataFile: Internal CD flag.

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

Public Member Functions

- CdFlag ()

Public Attributes

- int dead
- int zeroed

7.16.1 Detailed Description

BdsDataFile: Internal CD flag.

7.16.2 Constructor & Destructor Documentation

7.16.2.1 CdFlag()

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

7.16.3 Member Data Documentation

7.16.3.1 dead

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

7.16.3.2 zeroed

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

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

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

7.17 Bds::CdPacketData Class Reference

BdsDataFile: Internal CD data packet.

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

Public Attributes

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

7.17.1 Detailed Description

BdsDataFile: Internal CD data packet.

7.17.2 Member Data Documentation

7.17.2.1 auth

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

7.17.2.2 authKey

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

7.17.2.3 authSize

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

7.17.2.4 channels

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

7.17.2.5 crc

```
BUInt64 Bds::CdPacketData::crc
```

7.17.2.6 creator

```
char Bds::CdPacketData::creator[8]
```

7.17.2.7 destination

```
char Bds::CdPacketData::destination[8]
```

7.17.2.8 frameType

```
BUInt32 Bds::CdPacketData::frameType
```

7.17.2.9 numChannels

```
BUInt32 Bds::CdPacketData::numChannels
```

7.17.2.10 period

```
BUInt32 Bds::CdPacketData::period
```

7.17.2.11 sequenceNum

```
BUInt64 Bds::CdPacketData::sequenceNum
```

7.17.2.12 series

```
BUInt32 Bds::CdPacketData::series
```

7.17.2.13 startTime

```
BTimeStamp Bds::CdPacketData::startTime
```

7.17.2.14 trailerOffset

```
BUInt32 Bds::CdPacketData::trailerOffset
```

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

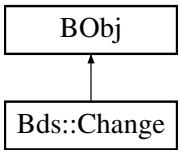
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h

7.18 Bds::Change Class Reference

This holds information on a medatdata or sensor data change.

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

Inheritance diagram for Bds::Change:



Public Member Functions

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

Public Attributes

- **BUInt32** id

  The unique id.
- **BUInt32** changeGroupId

  The Change group ID.
- **BTimeStamp** time

  The Time the change was made.
- **BString** type

  The change type.
- **BString** table

  The database table affected.
- **BUInt32** rowId

  The database row affected.

7.18.1 Detailed Description

This holds information on a medatdata 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:

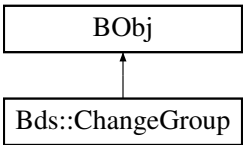
- BdsD.h
- BdsD.cc

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:

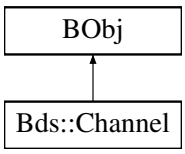
- BdsD.h
- BdsD.cc

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**(), **B↩String** 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 auxilary 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 auxilary identifier (component of station field)

7.20.4.3 channelType

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

The channels type (component of station field)

7.20.4.4 dataType

```
BString Bds::Channel::dataType
```

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

7.20.4.5 description

```
BString Bds::Channel::description
```

The channels description.

7.20.4.6 endTime

```
BTimeStamp Bds::Channel::endTime
```

The End Time the channel was available.

7.20.4.7 id

```
BUInt32 Bds::Channel::id
```

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

7.20.4.8 network

```
BString Bds::Channel::network
```

The Network Name.

7.20.4.9 startTime

```
BTimeStamp Bds::Channel::startTime
```

The Start Time.

7.20.4.10 station

```
BString Bds::Channel::station
```

The Stations name.

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

- BdsD.h
- BdsD.cc

7.21 Bds::ChannelInfo Class Reference

This class provides information on a channel.

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

Public Member Functions

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

Public Attributes

- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time.
- Station station

  The Station info.
- Location location

  The Station location.
- Channel channel

  The Channel data.
- **BString** source

  The data source.
- **BString** dataType

  The DataType (seismic, seismicUnknown, data, log, unknown, empty)
- Digitiser digitiser

  The Digitiser in use.
- Sensor sensor

  The Sensor in use.
- Calibration calibration

  The Calibration info.
- **BList**< Response > responses

  The list of frequency responses.

7.21.1 Detailed Description

This class provides information on a channel.

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

7.21.2 Constructor & Destructor Documentation

7.21.2.1 ChannelInfo()

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

7.21.3 Member Data Documentation

7.21.3.1 calibration

```
Calibration Bds::ChannelInfo::calibration
```

The Calibration info.

7.21.3.2 channel

```
Channel Bds::ChannelInfo::channel
```

The Channel data.

7.21.3.3 dataType

```
BString Bds::ChannelInfo::dataType
```

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

7.21.3.4 digitiser

```
Digitiser Bds::ChannelInfo::digitiser
```

The Digitiser in use.

7.21.3.5 endTime

```
BTimeStamp Bds::ChannelInfo::endTime
```

The End Time.

7.21.3.6 location

```
Location Bds::ChannelInfo::location
```

The Station location.

7.21.3.7 responses

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

The list of frequency responses.

7.21.3.8 sensor

```
Sensor Bds::ChannelInfo::sensor
```

The Sensor in use.

7.21.3.9 source

```
BString Bds::ChannelInfo::source
```

The data source.

7.21.3.10 startTime

```
BTimeStamp Bds::ChannelInfo::startTime
```

The Start Time.

7.21.3.11 station

```
Station Bds::ChannelInfo::station
```

The Station info.

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

- BdsD.h
- BdsD.cc

7.22 Bds::ChannelInfos Class Reference

This class provides metadata information on a set of channels.

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

Public Member Functions

- ChannelInfos (**BArray**< **BArray**< ChannelInfo > > channels= **BArray**< **BArray**< ChannelInfo > >())

Public Attributes

- **BArray**< **BArray**< ChannelInfo > > channels

7.22.1 Detailed Description

This class provides metadata information on a set of channels.

This provides all of the metadata information for a set of channels over a given time period. For each channel there can be one or more ChannelInfo objects each defining a set of 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 (
            BArray< BArray< ChannelInfo > > channels = BArray< BArray<ChannelInfo > >()
)
```

7.22.3 Member Data Documentation

7.22.3.1 channels

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

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

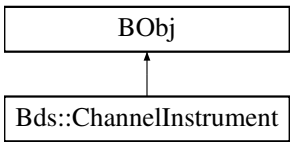
- BdsD.h
- BdsD.cc

7.23 Bds::ChannelInstrument Class Reference

This class defines a Channel's instrument.

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

Inheritance diagram for Bds::ChannelInstrument:



Public Member Functions

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

Public Attributes

- **BUInt32** id

  Unique ID when stored in a database or for other uses.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time the channel was available.
- **BUInt32** channelId

  The channels Id.
- **BString** source

  The source.
- **BUInt32** digitiserId

  The Digitiser in use.
- **BUInt32** sensorId

  The sensor in use.

7.23.1 Detailed Description

This class defines a Channel's instrument.

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

7.23.2 Constructor & Destructor Documentation

7.23.2.1 ChannelInstrument()

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

7.23.3 Member Function Documentation

7.23.3.1 getMember()

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

Reimplemented from **BObj**.

7.23.3.2 getMembers()

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

Reimplemented from **BObj**.

7.23.3.3 getType()

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

7.23.3.4 setMember()

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

Reimplemented from **BObj**.

7.23.3.5 setMembers()

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

Reimplemented from **BObj**.

7.23.4 Member Data Documentation

7.23.4.1 channelId

```
BUInt32 Bds::ChannelInstrument::channelId
```

The channels Id.

7.23.4.2 digitiserId

```
BUInt32 Bds::ChannelInstrument::digitiserId
```

The Digitiser in use.

7.23.4.3 endTime

```
BTimeStamp Bds::ChannelInstrument::endTime
```

The End Time the channel was available.

7.23.4.4 id

```
BUInt32 Bds::ChannelInstrument::id
```

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

7.23.4.5 sensorId

```
BUInt32 Bds::ChannelInstrument::sensorId
```

The sensor in use.

7.23.4.6 source

```
BString Bds::ChannelInstrument::source
```

The source.

7.23.4.7 startTime

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

The Start Time.

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

- BdsD.h
- BdsD.cc

7.24 Bds::ChannelName Class Reference

This class defines a full channel name.

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

Public Member Functions

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

Public Attributes

- **BString** network

  The Channels network.
- **BString** station

  The Channels station.
- **BString** channel

  The Channels name.
- **BString** source

  The Channels source.

7.24.1 Detailed Description

This class defines a full channel name.

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

7.24.2 Constructor & Destructor Documentation

7.24.2.1 ChannelName()

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

7.24.3 Member Data Documentation

7.24.3.1 channel

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

The Channels name.

7.24.3.2 network

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

The Channels network.

7.24.3.3 source

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

The Channels source.

7.24.3.4 station

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

The Channels station.

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

- BdsD.h
- BdsD.cc

7.25 Bds::CleanOptions Class Reference

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

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

Public Member Functions

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

Public Attributes

- **BInt32** logs

  Clean the Logs.
- **BInt32** changes

  Clean the changes.
- **BInt32** deletedFiles

  Clean deleted data files.

7.25.1 Detailed Description

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

7.25.2 Constructor & Destructor Documentation

7.25.2.1 CleanOptions()

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

7.25.3 Member Data Documentation

7.25.3.1 changes

```
BInt32 Bds::CleanOptions::changes
```

Clean the changes.

7.25.3.2 deletedFiles

```
BInt32 Bds::CleanOptions::deletedFiles
```

Clean deleted data files.

7.25.3.3 logs

```
BInt32 Bds::CleanOptions::logs
```

Clean the Logs.

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

- BdsD.h
- BdsD.cc

7.26 Bds::CompressSteim1 Class Reference

Steim1 un-compress class.

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

Public Member Functions

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

7.26.1 Detailed Description

Steim1 un-compress class.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 CompressSteim1()

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

7.26.3 Member Function Documentation

7.26.3.1 clear()

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

7.26.3.2 setByteOrder()

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

7.26.3.3 unCompress()

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

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

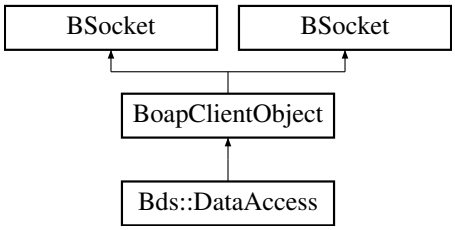
- /src/blacknest/bds/bds/bdsDataLib/BdsCompress.h
- /src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp

7.27 Bds::DataAccess Class Reference

This is the Data Access API interface.

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

Inheritance diagram for Bds::DataAccess:



Public Member Functions

- DataAccess (**BString** name="")
- **BError** connect (**BString** user, **BString** password)

  Provides user/password information for secure connection.
- **BError** validateUser (**BString** user, **BString** email)

  Checks the user given name or email.
- **BError** setUser (**BString** user, **BString** email)

  Sets user to given name or email.
- **BError** setUserReal ()

  Sets user back to real user.
- **BError** getVersion (**BString** &version, **BString** &name)

  Gets the software version and server name.
- **BError** userGetFromId (**BUInt32** id, User &user)

  Get user info given user ID.
- **BError** userGet (User &user)

  Get user info.
- **BError** userSet (User user)

  Set user info.
- **BError** userGetGroups (**BList**< **BString** > &groups)

  Get list of groups the user belongs to.
- **BError** groupGetList (**BList**< Group > &groups)

  Get list of Groups.
- **BError** networkGetList (**BList**< Network > &networks)

  Get list of Networks.
- **BError** stationGetList (Selection sel, **BList**< Station > &stations)

  Get list of Stations.
- **BError** channelGetList (Selection sel, **BList**< Channel > &channels)

  Get list of Channels.
- **BError** sourceGetList (**BList**< Source > &sources)

  Get list of Sources.
- **BError** sourcePriorityGetList (**BList**< SourcePriority > &sourcePrioritys)

  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 Responces.

- **BError** locationGetList (Selection sel, **BList**< Location > &locations)

  Get list of Station Locations.
- **BError** getSelectionInfo (SelectionGroup group, Selection selectionIn, SelectionInfo &selectionInfo)

  Get information on possible selections. Use in GUI programs to list options available.
- **BError** getSelections (SelectionGroup group, Selection selectionIn, Selection &selectionOut)

  Get selection list.
- **BError** dataAvailability (Selection selection, **BUInt32** num, **BArray**< DataAvailChan > &dataAvailChans)

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

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

  Return the channel MetaData in structured form.
- **BError** dataOpen (DataInfo dataInfo, **BString** mode, **BString** format, **BUInt32** flags, DataHandle &data↩Handle)

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

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

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

  Get information on the data file.
- **BError** dataSeekBlock (DataHandle dataHandle, **BUInt32** channel, **BUInt32** segment, **BTimeStamp** time, **BUInt32** &blockNumber)

  Searches for a data block matching the time given.
- **BError** dataGetBlock (DataHandle dataHandle, **BUInt32** channel, **BUInt32** segment, **BUInt32** block↩Number, DataBlock & **data**)

  Return a block of data.
- **BError** 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** noteGetList (Selection sel, **BList**< Note > &notes)

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

  Add or update a Note.
- **BError** noteWriteDocument (**BUInt32** id, **BString** format, **BArray**< **BUInt8** > **data**)

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

  Read a document associated with a Note.
- **BError** logUpdate (**BInt32** append, Log log, **BUInt32** &id)

  Add or update a Log item.
- **BError** logAppend (**BString** type, **BUInt32** priority, **BString** subSystem, **BString** title, **BString** description)

  Append a log item.
- **BError** modeSet (Mode mode, Mode &previousMode)

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

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

  Cleans the system logs and Changes information.

- **BError** databaseBackup (**BString** &ref)

  Backup the database.
- **BError** statisticsGet (**BDict**< **BString** > &info)

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

  Get list of data formats.

Additional Inherited Members

7.27.1 Detailed Description

This is the Data Access API interface.

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

7.27.2 Constructor & Destructor Documentation

7.27.2.1 DataAccess()

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

7.27.3 Member Function Documentation

7.27.3.1 calibrationGetList()

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

Get list of Calibrations.

7.27.3.2 channelGetList()

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

Get list of Channels.

7.27.3.3 channelInstrumentGetList()

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

Get list of Instruments.

7.27.3.4 clean()

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

Cleans the system logs and Changes information.

7.27.3.5 connect()

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

Provides user/password information for secure connection.

7.27.3.6 dataAvailability()

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

Return availability for data matching the given selection parameters.

7.27.3.7 databaseBackup()

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

Backup the database.

7.27.3.8 dataChannelGetList()

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

Get list of DataChannels.

7.27.3.9 dataClose()

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

Close a file.

7.27.3.10 dataFileGetList()

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

Get list of DataFiles.

7.27.3.11 dataFormatGetList()

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

Get list of data formats.

7.27.3.12 dataFormattedGetLength()

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

Read the raw data from the file.

7.27.3.13 dataFormattedRead()

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

Read the raw data from the file.

7.27.3.14 dataGetBlock()

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

Return a block of data.

7.27.3.15 dataGetChannelInfo()

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

Return the channel MetaData in structured form.

7.27.3.16 dataGetInfo()

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

Get information on the data file.

7.27.3.17 dataGetNotes()

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

Get notes on the data file.

7.27.3.18 dataGetWarnings()

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

Get information on the data file.

7.27.3.19 dataOpen()

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

Open a data file.

Parameters

in	*dataInfo*	The sensor data to open selection
in	*mode*	The open format. The mode should be set to "w" for writing data, "a" when appending data and "r" for reading data.
in	*format*	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	*flags*	A bitset of flags as defined by Bds::DataFlags.
out	*dataHandle*	The handle for the open data set

7.27.3.20 dataSearch()

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

Search for data matching the given selection parameters.

7.27.3.21 dataSeekBlock()

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

Searches for a data block matching the time given.

7.27.3.22 digitiserGet()

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

Get a Digitiser object given its ID.

7.27.3.23 digitiserGetList()

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

Get list of Digitisers.

7.27.3.24 getSelectionInfo()

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

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

7.27.3.25 getSelections()

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

Get selection list.

7.27.3.26 getVersion()

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

Gets the software version and server name.

7.27.3.27 groupGetList()

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

Get list of Groups.

7.27.3.28 locationGetList()

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

Get list of Station Locations.

7.27.3.29 logAppend()

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

Append a log item.

7.27.3.30 logUpdate()

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

Add or update a Log item.

7.27.3.31 modeSet()

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

Changes the system mode from Master to slave.

7.27.3.32 modeSnapshotPause()

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

Enables/disables backup synchronisation pause.

7.27.3.33 networkGetList()

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

Get list of Networks.

7.27.3.34 noteGetList()

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

Return a list of Notes.

7.27.3.35 noteReadDocument()

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

Read a document associated with a Note.

7.27.3.36 noteUpdate()

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

Add or update a Note.

7.27.3.37 noteWriteDocument()

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

Given a Note write a document associated with it.

7.27.3.38 responseGetList()

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

Get list of Responces.

7.27.3.39 sensorGet()

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

Get a Sensor object given its ID.

7.27.3.40 sensorGetList()

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

Get list of Sensors.

7.27.3.41 setUser()

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

Sets user to given name or email.

7.27.3.42 setUserReal()

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

Sets user back to real user.

7.27.3.43 sourceGetList()

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

Get list of Sources.

7.27.3.44 sourcePriorityGetList()

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

Get list of SourcePriorities.

7.27.3.45 stationGetList()

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

Get list of Stations.

7.27.3.46 statisticsGet()

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

Get a list of system statistics.

7.27.3.47 userGet()

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

Get user info.

7.27.3.48 userGetFromId()

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

Get user info given user ID.

7.27.3.49 userGetGroups()

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

Get list of groups the user belongs to.

7.27.3.50 userSet()

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

Set user info.

7.27.3.51 validateUser()

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

Checks the user given name or email.

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

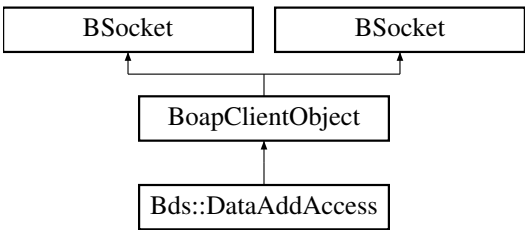
- BdsC.h
- BdsC.cc
- BdsLib.dox

7.28 Bds::DataAddAccess Class Reference

This is the DataAdd Access API interface.

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

Inheritance diagram for Bds::DataAddAccess:



Public Member Functions

- DataAddAccess (**BString** name="")
- **BError** connect (**BString** user, **BString** password)

  Provides user/password information.
- **BError** validateUser (**BString** user, **BString** email)

  Checks the user given name or email.
- **BError** setUser (**BString** user, **BString** email)

  Sets user to given name or email.
- **BError** setUserReal ()

  Sets user back to real user.
- **BError** getVersion (**BString** &version, **BString** &name)

  Gets the software version and server name.
- **BError** userGetFromId (**BUInt32** id, User &user)

  Get user info given user ID.
- **BError** userGet (User &user)

  Get user info.
- **BError** userSet (User user)

  Set user info.
- **BError** userGetGroups (**BList**< **BString** > &groups)

  Get list of groups the user belongs to.
- **BError** groupGetList (**BList**< Group > &groups)

  Get list of Groups.
- **BError** networkGetList (**BList**< Network > &networks)

  Get list of Networks.
- **BError** stationGetList (Selection sel, **BList**< Station > &stations)

  Get list of Stations.
- **BError** channelGetList (Selection sel, **BList**< Channel > &channels)

  Get list of Channels.
- **BError** sourceGetList (**BList**< Source > &sources)

  Get list of Sources.
- **BError** sourcePriorityGetList (**BList**< SourcePriority > &sourcePrioritys)

  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 Locations.
- **BError** getSelectionInfo (SelectionGroup group, Selection selectionIn, SelectionInfo &selectionInfo)

  Get information on possible selections. Use in GUI programs to list options available.
- **BError** getSelections (SelectionGroup group, Selection selectionIn, Selection &selectionOut)

  Get selection list.
- **BError** dataAvailability (Selection selection, **BUInt32** num, **BArray**< DataAvailChan > &dataAvailChans)

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

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

  Return the channel MetaData in structured form.
- **BError** dataOpen (DataInfo dataInfo, **BString** mode, **BString** format, **BUInt32** flags, DataHandle &data↩Handle)

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

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

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

  Get information on the data file.
- **BError** dataSeekBlock (DataHandle dataHandle, **BUInt32** channel, **BUInt32** segment, **BTimeStamp** time, **BUInt32** &blockNumber)

  Searches for a data block matching the time given.
- **BError** dataGetBlock (DataHandle dataHandle, **BUInt32** channel, **BUInt32** segment, **BUInt32** block↩Number, DataBlock & **data**)

  Return a block of data.
- **BError** dataSetInfo (DataHandle dataHandle, DataInfo dataInfo)

  Set the info when writing to a file.
- **BError** dataPutBlock (DataHandle dataHandle, DataBlock **data**)

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

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

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

  The total length in bytes of the formated data.
- **BError** noteGetList (Selection sel, **BList**< Note > &notes)

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

  Add or update a Note.
- **BError** noteWriteDocument (**BUInt32** id, **BString** format, **BArray**< **BUInt8** > **data**)

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

  Read a document associated with a Note.
- **BError** logUpdate (**BInt32** append, Log log, **BUInt32** &id)

  Append a log item ///< Add or update a Log item.
- **BError** logAppend (**BString** type, **BUInt32** priority, **BString** subSystem, **BString** title, **BString** description)

  Append a log item.
- **BError** modeSet (Mode mode, Mode &previousMode)

  Changes the system mode from Master to slave.

- **BError** modeSnapshotPause (**BInt32** on)

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

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

  Backup the database.
- **BError** statisticsGet (**BDict**< **BString** > &info)

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

  Get list of data formats.

Additional Inherited Members

7.28.1 Detailed Description

This is the DataAdd Access API interface.

This object provides the set of API functions that make RPC network calls to a BdsServer. It provides the basic and restricted user orientated read only, data access API along with the ability to import data. It will normally be used by data import client programs.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 DataAddAccess()

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

7.28.3 Member Function Documentation

7.28.3.1 calibrationGetList()

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

Get list of Calibrations.

7.28.3.2 channelGetList()

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

Get list of Channels.

7.28.3.3 channelInstrumentGetList()

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

Get list of Instruments.

7.28.3.4 clean()

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

Cleans the system logs and Changes information.

7.28.3.5 connect()

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

Provides user/password information.

7.28.3.6 dataAvailability()

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

Return availability for data matching the given selection parameters.

7.28.3.7 databaseBackup()

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

Backup the database.

7.28.3.8 dataChannelGetList()

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

Get list of DataChannels.

7.28.3.9 dataClose()

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

Close a file.

7.28.3.10 dataFileGetList()

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

Get list of DataFiles.

7.28.3.11 dataFormatGetList()

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

Get list of data formats.

7.28.3.12 dataFormattedGetLength()

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

The total length in bytes of the formated 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	*dataInfo*	The sensor data to open selection
in	*mode*	The open format. The mode should be set to "w" for writing data, "a" when appending data and "r" for reading data.
in	*format*	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	*flags*	A bitset of flags as defined by Bds::DataFlags.
out	*dataHandle*	The handle for the open data set

7.28.3.20 dataPutBlock()

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

Send a block of data.

7.28.3.21 dataSearch()

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

Search for data matching the given selection parameters.

7.28.3.22 dataSeekBlock()

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

Searches for a data block matching the time given.

7.28.3.23 dataSetInfo()

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

Set the info when writing to a file.

7.28.3.24 digitiserGet()

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

Get a Digitiser object given its ID.

7.28.3.25 digitiserGetList()

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

Get list of Channels.

7.28.3.26 getSelectionInfo()

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

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

7.28.3.27 getSelections()

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

Get selection list.

7.28.3.28 getVersion()

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

Gets the software version and server name.

7.28.3.29 groupGetList()

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

Get list of Groups.

7.28.3.30 locationGetList()

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

Get list of Station Locations.

7.28.3.31 logAppend()

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

Append a log item.

7.28.3.32 logUpdate()

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

Append a log item ///< Add or update a Log item.

7.28.3.33 modeSet()

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

Changes the system mode from Master to slave.

7.28.3.34 modeSnapshotPause()

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

Enables/disables backup synchronisation pause.

7.28.3.35 networkGetList()

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

Get list of Networks.

7.28.3.36 noteGetList()

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

Return a list of Notes.

7.28.3.37 noteReadDocument()

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

Read a document associated with a Note.

7.28.3.38 noteUpdate()

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

Add or update a Note.

7.28.3.39 noteWriteDocument()

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

Given a Note write a document associated with it.

7.28.3.40 responseGetList()

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

Get list of Responses.

7.28.3.41 sensorGet()

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

Get a Sensor object given its ID.

7.28.3.42 sensorGetList()

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

Get list of Sensors.

7.28.3.43 setUser()

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

Sets user to given name or email.

7.28.3.44 setUserReal()

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

Sets user back to real user.

7.28.3.45 sourceGetList()

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

Get list of Sources.

7.28.3.46 sourcePriorityGetList()

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

Get list of SourcePriorities.

7.28.3.47 stationGetList()

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

Get list of Stations.

7.28.3.48 statisticsGet()

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

Get a list of system statistics.

7.28.3.49 userGet()

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

Get user info.

7.28.3.50 userGetFromId()

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

Get user info given user ID.

7.28.3.51 userGetGroups()

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

Get list of groups the user belongs to.

7.28.3.52 userSet()

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

Set user info.

7.28.3.53 validateUser()

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

Checks the user given name or email.

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

- BdsC.h
- BdsC.cc
- BdsLib.dox

7.29 Bds::DataAvail Class Reference

This class provides availability information on a particular period of data.

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

Public Member Functions

- DataAvail (**BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), AvailType availType=AvailType())

Public Attributes

- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time.
- AvailType availType

  The availability type. Can be: AvailNone, AvailPartial or AvailFull.

7.29.1 Detailed Description

This class provides availability information on a particular period of data.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 DataAvail()

```
Bds::DataAvail::DataAvail (
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
           AvailType availType = AvailType() )
```

7.29.3 Member Data Documentation

7.29.3.1 availType

```
AvailType Bds::DataAvail::availType
```

The availability type. Can be: AvailNone, AvailPartial or AvailFull.

7.29.3.2 endTime

```
BTimeStamp Bds::DataAvail::endTime
```

The End Time.

7.29.3.3 startTime

```
BTimeStamp Bds::DataAvail::startTime
```

The Start Time.

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

- BdsD.h
- BdsD.cc

7.30 Bds::DataAvailChan Class Reference

This class defines avaiilablity information on a set of data.

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

Public Member Functions

- DataAvailChan (**BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), **B**↩ **String** 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 avaiilablity 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 acual 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= **B**↩ **Array**< **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 acual 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 varible, the endTime fields may be lined up with the next blocks startTime field to ensure contiguous data segments. The info field contains extra, free string format, metadata on the block if available. This could be quality information from the TapeDigitiser system for example.

7.31.2 Constructor & Destructor Documentation

7.31.2.1 DataBlock()

```
Bds::DataBlock::DataBlock (
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BUInt32 channelNumber = 0,
            BUInt32 segmentNumber = 0,
            BArray< BArray< BFloat64 > > channelData = BArray< BArray< BFloat64 > >(),
            BDict< BString > info = BDict< BString >() )
```

7.31.3 Member Data Documentation

7.31.3.1 channelData

```
BArray< BArray< BFloat64 > > Bds::DataBlock::channelData
```

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

7.31.3.2 channelNumber

```
BUInt32 Bds::DataBlock::channelNumber
```

The first channel number. (1, 2, 3 ...)

7.31.3.3 endTime

```
BTimeStamp Bds::DataBlock::endTime
```

The End Time the channel was available.

7.31.3.4 info

```
BDict< BString > Bds::DataBlock::info
```

Extra information on data or ASCII data.

7.31.3.5 segmentNumber

```
BUInt32 Bds::DataBlock::segmentNumber
```

The segment number. (1, 2, 3, ...)

7.31.3.6 startTime

```
BTimeStamp Bds::DataBlock::startTime
```

The Start Time.

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

- BdsD.h
- BdsD.cc

7.32 Bds::DataBlockPos Class Reference

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

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

Public Member Functions

- DataBlockPos (**BTimeStamp** startTime=0, **BTimeStamp** endTime=0, **BUInt64** position=0, **BUInt** order=0, int ref=0, **BUInt** numSamples=0)
- int operator< (const DataBlockPos &b) const

Public Attributes

- **BTimeStamp** startTime
- **BTimeStamp** endTime
- **BUInt64** position
- **BUInt** order
- int ref
- **BUInt** numSamples

7.32.1 Detailed Description

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

7.32.2 Constructor & Destructor Documentation

7.32.2.1 DataBlockPos()

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

7.32.3 Member Function Documentation

7.32.3.1 operator<()

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

7.32.4 Member Data Documentation

7.32.4.1 endTime

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

7.32.4.2 numSamples

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

7.32.4.3 order

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

7.32.4.4 position

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

7.32.4.5 ref

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

7.32.4.6 startTime

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

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

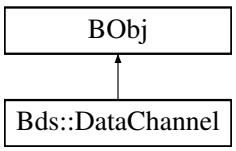
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h

7.33 Bds::DataChannel Class Reference

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

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

Inheritance diagram for Bds::DataChannel:



Public Member Functions

- DataChannel (**BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTime↩Stamp**(), **BString** network= **BString**(), **BString** station= **BString**(), **BString** channel= **BString**(), **BString** source= **BString**(), **BUInt32** numBlocks=0, **BUInt64** numSamples=0, **BFloat64** sampleRate=0, **BUInt32** sampleFormat=0, **BUInt32** dataFileId=0, **BUInt32** dataFileChannel=0, **BString** importFormat= **BString**(), **BString** importFilename= **BString**(), **BTimeStamp** importStartTime= **BTimeStamp**(), **BDict**< **BString** > info= **BDict**< **BString** >())
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  Unique ID when stored in a database or for other uses.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time.
- **BString** network

  The Network Name.
- **BString** station

  The Station name.
- **BString** channel

  The Channels name.
- **BString** source

  The Data Source.
- **BUInt32** numBlocks

  The total number of blocks per channel if known, 0 otherwise.
- **BUInt64** numSamples

  The total number of samples per channel if known, 0 otherwise.
- **BFloat64** sampleRate

  The data's sample rate.
- **BUInt32** sampleFormat

  The data sample format.
- **BUInt32** dataFileId

  The Data File Id. This links to the particular DataFileInfo where the data is stored.
- **BUInt32** dataFileChannel

  The Data File Channel number. The channel number within the data file. (1, 2, 3 ...)
- **BString** importFormat

  The original data format.
- **BString** importFilename

  The original data file name.
- **BTimeStamp** importStartTime

  The original import files start time.
- **BDict**< **BString** > info

  Extra info on the channel.

7.33.1 Detailed Description

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

This provides information on actual seismic data for a channel that is stored in the BDS system. The data will be stored in a particular file perhaps with other data channels. When known information on the channels numBlocks, numSamples and sampleRate will be provided. Generally this information will only be know if a data files has been imported rather than a live real-time data stream. Generally the seismic data file itself should be interrogated to find the definitive information. The info field provides extra details on the data contents which might come from one of the spefic data import formats.

7.33.2 Constructor & Destructor Documentation

7.33.2.1 DataChannel()

```
Bds::DataChannel::DataChannel (
            BUInt32 id = 0,
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BString network = BString(),
            BString station = BString(),
            BString channel = BString(),
            BString source = BString(),
            BUInt32 numBlocks = 0,
            BUInt64 numSamples = 0,
            BFloat64 sampleRate = 0,
            BUInt32 sampleFormat = 0,
            BUInt32 dataFileId = 0,
            BUInt32 dataFileChannel = 0,
            BString importFormat = BString(),
            BString importFilename = BString(),
            BTimeStamp importStartTime = BTimeStamp(),
            BDict< BString > info = BDict< BString >() )
```

7.33.3 Member Function Documentation

7.33.3.1 getMember()

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

Reimplemented from **BObj**.

7.33.3.2 getMembers()

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

Reimplemented from **BObj**.

7.33.3.3 getType()

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

7.33.3.4 setMember()

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

Reimplemented from **BObj**.

7.33.3.5 setMembers()

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

Reimplemented from **BObj**.

7.33.4 Member Data Documentation

7.33.4.1 channel

```
BString Bds::DataChannel::channel
```

The Channels name.

7.33.4.2 dataFileChannel

```
BUInt32 Bds::DataChannel::dataFileChannel
```

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

7.33.4.3 dataFileId

```
BUInt32 Bds::DataChannel::dataFileId
```

The Data File Id. This links to the particular DataFileInfo where the data is stored.

7.33.4.4 endTime

```
BTimeStamp Bds::DataChannel::endTime
```

The End Time.

7.33.4.5 id

```
BUInt32 Bds::DataChannel::id
```

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

7.33.4.6 importFilename

```
BString Bds::DataChannel::importFilename
```

The original data file name.

7.33.4.7 importFormat

```
BString Bds::DataChannel::importFormat
```

The original data format.

7.33.4.8 importStartTime

```
BTimeStamp Bds::DataChannel::importStartTime
```

The original import files start time.

7.33.4.9 info

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

Extra info on the channel.

7.33.4.10 network

```
BString Bds::DataChannel::network
```

The Network Name.

7.33.4.11 numBlocks

```
BUInt32 Bds::DataChannel::numBlocks
```

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

7.33.4.12 numSamples

```
BUInt64 Bds::DataChannel::numSamples
```

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

7.33.4.13 sampleFormat

```
BUInt32 Bds::DataChannel::sampleFormat
```

The data sample format.

7.33.4.14 sampleRate

```
BFloat64 Bds::DataChannel::sampleRate
```

The data's sample rate.

7.33.4.15 source

```
BString Bds::DataChannel::source
```

The Data Source.

7.33.4.16 startTime

```
BTimeStamp Bds::DataChannel::startTime
```

The Start Time.

7.33.4.17 station

```
BString Bds::DataChannel::station
```

The Station name.

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

- BdsD.h
- BdsD.cc

7.34 Bds::DataCollate Class Reference

Not sure if this is used or what it does.

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

Public Member Functions

- DataCollate ()
- ~DataCollate ()
- **BError** addSource (DataFile &dataFile, **BUInt** channel)
- **BError** readData (**BUInt32** blockNumber, DataBlock & **data**)

7.34.1 Detailed Description

Not sure if this is used or what it does.

7.34.2 Constructor & Destructor Documentation

7.34.2.1 DataCollate()

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

7.34.2.2 ~DataCollate()

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

7.34.3 Member Function Documentation

7.34.3.1 addSource()

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

7.34.3.2 readData()

```
BError Bds::DataCollate::readData (
             BUInt32 blockNumber,
            DataBlock & data )
```

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

- /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp

7.35 Bds::DataError Class Reference

This stores a data error. It includes and 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** end↩ Time, DataInfo &dataInfo, **BUInt** channel, **BString** description, **BString** user="")
- DataError & set (int errorNumber, **BString** title, **BString** importFilename, **BTimeStamp** startTime, **B**↩ **TimeStamp** 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.

7.35.1 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.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 DataError() [1/2]

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

7.35.2.2 DataError() [2/2]

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

7.35.3 Member Function Documentation

7.35.3.1 getErrorNumber()

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

Get The error number.

7.35.3.2 getString()

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

Get error message.

7.35.3.3 getTitle()

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

Get the title.

7.35.3.4 mergeDataInfo()

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

7.35.3.5 num()

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

Get The error number.

7.35.3.6 operator int()

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

Return error number.

7.35.3.7 set()

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

7.35.3.8 setString()

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

Set from string.

7.35.3.9 setStringUser()

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

Set from string given by user on command line.

7.35.3.10 str()

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

Return a char∗ string.

7.35.4 Member Data Documentation

7.35.4.1 ochannel

```
BString Bds::DataError::ochannel
```

The channel name.

7.35.4.2 odescription

```
BString Bds::DataError::odescription
```

The description.

7.35.4.3 oendTime

```
BTimeStamp Bds::DataError::oendTime
```

The end Time.

7.35.4.4 oerrorNumber

```
BInt32 Bds::DataError::oerrorNumber
```

Error number.

7.35.4.5 ofilename

```
BString Bds::DataError::ofilename
```

The import filename;.

7.35.4.6 onetwork

```
BString Bds::DataError::onetwork
```

The network Name.

7.35.4.7 osource

```
BString Bds::DataError::osource
```

The data Source.

7.35.4.8 ostartTime

```
BTimeStamp Bds::DataError::ostartTime
```

The start Time.

7.35.4.9 ostation

```
BString Bds::DataError::ostation
```

The station/array name.

7.35.4.10 otitle

```
BString Bds::DataError::otitle
```

The title.

7.35.4.11 ouser

```
BString Bds::DataError::ouser
```

The user.

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

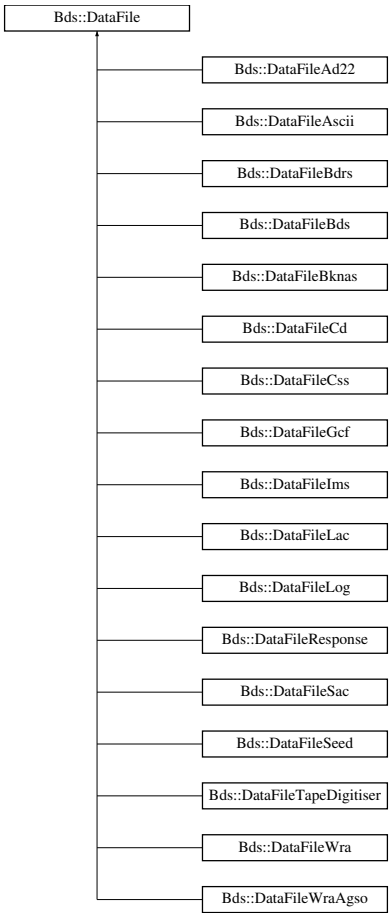
- BdsLib.h
- BdsLib.cpp

7.36 Bds::DataFile Class Reference

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

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

Inheritance diagram for Bds::DataFile:



Public Types

- enum DataOrder { DataOrderUnknown, DataOrderAll, DataOrderSample, DataOrderChannel }
- enum Features { FeatureNone = 0x00, FeatureCanWrite = 0x01, FeatureCanRead = 0x02 }
- enum WriteOptionsList { WriteOptionNone = 0x00, WriteOptionSensorData = 0x01, WriteOptionNoMetadata = 0x02 }
- enum ReadOptionsList { ReadOptionNone = 0x00, ReadOptionValidate = 0x01, ReadOptionFileNameProcess = 0x02, ReadOptionValidateCorruptions = 0x04, ReadOptionReorder = 0x08, ReadOptionDeleteDuplicates = 0x10, ReadOptionInfoExtra = 0x20, ReadOptionIgnoreSamplerate = 0x40, ReadOptionPrintBlocks = 0x80 }

Public Member Functions

- DataFile ()
- virtual ~DataFile ()
- virtual void init ()

  Initialise.
- virtual **BError** open (**BString** fileName, **BString** mode)

  Open the file for read or write.
- virtual **BError** close ()

  Close the file.
- virtual **BError** setFormat (**BString** format)

Set the sub-format.

- virtual **BString** getFileName ()

  Return the file name.
- virtual DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- virtual int getFeatures ()

  Get bitmask of supported features.
- virtual **BError** setInfo (const DataInfo &dataInfo, const ChannelInfos &channelInfos, WriteOptionsList options=WriteOptionNone)

  Set information on data for write.
- virtual **BError** start (**BUInt** channel, **BUInt** segment)

  Start writing next segment of data.
- virtual **BError** writeData (const DataBlock & **data**)

  Write a block of data.
- virtual **BError** end ()

  End write segment.
- virtual **BError** flush ()

  Flush data to disk.
- virtual **BError** fileNameProcess ()

  Parse the file name for a date/time.
- virtual **BError** getFormat (**BString** &format)

  Get sub-format.
- virtual **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- virtual **BError** seekBlock (**BUInt32** channel, **BUInt** segment, **BTimeStamp** time, **BUInt32** &blockNumber, **BUInt64** &sampleNumber, DataBlock & **data**)

  Find requested block on given channel given a time.
- virtual **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock &dataBlock)

  Read a block.
- virtual **BError** getMetaData (ChannelInfos &channelInfos)

  Return all known MetaData in the file.
- void dataErrorFixup (const DataInfo &dataInfo, **BList**< DataError > &errors)

  Fixup data errors, mainly start/end times to be within data.
- **BInt64** timeCompare (**BTimeStamp** t1, **BTimeStamp** t2, **BUInt** diff)

  Compare timestamps with a margin.
- int duplicateCheck (const DataBlock &data1, const DataBlock &data2, **BUInt** channel=0)

  Check if blocks are duplicates.
- **BUInt64** getFilePosition ()

Static Public Member Functions

- static DataFormat getFormats ()

  Get list of supported formats.

Protected Attributes

- **BString** ofileName
- **BString** omode
- **BTimeStamp** ofileNameTime
- **BFile** ofile
- **BString** oformat

7.36.1 Detailed Description

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

7.36.2 Member Enumeration Documentation

7.36.2.1 DataOrder

`enum Bds::DataFile::DataOrder`

Enumerator

DataOrderUnknown	
DataOrderAll	
DataOrderSample	
DataOrderChannel	

7.36.2.2 Features

`enum Bds::DataFile::Features`

Enumerator

FeatureNone	
FeatureCanWrite	
FeatureCanRead	

7.36.2.3 ReadOptionsList

`enum Bds::DataFile::ReadOptionsList`

Enumerator

ReadOptionNone	
ReadOptionValidate	
ReadOptionFileNameProcess	
ReadOptionValidateCorruptions	
ReadOptionReorder	
ReadOptionDeleteDuplicates	
ReadOptionInfoExtra	
ReadOptionIgnoreSamplerate	
ReadOptionPrintBlocks	

7.36.2.4 WriteOptionsList

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

Enumerator

WriteOptionNone	
WriteOptionSensorData	
WriteOptionNoMetadata	

7.36.3 Constructor & Destructor Documentation

7.36.3.1 DataFile()

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

7.36.3.2 ~DataFile()

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

7.36.4 Member Function Documentation

7.36.4.1 close()

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

Close the file.

Reimplemented in Bds::DataFileBds, Bds::DataFileSeed, and Bds::DataFileIms.

7.36.4.2 dataErrorFixup()

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

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

7.36.4.3 duplicateCheck()

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

Check if blocks are duplicates.

7.36.4.4 end()

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

End write segment.

Reimplemented in Bds::DataFileSeed, Bds::DataFileLog, Bds::DataFileAscii, and Bds::DataFileIms.

7.36.4.5 fileNameProcess()

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

Parse the file name for a date/time.

7.36.4.6 flush()

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

Flush data to disk.

Reimplemented in Bds::DataFileBds.

7.36.4.7 getDataOrder()

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

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

Reimplemented in Bds::DataFileBds, Bds::DataFileCd, Bds::DataFileCss, Bds::DataFileGcf, Bds::DataFileSeed, Bds::DataFileIms, Bds::DataFileAscii, Bds::DataFileLog, Bds::DataFileWra, Bds::DataFileAd22, Bds::DataFileBdrs, Bds::DataFileLac, and Bds::DataFileWraAgso.

7.36.4.8 getFeatures()

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

Get bitmask of supported features.

Reimplemented in Bds::DataFileCd, Bds::DataFileCss, Bds::DataFileGcf, Bds::DataFileSeed, Bds::DataFileIms, Bds::DataFileAscii, Bds::DataFileLog, Bds::DataFileResponse, Bds::DataFileSac, Bds::DataFileWra, Bds::DataFileAd22, Bds::DataFileBdrs, Bds::DataFileLac, and Bds::DataFileWraAgso.

7.36.4.9 getFileName()

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

Return the file name.

7.36.4.10 getFilePosition()

```
BUInt64 Bds::DataFile::getFilePosition ( )
```

7.36.4.11 getFormat()

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

Get sub-format.

7.36.4.12 getFormats()

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

Get list of supported formats.

7.36.4.13 getInfo()

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

Get info on data.

Reimplemented in Bds::DataFileBds, Bds::DataFileCd, Bds::DataFileCss, Bds::DataFileGcf, Bds::DataFileSeed, Bds::DataFileTapeDigitiser, Bds::DataFileLog, Bds::DataFileWra, Bds::DataFileAd22, Bds::DataFileBdrs, Bds::DataFileLac, and Bds::DataFileWraAgso.

7.36.4.14 getMetaData()

```
BError Bds::DataFile::getMetaData (
            ChannelInfos & channelInfos ) [virtual]
```

Return all known MetaData in the file.

Reimplemented in Bds::DataFileIms, and Bds::DataFileResponse.

7.36.4.15 init()

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

Initialise.

7.36.4.16 open()

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

Open the file for read or write.

Reimplemented in Bds::DataFileBds, Bds::DataFileTapeDigitiser, Bds::DataFileAscii, Bds::DataFileBknas, Bds::DataFileIms, and Bds::DataFileLog.

7.36.4.17 readData()

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

Read a block.

Reimplemented in Bds::DataFileBds, Bds::DataFileCd, Bds::DataFileCss, Bds::DataFileGcf, Bds::DataFileSeed, Bds::DataFileTapeDigitiser, Bds::DataFileLog, Bds::DataFileWra, Bds::DataFileAd22, Bds::DataFileBdrs, Bds::DataFileLac, and Bds::DataFileWraAgso.

7.36.4.18 seekBlock()

```
BError Bds::DataFile::seekBlock (
            BUInt32 channel,
            BUInt segment,
            BTimeStamp time,
            BUInt32 & blockNumber,
            BUInt64 & sampleNumber,
           DataBlock & data )  [virtual]
```

Find requested block on given channel given a time.

Reimplemented in Bds::DataFileBds.

7.36.4.19 setFormat()

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

Set the sub-format.

Reimplemented in Bds::DataFileBds, Bds::DataFileLog, Bds::DataFileSeed, Bds::DataFileAscii, and Bds::DataFileWra.

7.36.4.20 setInfo()

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

Set information on data for write.

Reimplemented in Bds::DataFileBds, Bds::DataFileSeed, Bds::DataFileLog, Bds::DataFileAscii, Bds::DataFileIms, Bds::DataFileResponse, Bds::DataFileBknas, and Bds::DataFileSac.

7.36.4.21 start()

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

Start writing next segment of data.

Reimplemented in Bds::DataFileSeed, Bds::DataFileLog, Bds::DataFileAscii, and Bds::DataFileIms.

7.36.4.22 timeCompare()

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

Compare timestamps with a margin.

7.36.4.23 writeData()

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

Write a block of data.

Reimplemented in Bds::DataFileBds, Bds::DataFileSeed, Bds::DataFileLog, Bds::DataFileAscii, Bds::DataFileIms, and Bds::DataFileBknas.

7.36.5 Member Data Documentation

7.36.5.1 ofile

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

7.36.5.2 ofileName

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

7.36.5.3 ofileNameTime

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

7.36.5.4 oformat

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

7.36.5.5 omode

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

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

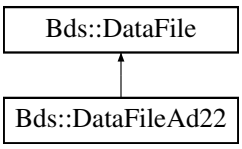
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp

7.37 Bds::DataFileAd22 Class Reference

Data file convertor for AD22 format files.

```
#include <BdsDataFileAd22.h>
```

Inheritance diagram for Bds::DataFileAd22:



Public Member Functions

- DataFileAd22 ()
- int getFeatures ()

  Get bitmask of supported features.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock & **data**)

  Read a block.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.37.1 Detailed Description

Data file convertor for AD22 format files.

7.37.2 Constructor & Destructor Documentation

7.37.2.1 DataFileAd22()

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

7.37.3 Member Function Documentation

7.37.3.1 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.37.3.2 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.37.3.3 getFormats()

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

7.37.3.4 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.37.3.5 readData()

```
BError Bds::DataFileAd22::readData (
             BUInt32 channel,
             BUInt segment,
             BUInt32 blockNumber,
            DataBlock & dataBlock ) [virtual]
```

Read a block.

Reimplemented from Bds::DataFile.

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

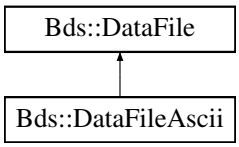
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp

7.38 Bds::DataFileAscii Class Reference

Data file convertor for ASCII format files.

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

Inheritance diagram for Bds::DataFileAscii:



Public Member Functions

- DataFileAscii ()
- **BError** open (**BString** fileName, **BString** mode)

  Open the file for read or write.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- int getFeatures ()

  Get bitmask of supported features.
- **BError** setFormat (**BString** format)

  Set the sub-format.
- **BError** setInfo (const DataInfo &dataInfo, const ChannelInfos &channelInfos, WriteOptionsList options=WriteOptionSensorData)

  Set information on data for write.
- **BError** start (**BUInt** channel, **BUInt** segment)

  Start writing next segment of data.
- **BError** writeData (const DataBlock & **data**)

  Write a block of data.
- **BError** end ()

  End write segment.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.38.1 Detailed Description

Data file convertor for ASCII format files.

7.38.2 Constructor & Destructor Documentation

7.38.2.1 DataFileAscii()

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

7.38.3 Member Function Documentation

7.38.3.1 end()

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

End write segment.

Reimplemented from Bds::DataFile.

7.38.3.2 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.38.3.3 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.38.3.4 getFormats()

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

7.38.3.5 open()

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

Open the file for read or write.

Reimplemented from Bds::DataFile.

7.38.3.6 setFormat()

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

Set the sub-format.

Reimplemented from Bds::DataFile.

7.38.3.7 setInfo()

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

Set information on data for write.

Reimplemented from Bds::DataFile.

7.38.3.8 start()

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

Start writing next segment of data.

Reimplemented from Bds::DataFile.

7.38.3.9 writeData()

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

Write a block of data.

Reimplemented from Bds::DataFile.

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

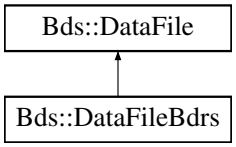
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp

7.39 Bds::DataFileBdrs Class Reference

Data file convertor for BDRS format files.

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

Inheritance diagram for Bds::DataFileBdrs:



Public Member Functions

- DataFileBdrs ()
- int getFeatures ()

  Get bitmask of supported features.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock & **data**)

  Read a block.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.39.1 Detailed Description

Data file convertor for BDRS format files.

7.39.2 Constructor & Destructor Documentation

7.39.2.1 DataFileBdrs()

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

7.39.3 Member Function Documentation

7.39.3.1 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.39.3.2 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.39.3.3 getFormats()

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

7.39.3.4 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.39.3.5 readData()

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

Read a block.

Reimplemented from Bds::DataFile.

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

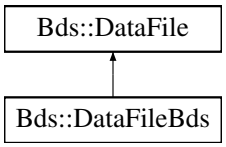
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp

7.40 Bds::DataFileBds Class Reference

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

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

Inheritance diagram for Bds::DataFileBds:



Public Types

- enum { StreamsMax = 256 }
- enum { DefaultBlockSize = 65536 }
- enum PackFormat { PackFormat_Unknown = 0, PackFormat_SM = 1, PackFormat_CM = 2, PackFormat_SM_CC = 3 }

Public Member Functions

- DataFileBds ()
- ~DataFileBds ()
- **BError** open (**BString** fileName, **BString** mode)

  Open the file for reading or writing.
- **BError** flush ()

  Flush any data to disk even if blocks are not full.
- **BError** close ()

  Close file.
- **BError** setFormat (**BString** format)

  Sets the sub-format.

- **BError** setInfo (const DataInfo &dataInfo, const ChannelInfos &channelInfos, WriteOptionsList options=WriteOptionSensorData)

  Sets the information.
- **BError** writeData (const DataBlock & **data**)

  Writes a data block to the file.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get information on open file.
- **BError** seekBlock (**BUInt32** channel, **BUInt** segment, **BTimeStamp** time, **BUInt32** &blockNumber, **BUInt64** &sampleNumber, DataBlock &dataBlock)

  Find the block that contains the samples for the time requested.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock &dataBlock)

  Read the data block for the given channel or all channels if blockNumber is 0.
- **BError** setDiskBlockSize (**BUInt32** blockSize)

  Sets up file/stream block size.
- **BUInt32** getDiskBlockSize ()

  Returns the data block size in bytes.

Static Public Member Functions

- static DataFormat getFormats ()

  Get the names of the supported formats.

Additional Inherited Members

7.40.1 Detailed Description

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

7.40.2 Member Enumeration Documentation

7.40.2.1 anonymous enum

```
anonymous enum
```

Enumerator

StreamsMax	

7.40.2.2 anonymous enum

```
anonymous enum
```

Enumerator

DefaultBlockSize	

7.40.2.3 PackFormat

```
enum Bds::DataFileBds::PackFormat
```

Enumerator

PackFormat_Unknown	
PackFormat_SM	
PackFormat_CM	
PackFormat_SM_CC	

7.40.3 Constructor & Destructor Documentation

7.40.3.1 DataFileBds()

```
Bds::DataFileBds::DataFileBds ( )
```

7.40.3.2 ~DataFileBds()

```
Bds::DataFileBds::~DataFileBds ( )
```

7.40.4 Member Function Documentation

7.40.4.1 close()

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

Close file.

Reimplemented from Bds::DataFile.

7.40.4.2 flush()

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

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

Reimplemented from Bds::DataFile.

7.40.4.3 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.40.4.4 getDiskBlockSize()

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

Returns the data block size in bytes.

7.40.4.5 getFormats()

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

Get the names of the supported formats.

7.40.4.6 getInfo()

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

Get information on open file.

Reimplemented from Bds::DataFile.

7.40.4.7 open()

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

Open the file for reading or writing.

Reimplemented from Bds::DataFile.

7.40.4.8 readData()

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

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

Reimplemented from Bds::DataFile.

7.40.4.9 seekBlock()

```
BError Bds::DataFileBds::seekBlock (
            BUInt32 channel,
            BUInt segment,
            BTimeStamp time,
            BUInt32 & blockNumber,
            BUInt64 & sampleNumber,
           DataBlock & dataBlock )  [virtual]
```

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

Reimplemented from Bds::DataFile.

7.40.4.10 setDiskBlockSize()

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

Sets up file/stream block size.

7.40.4.11 setFormat()

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

Sets the sub-format.

Reimplemented from Bds::DataFile.

7.40.4.12 setInfo()

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

Sets the information.

Reimplemented from Bds::DataFile.

7.40.4.13 writeData()

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

Writes a data block to the file.

Reimplemented from Bds::DataFile.

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

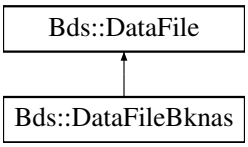
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp

7.41 Bds::DataFileBknas Class Reference

Data file convertor for BKNAS format files.

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

Inheritance diagram for Bds::DataFileBknas:



Public Member Functions

- DataFileBknas ()
- **BError** open (**BString** fileName, **BString** mode)

  Open the file for read or write.
- **BError** setInfo (const DataInfo &dataInfo, const ChannelInfos &channelInfos, WriteOptionsList options=WriteOptionSensorData)

  Set information on data for write.
- **BError** writeData (const DataBlock & **data**)

  Write a block of data.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.41.1 Detailed Description

Data file convertor for BKNAS format files.

7.41.2 Constructor & Destructor Documentation

7.41.2.1 DataFileBknas()

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

7.41.3 Member Function Documentation

7.41.3.1 getFormats()

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

7.41.3.2 open()

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

Open the file for read or write.

Reimplemented from Bds::DataFile.

7.41.3.3 setInfo()

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

Set information on data for write.

Reimplemented from Bds::DataFile.

7.41.3.4 writeData()

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

Write a block of data.

Reimplemented from Bds::DataFile.

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

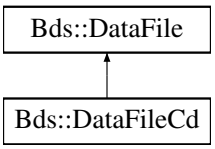
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp

7.42 Bds::DataFileCd Class Reference

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

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

Inheritance diagram for Bds::DataFileCd:



Public Member Functions

- DataFileCd ()
- int getFeatures ()

  Get bitmask of supported features.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock & **data**)

  Read a block.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.42.1 Detailed Description

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

7.42.2 Constructor & Destructor Documentation

7.42.2.1 DataFileCd()

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

7.42.3 Member Function Documentation

7.42.3.1 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.42.3.2 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.42.3.3 getFormats()

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

7.42.3.4 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.42.3.5 readData()

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

Read a block.

Reimplemented from Bds::DataFile.

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

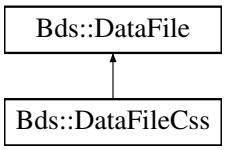
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.cpp

7.43 Bds::DataFileCss Class Reference

Data file convertor for CSS format files.

```
#include <BdsDataFileCss.h>
```

Inheritance diagram for Bds::DataFileCss:



Public Member Functions

- DataFileCss ()
- int getFeatures ()

  Get bitmask of supported features.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock & **data**)

  Read a block.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.43.1 Detailed Description

Data file convertor for CSS format files.

7.43.2 Constructor & Destructor Documentation

7.43.2.1 DataFileCss()

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

7.43.3 Member Function Documentation

7.43.3.1 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.43.3.2 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.43.3.3 getFormats()

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

7.43.3.4 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.43.3.5 readData()

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

Read a block.

Reimplemented from Bds::DataFile.

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

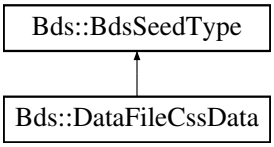
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp

7.44 Bds::DataFileCssData Class Reference

DataFileCss internal CSS data type.

```
#include <BdsDataFileCss.h>
```

Inheritance diagram for Bds::DataFileCssData:



Public Member Functions

- DataFileCssData ()
- ~DataFileCssData ()
- **BError** set (**BString** line)

Public Attributes

- **BString** sta
- **BString** chan
- double startTime
- int wfid
- int chanid
- int jdate
- double endTime
- int nsamp
- double sampleRate
- double calibrationFactor
- double calibrationFreq

- **BString** instType
- **BString** segtype
- **BString** datatype
- **BString** clip
- **BString** dirName
- **BString** fileName
- **BUInt32** fileOffset
- int commId
- **BString** loadDate
- **BFile** ∗ file
- **BUInt32** sampleFormat
- **BUInt32** sampleSize
- int sampleBigEndian

7.44.1 Detailed Description

DataFileCss internal CSS data type.

7.44.2 Constructor & Destructor Documentation

7.44.2.1 DataFileCssData()

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

7.44.2.2 ∼DataFileCssData()

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

7.44.3 Member Function Documentation

7.44.3.1 set()

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

7.44.4 Member Data Documentation

7.44.4.1 calibrationFactor

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

7.44.4.2 calibrationFreq

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

7.44.4.3 chan

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

7.44.4.4 chanid

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

7.44.4.5 clip

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

7.44.4.6 commId

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

7.44.4.7 datatype

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

7.44.4.8 dirName

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

7.44.4.9 endTime

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

7.44.4.10 file

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

7.44.4.11 fileName

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

7.44.4.12 fileOffset

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

7.44.4.13 instType

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

7.44.4.14 jdate

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

7.44.4.15 loadDate

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

7.44.4.16 nsamp

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

7.44.4.17 sampleBigEndian

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

7.44.4.18 sampleFormat

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

7.44.4.19 sampleRate

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

7.44.4.20 sampleSize

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

7.44.4.21 segtype

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

7.44.4.22 sta

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

7.44.4.23 startTime

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

7.44.4.24 wfid

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

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

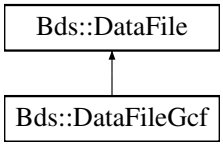
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCss.cpp

7.45 Bds::DataFileGcf Class Reference

Data file convertor for GCF format files.

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

Inheritance diagram for Bds::DataFileGcf:



Public Member Functions

- DataFileGcf ()
- int getFeatures ()

  Get bitmask of supported features.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock & **data**)

  Read a block.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.45.1 Detailed Description

Data file convertor for GCF format files.

7.45.2 Constructor & Destructor Documentation

7.45.2.1 DataFileGcf()

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

7.45.3 Member Function Documentation

7.45.3.1 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.45.3.2 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.45.3.3 getFormats()

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

7.45.3.4 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.45.3.5 readData()

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

Read a block.

Reimplemented from Bds::DataFile.

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

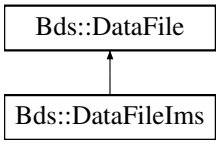
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.cpp

7.46 Bds::DataFileIms Class Reference

Data file convertor for IMS format files.

```
#include <BdsDataFileIms.h>
```

Inheritance diagram for Bds::DataFileIms:



Public Member Functions

- DataFileIms ()
- **BError** open (**BString** fileName, **BString** mode)

  Open the file for read or write.
- **BError** close ()

  Close the file.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- int getFeatures ()

  Get bitmask of supported features.
- **BError** setInfo (const DataInfo &dataInfo, const 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)

  Return all known MetaData in the file.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.46.1 Detailed Description

Data file convertor for IMS format files.

7.46.2 Constructor & Destructor Documentation

7.46.2.1 DataFileIms()

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

7.46.3 Member Function Documentation

7.46.3.1 close()

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

Close the file.

Reimplemented from Bds::DataFile.

7.46.3.2 end()

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

End write segment.

Reimplemented from Bds::DataFile.

7.46.3.3 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.46.3.4 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.46.3.5 getFormats()

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

7.46.3.6 getMetaData()

```
BError Bds::DataFileIms::getMetaData (
            ChannelInfos & channelInfos ) [virtual]
```

Return all known MetaData in the file.

Reimplemented from Bds::DataFile.

7.46.3.7 open()

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

Open the file for read or write.

Reimplemented from Bds::DataFile.

7.46.3.8 setInfo()

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

Set information on data for write.

Reimplemented from Bds::DataFile.

7.46.3.9 start()

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

Start writing next segment of data.

Reimplemented from Bds::DataFile.

7.46.3.10 writeData()

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

Write a block of data.

Reimplemented from Bds::DataFile.

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

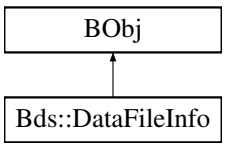
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.cpp

7.47 Bds::DataFileInfo Class Reference

This class defines information on a sensor data file.

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

Inheritance diagram for Bds::DataFileInfo:



Public Member Functions

- DataFileInfo (**BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTime**↩ **Stamp**(), **BString** location= **BString**(), **BString** format= **BString**(), **BString** url= **BString**(), **BString** comment= **BString**(), **BUInt32** importUserId=0, **BTimeStamp** importTime= **BTimeStamp**(), **BString** state= **BString**())
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  Unique ID when stored in a database or for other uses.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time.
- **BString** location

  The storage location.
- **BString** format

  The data format.
- **BString** url

  The URL for file access.
- **BString** comment

  A comment on the file.
- **BUInt32** importUserId

  The user ID of the importing user.
- **BTimeStamp** importTime

  The Time the data was imported.
- **BString** state

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

7.47.1 Detailed Description

This class defines information on a sensor data file.

The raw sensor data for a seismic channel is stored in files in the BDS system. This class defines the database entry that describes this file, its storage location and status. A single file can store one or more channels of seismic data in different formats.

7.47.2 Constructor & Destructor Documentation

7.47.2.1 DataFileInfo()

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

7.47.3 Member Function Documentation

7.47.3.1 getMember()

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

Reimplemented from **BObj**.

7.47.3.2 getMembers()

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

Reimplemented from **BObj**.

7.47.3.3 getType()

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

7.47.3.4 setMember()

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

Reimplemented from **BObj**.

7.47.3.5 setMembers()

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

Reimplemented from **BObj**.

7.47.4 Member Data Documentation

7.47.4.1 comment

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

A comment on the file.

7.47.4.2 endTime

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

The End Time.

7.47.4.3 format

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

The data format.

7.47.4.4 id

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

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

7.47.4.5 importTime

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

The Time the data was imported.

7.47.4.6 importUserId

```
BUInt32 Bds::DataFileInfo::importUserId
```

The user ID of the importing user.

7.47.4.7 location

```
BString Bds::DataFileInfo::location
```

The storage location.

7.47.4.8 startTime

```
BTimeStamp Bds::DataFileInfo::startTime
```

The Start Time.

7.47.4.9 state

```
BString Bds::DataFileInfo::state
```

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

7.47.4.10 url

```
BString Bds::DataFileInfo::url
```

The URL for file access.

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

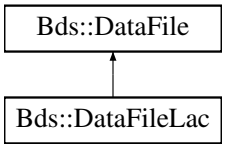
- BdsD.h
- BdsD.cc

7.48 Bds::DataFileLac Class Reference

Data file convertor for LAC format files.

```
#include <BdsDataFileLac.h>
```

Inheritance diagram for Bds::DataFileLac:



Public Member Functions

- DataFileLac ()
- int getFeatures ()

  Get bitmask of supported features.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock & **data**)

  Read a block.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.48.1 Detailed Description

Data file convertor for LAC format files.

7.48.2 Constructor & Destructor Documentation

7.48.2.1 DataFileLac()

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

7.48.3 Member Function Documentation

7.48.3.1 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.48.3.2 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.48.3.3 getFormats()

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

7.48.3.4 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.48.3.5 readData()

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

Read a block.

Reimplemented from Bds::DataFile.

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

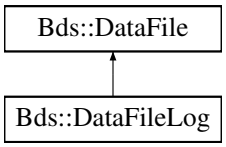
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp

7.49 Bds::DataFileLog Class Reference

Data file convertor for LOG format files.

```
#include <BdsDataFileLog.h>
```

Inheritance diagram for Bds::DataFileLog:



Public Member Functions

- DataFileLog ()
- **BError** open (**BString** fileName, **BString** mode)

  Open the file for read or write.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- int getFeatures ()

  Get bitmask of supported features.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock & **data**)

  Read a block.
- **BError** setFormat (**BString** format)

  Set the sub-format.
- **BError** setInfo (const DataInfo &dataInfo, const ChannelInfos &channelInfos, WriteOptionsList options=WriteOptionSensorData)

  Set information on data for write.
- **BError** start (**BUInt** channel, **BUInt** segment)

  Start writing next segment of data.
- **BError** writeData (const DataBlock & **data**)

  Write a block of data.
- **BError** end ()

  End write segment.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.49.1 Detailed Description

Data file convertor for LOG format files.

7.49.2 Constructor & Destructor Documentation

7.49.2.1 DataFileLog()

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

7.49.3 Member Function Documentation

7.49.3.1 end()

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

End write segment.

Reimplemented from Bds::DataFile.

7.49.3.2 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.49.3.3 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.49.3.4 getFormats()

```
DataFormat Bds::DataFileLog::getFormats ( ) [static]
```

7.49.3.5 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.49.3.6 open()

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

Open the file for read or write.

Reimplemented from Bds::DataFile.

7.49.3.7 readData()

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

Read a block.

Reimplemented from Bds::DataFile.

7.49.3.8 setFormat()

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

Set the sub-format.

Reimplemented from Bds::DataFile.

7.49.3.9 setInfo()

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

Set information on data for write.

Reimplemented from Bds::DataFile.

7.49.3.10 start()

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

Start writing next segment of data.

Reimplemented from Bds::DataFile.

7.49.3.11 writeData()

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

Write a block of data.

Reimplemented from Bds::DataFile.

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

- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.cpp

7.50 Bds::DataFileOptions Class Reference

This defines a list of BDS data convtor options.

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

Public Member Functions

- DataFileOptions (int options=0)
- operator int ()
- DataFileOptions & operator|= (int o)

Public Attributes

- int ooptionList
- **BArray**< **BUInt** > oignoreBlockList

7.50.1 Detailed Description

This defines a list of BDS data convtor options.

7.50.2 Constructor & Destructor Documentation

7.50.2.1 DataFileOptions()

```
Bds::DataFileOptions::DataFileOptions (
            int options = 0 )  [inline]
```

7.50.3 Member Function Documentation

7.50.3.1 operator int()

```
Bds::DataFileOptions::operator int ( )  [inline]
```

7.50.3.2 operator"|=()

```
DataFileOptions& Bds::DataFileOptions::operator|= (
            int o )  [inline]
```

7.50.4 Member Data Documentation

7.50.4.1 oignoreBlockList

```
BArray< BUInt> Bds::DataFileOptions::oignoreBlockList
```

7.50.4.2 ooptionList

```
int Bds::DataFileOptions::ooptionList
```

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

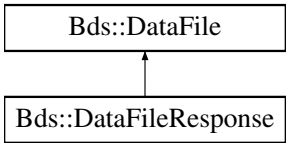
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h

7.51 Bds::DataFileResponse Class Reference

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

```
#include <BdsDataFileResponse.h>
```

Inheritance diagram for Bds::DataFileResponse:



Public Member Functions

- DataFileResponse ()
- int getFeatures ()

  Get bitmask of supported features.
- **BError** getMetaData (ChannelInfos &channelInfos)

  Return all known MetaData in the file.
- **BError** setInfo (const DataInfo &dataInfo, const ChannelInfos &channelInfos, WriteOptionsList options)

  Set information on data for write.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.51.1 Detailed Description

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

7.51.2 Constructor & Destructor Documentation

7.51.2.1 DataFileResponse()

```
Bds::DataFileResponse::DataFileResponse ( )
```

7.51.3 Member Function Documentation

7.51.3.1 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.51.3.2 getFormats()

```
DataFormat Bds::DataFileResponse::getFormats ( )  [static]
```

7.51.3.3 getMetaData()

```
BError Bds::DataFileResponse::getMetaData (
            ChannelInfos & channelInfos )  [virtual]
```

Return all known MetaData in the file.

Reimplemented from Bds::DataFile.

7.51.3.4 setInfo()

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

Set information on data for write.

Reimplemented from Bds::DataFile.

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

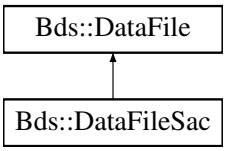
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp

7.52 Bds::DataFileSac Class Reference

Data file convertor for SAC format files.

```
#include <BdsDataFileSac.h>
```

Inheritance diagram for Bds::DataFileSac:



Public Member Functions

- DataFileSac ()
- int getFeatures ()

  Get bitmask of supported features.
- **BError** setInfo (const DataInfo &dataInfo, const ChannelInfos &channelInfos, WriteOptionsList options=WriteOptionNone)

  Set information on data for write.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.52.1 Detailed Description

Data file convertor for SAC format files.

7.52.2 Constructor & Destructor Documentation

7.52.2.1 DataFileSac()

```
Bds::DataFileSac::DataFileSac ( )
```

7.52.3 Member Function Documentation

7.52.3.1 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.52.3.2 getFormats()

```
DataFormat Bds::DataFileSac::getFormats ( )  [static]
```

7.52.3.3 setInfo()

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

Set information on data for write.

Reimplemented from Bds::DataFile.

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

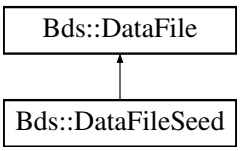
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp

7.53 Bds::DataFileSeed Class Reference

Data file convertor for SEED file formats.

```
#include <BdsDataFileSeed.h>
```

Inheritance diagram for Bds::DataFileSeed:



Public Member Functions

- DataFileSeed ()
- ~DataFileSeed ()
- **BError** close ()

  Close the file.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- int getFeatures ()

  Get bitmask of supported features.
- **BError** setFormat (**BString** format)

  Set the sub-format.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock & **data**)

  Read a block.
- **BError** setInfo (const DataInfo &dataInfo, const ChannelInfos &channelInfos, WriteOptionsList options=WriteOptionSensorData)

  Set information on data for write.
- **BError** start (**BUInt** channel, **BUInt** segment)

  Start writing next segment of data.
- **BError** writeData (const DataBlock & **data**)

  Write a block of data.
- **BError** end ()

  End write segment.
- void msrFileWrite (void * **data**, int len)

Static Public Member Functions

- static DataFormat getFormats ()

Static Public Attributes

- static **BError** omsrErr

  MSR processing error.
- static int onoLock

  Disable libmseed locking.

Additional Inherited Members

7.53.1 Detailed Description

Data file convertor for SEED file formats.

7.53.2 Constructor & Destructor Documentation

7.53.2.1 DataFileSeed()

```
Bds::DataFileSeed::DataFileSeed ( )
```

7.53.2.2 ~DataFileSeed()

```
Bds::DataFileSeed::~DataFileSeed ( )
```

7.53.3 Member Function Documentation

7.53.3.1 close()

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

Close the file.

Reimplemented from Bds::DataFile.

7.53.3.2 end()

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

End write segment.

Reimplemented from Bds::DataFile.

7.53.3.3 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.53.3.4 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.53.3.5 getFormats()

```
DataFormat Bds::DataFileSeed::getFormats ( ) [static]
```

7.53.3.6 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.53.3.7 msrFileWrite()

```
void Bds::DataFileSeed::msrFileWrite (
            void * data,
            int len )
```

7.53.3.8 readData()

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

Read a block.

Reimplemented from Bds::DataFile.

7.53.3.9 setFormat()

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

Set the sub-format.

Reimplemented from Bds::DataFile.

7.53.3.10 setInfo()

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

Set information on data for write.

Reimplemented from Bds::DataFile.

7.53.3.11 start()

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

Start writing next segment of data.

Reimplemented from Bds::DataFile.

7.53.3.12 writeData()

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

Write a block of data.

Reimplemented from Bds::DataFile.

7.53.4 Member Data Documentation

7.53.4.1 omsrErr

```
BError Bds::DataFileSeed::omsrErr  [static]
```

MSR processing error.

7.53.4.2 onoLock

```
int Bds::DataFileSeed::onoLock  [static]
```

Disable libmseed locking.

Disable libmseed lock, for sequential programs.

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

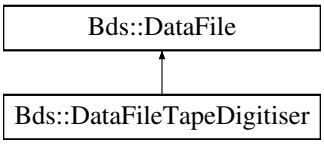
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h
- /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp

7.54 Bds::DataFileTapeDigitiser Class Reference

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

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

Inheritance diagram for Bds::DataFileTapeDigitiser:



Public Member Functions

- DataFileTapeDigitiser ()
- **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 DataFormat getFormats ()

Additional Inherited Members

7.54.1 Detailed Description

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

7.54.2 Constructor & Destructor Documentation

7.54.2.1 DataFileTapeDigitiser()

```
Bds::DataFileTapeDigitiser::DataFileTapeDigitiser ( )
```

7.54.3 Member Function Documentation

7.54.3.1 getFormats()

```
DataFormat Bds::DataFileTapeDigitiser::getFormats ( )  [static]
```

7.54.3.2 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.54.3.3 open()

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

Open the file for reading or writing.

Reimplemented from Bds::DataFile.

7.54.3.4 readData()

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

Read a block.

Reimplemented from Bds::DataFile.

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

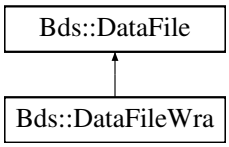
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp

7.55 Bds::DataFileWra Class Reference

Data file convertor for WRA format files.

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

Inheritance diagram for Bds::DataFileWra:



Public Member Functions

- DataFileWra ()
- **BError** setFormat (**BString** format)

  Set the sub-format.
- int getFeatures ()

  Get bitmask of supported features.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock & **data**)

  Read a block.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.55.1 Detailed Description

Data file convertor for WRA format files.

7.55.2 Constructor & Destructor Documentation

7.55.2.1 DataFileWra()

```
Bds::DataFileWra::DataFileWra ( )
```

7.55.3 Member Function Documentation

7.55.3.1 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.55.3.2 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.55.3.3 getFormats()

```
DataFormat Bds::DataFileWra::getFormats ( ) [static]
```

7.55.3.4 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.55.3.5 readData()

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

Read a block.

Reimplemented from Bds::DataFile.

7.55.3.6 setFormat()

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

Set the sub-format.

Reimplemented from Bds::DataFile.

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

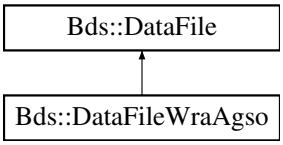
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp

7.56 Bds::DataFileWraAgso Class Reference

Data file convertor for WRA AGSO format files.

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

Inheritance diagram for Bds::DataFileWraAgso:



Public Member Functions

- DataFileWraAgso ()
- int getFeatures ()

  Get bitmask of supported features.
- DataOrder getDataOrder ()

  Get the expected order of writing data, by sample or by channel.
- **BError** getInfo (DataInfo &dataInfo, DataFileOptions options, **BList**< DataError > &errors)

  Get info on data.
- **BError** readData (**BUInt32** channel, **BUInt** segment, **BUInt32** blockNumber, DataBlock & **data**)

  Read a block.

Static Public Member Functions

- static DataFormat getFormats ()

Additional Inherited Members

7.56.1 Detailed Description

Data file convertor for WRA AGSO format files.

7.56.2 Constructor & Destructor Documentation

7.56.2.1 DataFileWraAgso()

```
Bds::DataFileWraAgso::DataFileWraAgso ( )
```

7.56.3 Member Function Documentation

7.56.3.1 getDataOrder()

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

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

Reimplemented from Bds::DataFile.

7.56.3.2 getFeatures()

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

Get bitmask of supported features.

Reimplemented from Bds::DataFile.

7.56.3.3 getFormats()

```
DataFormat Bds::DataFileWraAgso::getFormats ( ) [static]
```

7.56.3.4 getInfo()

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

Get info on data.

Reimplemented from Bds::DataFile.

7.56.3.5 readData()

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

Read a block.

Reimplemented from Bds::DataFile.

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

- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp

7.57 Bds::DataFormat Class Reference

This holds information on a seismic data format.

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

Public Member Functions

- DataFormat (**BList**< **BString** > names= **BList**< **BString** >(), **BString** description= **BString**(), **BInt32** dataRead=0, **BInt32** dataWrite=0, **BInt32** metaDataRead=0, **BInt32** metaDataWrite=0, **BString** extension= **BString**())

Public Attributes

- **BList**< **BString** > names

  The format names.
- **BString** description

  The description.
- **BInt32** dataRead

  Ability to read data.
- **BInt32** dataWrite

  Ability to write data.
- **BInt32** metaDataRead

  MetaData read supported.
- **BInt32** metaDataWrite

  MetaData write supported.
- **BString** extension

  Default filename extension.

7.57.1 Detailed Description

This holds information on a seismic data format.

It is used by the BDS data convertors to define which data formats they support..

7.57.2 Constructor & Destructor Documentation

7.57.2.1 DataFormat()

```
Bds::DataFormat::DataFormat (
            BList< BString > names = BList< BString >(),
            BString description = BString(),
            BInt32 dataRead = 0,
            BInt32 dataWrite = 0,
            BInt32 metaDataRead = 0,
            BInt32 metaDataWrite = 0,
            BString extension = BString() )
```

7.57.3 Member Data Documentation

7.57.3.1 dataRead

```
BInt32 Bds::DataFormat::dataRead
```

Ability to read data.

7.57.3.2 dataWrite

```
BInt32 Bds::DataFormat::dataWrite
```

Ability to write data.

7.57.3.3 description

```
BString Bds::DataFormat::description
```

The description.

7.57.3.4 extension

```
BString Bds::DataFormat::extension
```

Default filename extension.

7.57.3.5 metaDataRead

```
BInt32 Bds::DataFormat::metaDataRead
```

MetaData read supported.

7.57.3.6 metaDataWrite

```
BInt32 Bds::DataFormat::metaDataWrite
```

MetaData write supported.

7.57.3.7 names

```
BList< BString > Bds::DataFormat::names
```

The format names.

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

- BdsD.h
- BdsD.cc

7.58 Bds::DataFormats Class Reference

This class defines the interface for generic data file access.

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

Public Member Functions

- DataFormats ()
- ~DataFormats ()
- **BError** formatList (**BList**< DataFormat > &formats)
- **BError** formatGet (**BString** format, DataFile *&dataFile)

Protected Member Functions

- int findFormat (DataFormat dataFormat, **BString string**)

7.58.1 Detailed Description

This class defines the interface for generic data file access.

It allows programs to get a list of all of the supported data formats and to get a suitable BDS data convertor for accessing the file.

7.58.2 Constructor & Destructor Documentation

7.58.2.1 DataFormats()

```
Bds::DataFormats::DataFormats ( )
```

7.58.2.2 ~DataFormats()

```
Bds::DataFormats::~DataFormats ( )
```

7.58.3 Member Function Documentation

7.58.3.1 findFormat()

```
int Bds::DataFormats::findFormat (
            DataFormat dataFormat,
             BString string )  [protected]
```

7.58.3.2 formatGet()

```
 BError Bds::DataFormats::formatGet (
             BString format,
            DataFile *& dataFile )
```

7.58.3.3 formatList()

```
BError Bds::DataFormats::formatList (
            BList< DataFormat > & formats )
```

The documentation for this class was generated from the following files:

- /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h
- /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp

7.59 Bds::DataHandle Class Reference

This defines a handle to a sensor data stream/file when opened for read or write.

```
#include <BdsD.h>
```

Public Member Functions

- DataHandle (**BUInt32** handle=0, **BUInt32** dataFileId=0)

Public Attributes

- **BUInt32** handle

  Opaque file handle.
- **BUInt32** dataFileId

  The data file ID if opened for write.

7.59.1 Detailed Description

This defines a handle to a sensor data stream/file when opened for read or write.

7.59.2 Constructor & Destructor Documentation

7.59.2.1 DataHandle()

```
Bds::DataHandle::DataHandle (
            BUInt32 handle = 0,
            BUInt32 dataFileId = 0 )
```

7.59.3 Member Data Documentation

7.59.3.1 dataFileId

```
BUInt32 Bds::DataHandle::dataFileId
```

The data file ID if opened for write.

7.59.3.2 handle

```
BUInt32 Bds::DataHandle::handle
```

Opaque file handle.

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.60 Bds::DataInfo Class Reference

This class defines information on a set of data.

```
#include <BdsD.h>
```

Public Member Functions

- DataInfo (**BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), **BString** array= **BString**(), **BString** description= **BString**(), **BUInt32** synchronous=0, **BArray**< **BArray**< DataChannel > > channels= **BArray**< **BArray**< DataChannel > >(), **BDict**< **BString** > info= **B**↩ **Dict**< **BString** >(), **BDict**< **BString** > infoExtra= **BDict**< **BString** >(), **BList**< **BString** > warnings= **BList**< **BString** >())

Public Attributes

- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time.
- **BString** array

  The Seismic Array that all of the channels are from, if just one.
- **BString** description

  The Comment.
- **BUInt32** synchronous

  The channels are synchronously sampled.
- **BArray**< **BArray**< DataChannel > > channels

  The Data channels. Each channel can have multiple segments of data.
- **BDict**< **BString** > info

  Info on the set of channels.
- **BDict**< **BString** > infoExtra

  Extra Info on the set of channels. Used for extended error/logging information.
- **BList**< **BString** > warnings

  Warnings on the data set.

7.60.1 Detailed Description

This class defines information on a set of data.

This describes a set of seismic data. It is returns basic informatuion 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 equiring detailed information from a file it will contain an entry per contiguous data segemnt in the file.

7.60.2 Constructor & Destructor Documentation

7.60.2.1 DataInfo()

```
Bds::DataInfo::DataInfo (
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BString array = BString(),
            BString description = BString(),
            BUInt32 synchronous = 0,
            BArray< BArray< DataChannel > > channels = BArray< BArray<DataChannel > >(),
            BDict< BString > info = BDict< BString >(),
            BDict< BString > infoExtra = BDict< BString >(),
            BList< BString > warnings = BList< BString >() )
```

7.60.3 Member Data Documentation

7.60.3.1 array

```
BString Bds::DataInfo::array
```

The Seismic Array that all of the channels are from, if just one.

7.60.3.2 channels

```
BArray< BArray<DataChannel > > Bds::DataInfo::channels
```

The Data channels. Each channel can have multiple segments of data.

7.60.3.3 description

```
BString Bds::DataInfo::description
```

The Comment.

7.60.3.4 endTime

```
BTimeStamp Bds::DataInfo::endTime
```

The End Time.

7.60.3.5 info

```
BDict< BString > Bds::DataInfo::info
```

Info on the set of channels.

7.60.3.6 infoExtra

```
BDict< BString > Bds::DataInfo::infoExtra
```

Extra Info on the set of channels. Used for extended error/logging information.

7.60.3.7 startTime

```
BTimeStamp Bds::DataInfo::startTime
```

The Start Time.

7.60.3.8 synchronous

```
BUInt32 Bds::DataInfo::synchronous
```

The channels are synchronously sampled.

7.60.3.9 warnings

```
BList< BString > Bds::DataInfo::warnings
```

Warnings on the data set.

The documentation for this class was generated from the following files:

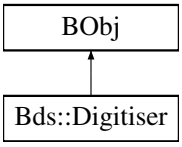
- BdsD.h
- BdsD.cc

7.61 Bds::Digitiser Class Reference

This class defines a seismic Digitiser.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::Digitiser:



Public Member Functions

- Digitiser (**BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTime↩Stamp**(), **BString** name= **BString**(), **BString** type= **BString**(), **BString** serialNumber= **BString**(), **B↩UInt32** numberChannels=0, **BFloat64** baseSamplingFrequency=0, **BFloat64** initialSamplingFrequency=0, **BFloat64** gain=0, **BInt32** shared=0)
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  The ID.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time the channel was available.
- **BString** name

  The Digitisers name.
- **BString** type

  The Digitisers type.

- **BString** serialNumber

  The digitisers's serial number.
- **BUInt32** numberChannels

  The number of supported channels.
- **BFloat64** baseSamplingFrequency

  The base sampling frequency.
- **BFloat64** initialSamplingFrequency

  The initial pre-decimation sampling frequency.
- **BFloat64** gain

  The overall gain of the digitiser at the manufacturers calibration frequency. (For information only)
- **BInt32** shared

  This digitiser is shared.

7.61.1 Detailed Description

This class defines a seismic Digitiser.

This just stores information on the seismic instrument's digitiser. Its contents is generally for information only.

7.61.2 Constructor & Destructor Documentation

7.61.2.1 Digitiser()

```
Bds::Digitiser::Digitiser (
            BUInt32 id = 0,
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BString name = BString(),
            BString type = BString(),
            BString serialNumber = BString(),
            BUInt32 numberChannels = 0,
            BFloat64 baseSamplingFrequency = 0,
            BFloat64 initialSamplingFrequency = 0,
            BFloat64 gain = 0,
            BInt32 shared = 0 )
```

7.61.3 Member Function Documentation

7.61.3.1 getMember()

```
BError Bds::Digitiser::getMember (
            BString name,
            BString & value ) [virtual]
```

Reimplemented from **BObj**.

7.61.3.2 getMembers()

```
BError Bds::Digitiser::getMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.61.3.3 getType()

```
BString Bds::Digitiser::getType ( )
```

7.61.3.4 setMember()

```
BError Bds::Digitiser::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.61.3.5 setMembers()

```
BError Bds::Digitiser::setMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.61.4 Member Data Documentation

7.61.4.1 baseSamplingFrequency

```
BFloat64 Bds::Digitiser::baseSamplingFrequency
```

The base sampling frequency.

7.61.4.2 endTime

```
BTimeStamp Bds::Digitiser::endTime
```

The End Time the channel was available.

7.61.4.3 gain

```
BFloat64 Bds::Digitiser::gain
```

The overall gain of the digitiser at the manufacturers calibration frequency. (For information only)

7.61.4.4 id

```
BUInt32 Bds::Digitiser::id
```

The ID.

7.61.4.5 initialSamplingFrequency

```
BFloat64 Bds::Digitiser::initialSamplingFrequency
```

The initial pre-decimation sampling frequency.

7.61.4.6 name

```
BString Bds::Digitiser::name
```

The Digitisers name.

7.61.4.7 numberChannels

```
BUInt32 Bds::Digitiser::numberChannels
```

The number of supported channels.

7.61.4.8 serialNumber

```
BString Bds::Digitiser::serialNumber
```

The digitisers's serial number.

7.61.4.9 shared

```
BInt32 Bds::Digitiser::shared
```

This digitiser is shared.

7.61.4.10 startTime

```
BTimeStamp Bds::Digitiser::startTime
```

The Start Time.

7.61.4.11 type

```
BString Bds::Digitiser::type
```

The Digitisers type.

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.62 Bds::Fap Class Reference

This class defines an entry in an Amplitude/Phase Response table.

```
#include <BdsD.h>
```

Public Member Functions

- Fap (**BFloat64** frequency=0, **BFloat64** amplitude=0, **BFloat64** phase=0)

Public Attributes

- **BFloat64** frequency

  The frequency.
- **BFloat64** amplitude

  The Amplitude.
- **BFloat64** phase

  The Phase.

7.62.1 Detailed Description

This class defines an entry in an Amplitude/Phase Response table.

7.62.2 Constructor & Destructor Documentation

7.62.2.1 Fap()

```
Bds::Fap::Fap (
            BFloat64 frequency = 0,
            BFloat64 amplitude = 0,
            BFloat64 phase = 0 )
```

7.62.3 Member Data Documentation

7.62.3.1 amplitude

```
BFloat64 Bds::Fap::amplitude
```

The Amplitude.

7.62.3.2 frequency

```
BFloat64 Bds::Fap::frequency
```

The frequency.

7.62.3.3 phase

```
BFloat64 Bds::Fap::phase
```

The Phase.

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.63 Bds::Fir Class Reference

This class defines an FIR response table.

```
#include <BdsD.h>
```

Public Member Functions

- Fir (**BArray**< FirEntry > b= **BArray**< FirEntry >(), **BArray**< FirEntry > a= **BArray**< FirEntry >())

Public Attributes

- **BArray**< FirEntry > b

  Numerator.
- **BArray**< FirEntry > a

  Denominator.

7.63.1 Detailed Description

This class defines an FIR response table.

This has an array of the A and B coefficients.

7.63.2 Constructor & Destructor Documentation

7.63.2.1 Fir()

```
Bds::Fir::Fir (
            BArray< FirEntry > b =  BArray<FirEntry >(),
            BArray< FirEntry > a =  BArray<FirEntry >() )
```

7.63.3 Member Data Documentation

7.63.3.1 a

```
BArray<FirEntry > Bds::Fir::a
```

Denominator.

7.63.3.2 b

```
BArray<FirEntry > Bds::Fir::b
```

Numerator.

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.64 Bds::FirEntry Class Reference

This class defines an entry in a FIR coefficient table.

```
#include <BdsD.h>
```

Public Member Functions

- FirEntry (**BFloat64** coefficient=0, **BFloat64** error=0)

Public Attributes

- **BFloat64** coefficient

  Value.
- **BFloat64** error

  Error.

7.64.1 Detailed Description

This class defines an entry in a FIR coefficient table.

7.64.2 Constructor & Destructor Documentation

7.64.2.1 FirEntry()

```
Bds::FirEntry::FirEntry (
            BFloat64 coefficient = 0,
            BFloat64 error = 0 )
```

7.64.3 Member Data Documentation

7.64.3.1 coefficient

```
BFloat64 Bds::FirEntry::coefficient
```

Value.

7.64.3.2 error

```
BFloat64 Bds::FirEntry::error
```

Error.

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.65 Bds::GcfChannel Struct Reference

DataFileGcf internal GCF channel information.

```
#include <BdsDataFileGcf.h>
```

Public Attributes

- **BString** systemId
- **BString** streamId
- **BUInt** type
- **BUInt** sampleRate
- **BUInt** format
- **BUInt32** channel

7.65.1 Detailed Description

DataFileGcf internal GCF channel information.

7.65.2 Member Data Documentation

7.65.2.1 channel

```
BUInt32 Bds::GcfChannel::channel
```

7.65.2.2 format

```
BUInt Bds::GcfChannel::format
```

7.65.2.3 sampleRate

```
BUInt Bds::GcfChannel::sampleRate
```

7.65.2.4 streamId

```
BString Bds::GcfChannel::streamId
```

7.65.2.5 systemId

```
BString Bds::GcfChannel::systemId
```

7.65.2.6 type

```
BUInt Bds::GcfChannel::type
```

The documentation for this struct was generated from the following file:

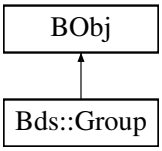
- /src/blacknest/bds/bds/bdsDataLib/BdsDataFileGcf.h

7.66 Bds::Group Class Reference

This holds information on a user security group.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::Group:



Public Member Functions

- Group (**BUInt32** id=0, **BString** group= **BString**(), **BString** description= **BString**())
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  The unique id.
- **BString** group

  The Group name.
- **BString** description

  The Groups description.

7.66.1 Detailed Description

This holds information on a user security group.

The BDS has the concept of a security group that users can belong to. This class defines that security group as stored in the database.

7.66.2 Constructor & Destructor Documentation

7.66.2.1 Group()

```
Bds::Group::Group (
            BUInt32 id = 0,
            BString group = BString(),
            BString description = BString() )
```

7.66.3 Member Function Documentation

7.66.3.1 getMember()

```
BError Bds::Group::getMember (
            BString name,
            BString & value )  [virtual]
```

Reimplemented from **BObj**.

7.66.3.2 getMembers()

```
BError Bds::Group::getMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.66.3.3 getType()

```
BString Bds::Group::getType ( )
```

7.66.3.4 setMember()

```
BError Bds::Group::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.66.3.5 setMembers()

```
BError Bds::Group::setMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.66.4 Member Data Documentation

7.66.4.1 description

```
BString Bds::Group::description
```

The Groups description.

7.66.4.2 group

```
BString Bds::Group::group
```

The Group name.

7.66.4.3 id

```
BUInt32 Bds::Group::id
```

The unique id.

The documentation for this class was generated from the following files:

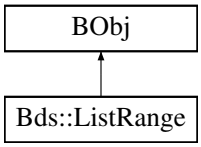
- BdsD.h
- BdsD.cc

7.67 Bds::ListRange Class Reference

This class defines an integer based range.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::ListRange:



Public Member Functions

- ListRange (**BUInt32** start=0, **BUInt32** number=0, **BInt32** reverse=0)
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** start

  The start position.
- **BUInt32** number

  The number of items.
- **BInt32** reverse

  List from end.

7.67.1 Detailed Description

This class defines an integer based range.

It is used for limit the number of items returned in selections etc.

7.67.2 Constructor & Destructor Documentation

7.67.2.1 ListRange()

```
Bds::ListRange::ListRange (
            BUInt32 start = 0,
            BUInt32 number = 0,
            BInt32 reverse = 0 )
```

7.67.3 Member Function Documentation

7.67.3.1 getMember()

```
BError Bds::ListRange::getMember (
            BString name,
            BString & value )  [virtual]
```

Reimplemented from **BObj**.

7.67.3.2 getMembers()

```
BError Bds::ListRange::getMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.67.3.3 getType()

```
BString Bds::ListRange::getType ( )
```

7.67.3.4 setMember()

```
BError Bds::ListRange::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.67.3.5 setMembers()

```
BError Bds::ListRange::setMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.67.4 Member Data Documentation

7.67.4.1 number

```
BUInt32 Bds::ListRange::number
```

The number of items.

7.67.4.2 reverse

```
BInt32 Bds::ListRange::reverse
```

List from end.

7.67.4.3 start

```
BUInt32 Bds::ListRange::start
```

The start position.

The documentation for this class was generated from the following files:

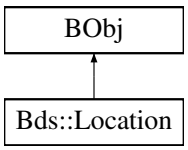
- BdsD.h
- BdsD.cc

7.68 Bds::Location Class Reference

This class defines the physical location of a Station.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::Location:



Public Member Functions

- Location (**BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTime↩ Stamp**(), **BString** network= **BString**(), **BString** station= **BString**(), **BString** datum= **BString**(), **B↩ Float64** longitude=0, **BFloat64** latitude=0, **BFloat64** elevation=0, **BFloat64** arrayOffsetEast=0, **BFloat64** arrayOffsetNorth=0)
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  The ID.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time the channel was available.
- **BString** network

  The Network/Organisation Name.
- **BString** station

  The station this location is for.
- **BString** datum

  The locations Datum.
- **BFloat64** longitude

  The longitude in degrees using the WGS84 datum.
- **BFloat64** latitude

  The Latitude in degrees using the WGS84 datum.
- **BFloat64** elevation

  The ground level elevation in meters from the WGS84 ellipsoid (Sea level)
- **BFloat64** arrayOffsetEast

  The Array offset in in an array in an easterly direction.
- **BFloat64** arrayOffsetNorth

  The Array offset in in an array in a northerly direction.

7.68.1 Detailed Description

This class defines the physical location of a Station.

This defines the physical location of the station using WGS84 longitude and latitude parameters. It also defines the stations elevation and if part of a seismic array, the offset with respect to the arrays centre location.

7.68.2 Constructor & Destructor Documentation

7.68.2.1 Location()

```
Bds::Location::Location (
            BUInt32 id = 0,
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BString network = BString(),
            BString station = BString(),
            BString datum = BString(),
            BFloat64 longitude = 0,
            BFloat64 latitude = 0,
            BFloat64 elevation = 0,
            BFloat64 arrayOffsetEast = 0,
            BFloat64 arrayOffsetNorth = 0 )
```

7.68.3 Member Function Documentation

7.68.3.1 getMember()

```
BError Bds::Location::getMember (
            BString name,
            BString & value )  [virtual]
```

Reimplemented from **BObj**.

7.68.3.2 getMembers()

```
BError Bds::Location::getMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.68.3.3 getType()

```
BString Bds::Location::getType ( )
```

7.68.3.4 setMember()

```
BError Bds::Location::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.68.3.5 setMembers()

```
BError Bds::Location::setMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.68.4 Member Data Documentation

7.68.4.1 arrayOffsetEast

```
BFloat64 Bds::Location::arrayOffsetEast
```

The Array offset in in an array in an easterly direction.

7.68.4.2 arrayOffsetNorth

```
BFloat64 Bds::Location::arrayOffsetNorth
```

The Array offset in in an array in a northerly direction.

7.68.4.3 datum

```
BString Bds::Location::datum
```

The locations Datum.

7.68.4.4 elevation

```
BFloat64 Bds::Location::elevation
```

The ground level elevation in meters from the WGS84 ellipsoid (Sea level)

7.68.4.5 endTime

```
BTimeStamp Bds::Location::endTime
```

The End Time the channel was available.

7.68.4.6 id

```
BUInt32 Bds::Location::id
```

The ID.

7.68.4.7 latitude

```
BFloat64 Bds::Location::latitude
```

The Latitude in degrees using the WGS84 datum.

7.68.4.8 longitude

```
BFloat64 Bds::Location::longitude
```

The longitude in degrees using the WGS84 datum.

7.68.4.9 network

```
BString Bds::Location::network
```

The Network/Organisation Name.

7.68.4.10 startTime

```
BTimeStamp Bds::Location::startTime
```

The Start Time.

7.68.4.11 station

```
BString Bds::Location::station
```

The station this location is for.

The documentation for this class was generated from the following files:

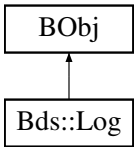
- BdsD.h
- BdsD.cc

7.69 Bds::Log Class Reference

This holds information on a Log entry.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::Log:



Public Member Functions

- Log (**BUInt32** id=0, **BTimeStamp** time= **BTimeStamp**(), **BString** type= **BString**(), **BUInt32** priority=0, **BString** subSystem= **BString**(), **BString** title= **BString**(), **BString** description= **BString**())
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  The unique id.
- **BTimeStamp** time

  The Time.
- **BString** type

  The Type.
- **BUInt32** priority

  The priority 0 to 5.
- **BString** subSystem

  The SubSystem.
- **BString** title

  The Changes title.
- **BString** description

  The Description of the change.

7.69.1 Detailed Description

This holds information on a Log entry.

Log entries are added automatically and manually to the system. A system administartor can view these logs.

7.69.2 Constructor & Destructor Documentation

7.69.2.1 Log()

```
Bds::Log::Log (
            BUInt32 id = 0,
            BTimeStamp time = BTimeStamp(),
            BString type = BString(),
            BUInt32 priority = 0,
            BString subSystem = BString(),
            BString title = BString(),
            BString description = BString() )
```

7.69.3 Member Function Documentation

7.69.3.1 getMember()

```
BError Bds::Log::getMember (
            BString name,
            BString & value ) [virtual]
```

Reimplemented from **BObj**.

7.69.3.2 getMembers()

```
BError Bds::Log::getMembers (
            BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

7.69.3.3 getType()

```
BString Bds::Log::getType ( )
```

7.69.3.4 setMember()

```
BError Bds::Log::setMember (
            BString name,
            BString value ) [virtual]
```

Reimplemented from **BObj**.

7.69.3.5 setMembers()

```
BError Bds::Log::setMembers (
            BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

7.69.4 Member Data Documentation

7.69.4.1 description

```
BString Bds::Log::description
```

The Description of the change.

7.69.4.2 id

```
BUInt32 Bds::Log::id
```

The unique id.

7.69.4.3 priority

```
BUInt32 Bds::Log::priority
```

The priority 0 to 5.

7.69.4.4 subSystem

```
BString Bds::Log::subSystem
```

The SubSystem.

7.69.4.5 time

```
BTimeStamp Bds::Log::time
```

The Time.

7.69.4.6 title

```
BString Bds::Log::title
```

The Changes title.

7.69.4.7 type

```
BString Bds::Log::type
```

The Type.

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.70 Bds::LogSelect Class Reference

This defines the selection cirteria when requesting a set of log entries.

```
#include <BdsD.h>
```

Public Member Functions

- LogSelect (**BTimeStamp** startTime= **BTimeStamp**(), **BString** type= **BString**(), **BUInt32** priority=0, **B**↩ **String** subSystem= **BString**())

Public Attributes

- **BTimeStamp** startTime

  The start time.
- **BString** type

  The Type.
- **BUInt32** priority

  The priority 0 to 5.
- **BString** subSystem

  The SubSystem.

7.70.1 Detailed Description

This defines the selection cirteria when requesting a set of log entries.

7.70.2 Constructor & Destructor Documentation

7.70.2.1 LogSelect()

```
Bds::LogSelect::LogSelect (
            BTimeStamp startTime = BTimeStamp(),
            BString type = BString(),
            BUInt32 priority = 0,
            BString subSystem = BString() )
```

7.70.3 Member Data Documentation

7.70.3.1 priority

```
BUInt32 Bds::LogSelect::priority
```

The priority 0 to 5.

7.70.3.2 startTime

```
BTimeStamp Bds::LogSelect::startTime
```

The start time.

7.70.3.3 subSystem

```
BString Bds::LogSelect::subSystem
```

The SubSystem.

7.70.3.4 type

```
BString Bds::LogSelect::type
```

The Type.

The documentation for this class was generated from the following files:

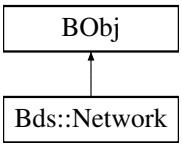
- BdsD.h
- BdsD.cc

7.71 Bds::Network Class Reference

This class defines a seismic Network organisation.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::Network:



Public Member Functions

- Network (**BUInt32** id=0, **BString** network= **BString**(), **BString** description= **BString**(), **BList**< **BString** > stations= **BList**< **BString** >())
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  Unique ID when stored in a database or for other uses.
- **BString** network

  The name.
- **BString** description

  The organisations description.
- **BList**< **BString** > stations

  The list of arrays/stations the Network uses.

7.71.1 Detailed Description

This class defines a seismic Network organisation.

Typical Seisomic Networks are "BN", IDC" etc.

7.71.2 Constructor & Destructor Documentation

7.71.2.1 Network()

```
Bds::Network::Network (
            BUInt32 id = 0,
            BString network = BString(),
            BString description = BString(),
            BList< BString > stations = BList< BString >() )
```

7.71.3 Member Function Documentation

7.71.3.1 getMember()

```
BError Bds::Network::getMember (
            BString name,
            BString & value ) [virtual]
```

Reimplemented from **BObj**.

7.71.3.2 getMembers()

```
BError Bds::Network::getMembers (
            BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

7.71.3.3 getType()

```
BString Bds::Network::getType ( )
```

7.71.3.4 setMember()

```
BError Bds::Network::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.71.3.5 setMembers()

```
BError Bds::Network::setMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.71.4 Member Data Documentation

7.71.4.1 description

```
BString Bds::Network::description
```

The organisations description.

7.71.4.2 id

```
BUInt32 Bds::Network::id
```

Unique ID when stored in a database or for other uses.

7.71.4.3 network

```
BString Bds::Network::network
```

The name.

7.71.4.4 stations

```
BList< BString > Bds::Network::stations
```

The list of arrays/stations the Network uses.

The documentation for this class was generated from the following files:

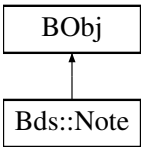
- BdsD.h
- BdsD.cc

7.72 Bds::Note Class Reference

This holds information on a Note for general information.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::Note:



Public Member Functions

- Note (**BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), **BString** network= **BString**(), **BString** station= **BString**(), **BString** channel= **BString**(), **BString** source= **BString**(), **BString** type= **BString**(), **BString** user= **BString**(), **BTimeStamp** timeAdded= **BTimeStamp**(), **BInt32** errorNumber=0, **BString** title= **BString**(), **BString** description= **BString**(), **BString** docFormat= **BString**(), **BString** docUrl= **BString**(), **BUInt32** dataFileId=0, **BString** importFilename= **BString**())
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  The unique id.
- **BTimeStamp** startTime

  The Start Time note is for.
- **BTimeStamp** endTime

  The End Time note is for.
- **BString** network

  The Network Name.
- **BString** station

The Station/Array name.

- **BString** channel

  The Channels name.
- **BString** source

  The Data Source.
- **BString** type

  The Type (note, warning, error ...)
- **BString** user

  The user.
- **BTimeStamp** timeAdded

  The Time Entered.
- **BInt32** errorNumber

  Error number if error.
- **BString** title

  The title.
- **BString** description

  The Description.
- **BString** docFormat

  Document format if any.
- **BString** docUrl

  Document Url if any.
- **BUInt32** dataFileId

  The data file id associated with this note.
- **BString** importFilename

  The import filename.

7.72.1 Detailed Description

This holds information on a Note for general information.

Normally a Note can be added for a particular set of data over a particular time period. These notes are sometimes added automaically during a data import process or by a user. A data user can then ask for any notes for a particular set of data.

7.72.2 Constructor & Destructor Documentation

7.72.2.1 Note()

```
Bds::Note::Note (
            BUInt32 id = 0,
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BString network = BString(),
            BString station = BString(),
            BString channel = BString(),
            BString source = BString(),
            BString type = BString(),
```

```
            BString user = BString(),
            BTimeStamp timeAdded = BTimeStamp(),
            BInt32 errorNumber = 0,
            BString title = BString(),
            BString description = BString(),
            BString docFormat = BString(),
            BString docUrl = BString(),
            BUInt32 dataFileId = 0,
            BString importFilename = BString() )
```

7.72.3 Member Function Documentation

7.72.3.1 getMember()

```
BError Bds::Note::getMember (
            BString name,
            BString & value )  [virtual]
```

Reimplemented from **BObj**.

7.72.3.2 getMembers()

```
BError Bds::Note::getMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.72.3.3 getType()

```
BString Bds::Note::getType ( )
```

7.72.3.4 setMember()

```
BError Bds::Note::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.72.3.5 setMembers()

```
BError Bds::Note::setMembers (
            BDictString & members ) [virtual]
```

Reimplemented from **BObj**.

7.72.4 Member Data Documentation

7.72.4.1 channel

```
BString Bds::Note::channel
```

The Channels name.

7.72.4.2 dataFileId

```
BUInt32 Bds::Note::dataFileId
```

The data file id associated with this note.

7.72.4.3 description

```
BString Bds::Note::description
```

The Description.

7.72.4.4 docFormat

```
BString Bds::Note::docFormat
```

Document format if any.

7.72.4.5 docUrl

```
BString Bds::Note::docUrl
```

Document Url if any.

7.72.4.6 endTime

```
BTimeStamp Bds::Note::endTime
```

The End Time note is for.

7.72.4.7 errorNumber

```
BInt32 Bds::Note::errorNumber
```

Error number if error.

7.72.4.8 id

```
BUInt32 Bds::Note::id
```

The unique id.

7.72.4.9 importFilename

```
BString Bds::Note::importFilename
```

The import filename.

7.72.4.10 network

```
BString Bds::Note::network
```

The Network Name.

7.72.4.11 source

```
BString Bds::Note::source
```

The Data Source.

7.72.4.12 startTime

```
BTimeStamp Bds::Note::startTime
```

The Start Time note is for.

7.72.4.13 station

```
BString Bds::Note::station
```

The Station/Array name.

7.72.4.14 timeAdded

```
BTimeStamp Bds::Note::timeAdded
```

The Time Entered.

7.72.4.15 title

```
BString Bds::Note::title
```

The title.

7.72.4.16 type

```
BString Bds::Note::type
```

The Type (note, warning, error ...)

7.72.4.17 user

```
BString Bds::Note::user
```

The user.

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.73 Bds::Point Class Reference

This class defines an X,Y location.

```
#include <BdsD.h>
```

Public Member Functions

- Point (**BFloat64** x=0, **BFloat64** y=0)

Public Attributes

- **BFloat64** x
- **BFloat64** y

7.73.1 Detailed Description

This class defines an X,Y location.

The class simply stores the x and y point values.

7.73.2 Constructor & Destructor Documentation

7.73.2.1 Point()

```
Bds::Point::Point (
            BFloat64 x = 0,
            BFloat64 y = 0 )
```

7.73.3 Member Data Documentation

7.73.3.1 x

```
BFloat64 Bds::Point::x
```

7.73.3.2 y

```
BFloat64 Bds::Point::y
```

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.74 Bds::PoleZero Class Reference

This class defines a Pole/Zero Response.

```
#include <BdsD.h>
```

Public Member Functions

- PoleZero (**BArray**< **BComplex** > poles= **BArray**< **BComplex** >(), **BArray**< **BComplex** > zeros= **BArray**< **BComplex** >())

Public Attributes

- **BArray**< **BComplex** > poles

  Poles.
- **BArray**< **BComplex** > zeros

  Zeros.

7.74.1 Detailed Description

This class defines a Pole/Zero Response.

It consits of an array of Complex poles and an array of Complex zeros.

7.74.2 Constructor & Destructor Documentation

7.74.2.1 PoleZero()

```
Bds::PoleZero::PoleZero (
            BArray< BComplex > poles = BArray< BComplex >(),
            BArray< BComplex > zeros = BArray< BComplex >() )
```

7.74.3 Member Data Documentation

7.74.3.1 poles

```
BArray< BComplex > Bds::PoleZero::poles
```

Poles.

7.74.3.2 zeros

```
BArray< BComplex > Bds::PoleZero::zeros
```

Zeros.

The documentation for this class was generated from the following files:

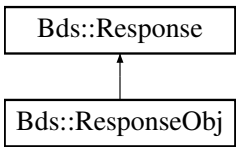
- BdsD.h
- BdsD.cc

7.75 Bds::Response Class Reference

This class defines a seismic Response characteristic.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::Response:



Public Member Functions

- Response (**BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTime**↩ **Stamp**(), **BString** network= **BString**(), **BString** station= **BString**(), **BString** channel= **BString**(), **B**↩ **String** source= **BString**(), **BUInt32** stage=0, **BString** name= **BString**(), **BString** type= **BString**(), PoleZero poleZeros=PoleZero(), **BArray**< Fap > faps= **BArray**< Fap >(), Fir fir=Fir(), **BFloat64** gain=0, **BFloat64** gainFrequency=0, **BString** stageType= **BString**(), **BFloat64** decimation=0, **BString** symmetry= **BString**(), **BString** description= **BString**(), **BInt32** measured=0, **BFloat64** sampleRate=0)

Public Attributes

- **BUInt32** id

  The ID.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time the channel was available.
- **BString** network

  The Network/Organisation Name.
- **BString** station

  The station.
- **BString** channel

  The channel.
- **BString** source

  The source.
- **BUInt32** stage

  The stage (1, 2, 3, ...)
- **BString** name

  The response name. (Anti-Aliasing filter, Digitiser, post filter etc)
- **BString** type

  The type of response (PoleZero,AmplitudePhase or FIR Coefficients)
- PoleZero poleZeros

  PoleZero, AmplitudePhase or FIR Coefficient data.
- **BArray**< Fap > faps

  The Frequency/Amplitude/Phase table.
- Fir fir

  The FIR filters coefficients.
- **BFloat64** gain

  Overall gain at gain frequency. (For information)
- **BFloat64** gainFrequency

  Frequency that gain is valid for. (For information)
- **BString** stageType

  The stage type: A - Analog (rad/sec), B - Analog (Hz), C - Composite, D - Digital.
- **BFloat64** decimation

  Decimation performed post filter.
- **BString** symmetry

  Symmetry for FIR coefficients (A = asymmetric, B = symmetric[odd], C = symmetric[even]) ??
- **BString** description

  Misc description.
- **BInt32** measured

  If response was a measured response.
- **BFloat64** sampleRate

  The stage's sample rate.

7.75.1 Detailed Description

This class defines a seismic Response characteristic.

For each seismic channel there is a frequency response characteristic. There can be multiple stages in a channels frequency response, this response data describes one of those stages frequencies responses. The stage parameter defines which stage it is for (1, 2, 3, ...) Stage 1 is reserved to store an overall channel response. A response can be in the form of an array of poles and zeros, a FAP array, or a set of FIR coefficients. This object contains members for other responce characteristics as defined in various seismic response databases.

7.75.2 Constructor & Destructor Documentation

7.75.2.1 Response()

```
Bds::Response::Response (
            BUInt32 id = 0,
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BString network = BString(),
            BString station = BString(),
            BString channel = BString(),
            BString source = BString(),
            BUInt32 stage = 0,
            BString name = BString(),
            BString type = BString(),
           PoleZero poleZeros = PoleZero(),
            BArray< Fap > faps = BArray<Fap >(),
           Fir fir = Fir(),
            BFloat64 gain = 0,
            BFloat64 gainFrequency = 0,
            BString stageType = BString(),
            BFloat64 decimation = 0,
            BString symmetry = BString(),
            BString description = BString(),
            BInt32 measured = 0,
            BFloat64 sampleRate = 0 )
```

7.75.3 Member Data Documentation

7.75.3.1 channel

```
BString Bds::Response::channel
```

The channel.

7.75.3.2 decimation

```
BFloat64 Bds::Response::decimation
```

Decimation performed post filter.

7.75.3.3 description

```
BString Bds::Response::description
```

Misc description.

7.75.3.4 endTime

```
BTimeStamp Bds::Response::endTime
```

The End Time the channel was available.

7.75.3.5 faps

```
BArray<Fap > Bds::Response::faps
```

The Frequency/Amplitude/Phase table.

7.75.3.6 fir

```
Fir Bds::Response::fir
```

The FIR filters coefficients.

7.75.3.7 gain

```
BFloat64 Bds::Response::gain
```

Overall gain at gain frequency. (For information)

7.75.3.8 gainFrequency

```
BFloat64 Bds::Response::gainFrequency
```

Frequency that gain is valid for. (For information)

7.75.3.9 id

```
BUInt32 Bds::Response::id
```

The ID.

7.75.3.10 measured

```
BInt32 Bds::Response::measured
```

If response was a measured response.

7.75.3.11 name

```
BString Bds::Response::name
```

The response name. (Anti-Aliasing filter, Digitiser, post filter etc)

7.75.3.12 network

```
BString Bds::Response::network
```

The Network/Organisation Name.

7.75.3.13 poleZeros

```
PoleZero Bds::Response::poleZeros
```

PoleZero, AmplitudePhase or FIR Coefficient data.

7.75.3.14 sampleRate

```
BFloat64 Bds::Response::sampleRate
```

The stage's sample rate.

7.75.3.15 source

```
BString Bds::Response::source
```

The source.

7.75.3.16 stage

```
BUInt32 Bds::Response::stage
```

The stage (1, 2, 3, ...)

7.75.3.17 stageType

```
BString Bds::Response::stageType
```

The stage type: A - Analog (rad/sec), B - Analog (Hz), C - Composite, D - Digital.

7.75.3.18 startTime

```
BTimeStamp Bds::Response::startTime
```

The Start Time.

7.75.3.19 station

```
BString Bds::Response::station
```

The station.

7.75.3.20 symmetry

```
BString Bds::Response::symmetry
```

Symmetry for FIR coefficients (A = asymmetric, B = symmetric[odd], C = symmetric[even]) ??

7.75.3.21 type

```
BString Bds::Response::type
```

The type of response (PoleZero,AmplitudePhase or FIR Coefficients)

The documentation for this class was generated from the following files:

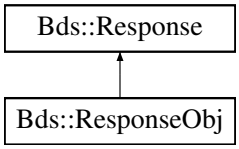
- BdsD.h
- BdsD.cc

7.76 Bds::ResponseObj Class Reference

Response object adding string conversion.

```
#include <BdsLib.h>
```

Inheritance diagram for Bds::ResponseObj:



Public Member Functions

- ResponseObj (const Response &response)
- ~ResponseObj ()
- **BString** getString ()
- void setString (**BString** str)

Additional Inherited Members

7.76.1 Detailed Description

Response object adding string conversion.

7.76.2 Constructor & Destructor Documentation

7.76.2.1 ResponseObj()

```
Bds::ResponseObj::ResponseObj (
            const Response & response )
```

7.76.2.2 ~ResponseObj()

```
Bds::ResponseObj::~ResponseObj ( )
```

7.76.3 Member Function Documentation

7.76.3.1 getString()

```
BString Bds::ResponseObj::getString ( )
```

7.76.3.2 setString()

```
void Bds::ResponseObj::setString (
            BString str )
```

The documentation for this class was generated from the following files:

- BdsLib.h
- BdsLib.cpp

7.77 Bds::Selection Class Reference

This class defines a generic metadata or seismic data selection.

```
#include <BdsD.h>
```

Public Member Functions

- Selection (**BUInt32** id=0, ListRange range=ListRange(), **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), **BList**< SelectionChannel > channels= **BList**< SelectionChannel >(), **BUInt32** channelId=0, **BUInt32** digitiserId=0, **BUInt32** sensorId=0, **BUInt32** sensorOldId=0, **BInt32** completeSegments=0, **BString** calibrationName= **BString**())

Public Attributes

- **BUInt32** id

  The ID of the record to return.
- ListRange range

  The range of data to return.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time.
- **BList**< SelectionChannel > channels

  The data channels to select.
- **BUInt32** channelId

  The Channel id.
- **BUInt32** digitiserId

  The Digitiser id.
- **BUInt32** sensorId

  The Sensor id.
- **BUInt32** sensorOldId

  The Sensor old id.
- **BInt32** completeSegments

  Do not clip the segment times to match the required time period.
- **BString** calibrationName

  Calibration name to use.

7.77.1 Detailed Description

This class defines a generic metadata or seismic data selection.

This defines a set of selection criteria when selecting items from the BDS metadata or seismic data sets. The fields, when set, limit the ietms returned by the settings provided. In effect it has an "AND" type of function with the parameters provided. There are some specific parameters like: channelId, digitiserId, sensorId etc which are used with specific data selection functions.

7.77.2 Constructor & Destructor Documentation

7.77.2.1 Selection()

```
Bds::Selection::Selection (
            BUInt32 id = 0,
           ListRange range = ListRange(),
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BList< SelectionChannel > channels = BList<SelectionChannel >(),
            BUInt32 channelId = 0,
            BUInt32 digitiserId = 0,
            BUInt32 sensorId = 0,
            BUInt32 sensorOldId = 0,
            BInt32 completeSegments = 0,
            BString calibrationName = BString() )
```

7.77.3 Member Data Documentation

7.77.3.1 calibrationName

```
BString Bds::Selection::calibrationName
```

Calibration name to use.

7.77.3.2 channelId

```
BUInt32 Bds::Selection::channelId
```

The Channel id.

7.77.3.3 channels

```
BList<SelectionChannel > Bds::Selection::channels
```

The data channels to select.

7.77.3.4 completeSegments

```
BInt32 Bds::Selection::completeSegments
```

Do not clip the segment times to match the required time period.

7.77.3.5 digitiserId

```
BUInt32 Bds::Selection::digitiserId
```

The Digitiser id.

7.77.3.6 endTime

```
BTimeStamp Bds::Selection::endTime
```

The End Time.

7.77.3.7 id

```
BUInt32 Bds::Selection::id
```

The ID of the record to return.

7.77.3.8 range

```
ListRange Bds::Selection::range
```

The range of data to return.

7.77.3.9 sensorId

```
BUInt32 Bds::Selection::sensorId
```

The Sensor id.

7.77.3.10 sensorOldId

```
BUInt32 Bds::Selection::sensorOldId
```

The Sensor old id.

7.77.3.11 startTime

```
BTimeStamp Bds::Selection::startTime
```

The Start Time.

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.78 Bds::SelectionChannel Class Reference

This class defines a channel for selection.

```
#include <BdsD.h>
```

Public Member Functions

- SelectionChannel (**BString** network= **BString**(), **BString** station= **BString**(), **BString** channel= **BString**(), **BString** source= **BString**())

Public Attributes

- **BString** network
- **BString** station
- **BString** channel
- **BString** source

7.78.1 Detailed Description

This class defines a channel for selection.

It contains the network:station:channel:source names.

7.78.2 Constructor & Destructor Documentation

7.78.2.1 SelectionChannel()

```
Bds::SelectionChannel::SelectionChannel (
            BString network =  BString(),
            BString station =  BString(),
            BString channel =  BString(),
            BString source =  BString() )
```

7.78.3 Member Data Documentation

7.78.3.1 channel

```
BString Bds::SelectionChannel::channel
```

7.78.3.2 network

```
BString Bds::SelectionChannel::network
```

7.78.3.3 source

```
BString Bds::SelectionChannel::source
```

7.78.3.4 station

```
BString Bds::SelectionChannel::station
```

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.79 Bds::SelectionInfo Class Reference

This class defines the set of metadata or siesmic data selected when getSelectionInfo() is use.

```
#include <BdsD.h>
```

Public Member Functions

- SelectionInfo (**BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**(), **BList**< **BString** > networks= **BList**< **BString** >(), **BList**< **BString** > arrays= **BList**< **BString** >(), **BList**< **BString** > stations= **BList**< **BString** >(), **BList**< **BString** > arraysAndStations= **BList**< **BString** >(), **BList**< **BString** > channels= **BList**< **BString** >(), **BList**< **BString** > sources= **BList**< **BString** >(), **BUInt32** numDataChannels=0)

Public Attributes

- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time.
- **BList**< **BString** > networks

  The list of Network Names.
- **BList**< **BString** > arrays

  The list of Array names.
- **BList**< **BString** > stations

  The list of Station names.
- **BList**< **BString** > arraysAndStations

  The list of Array and Station names.
- **BList**< **BString** > channels

  The list of Channels.
- **BList**< **BString** > sources

  The list of Data Sources.
- **BUInt32** numDataChannels

  The number of sets of data in the system matching the criteria.

7.79.1 Detailed Description

This class defines the set of metadata or siesmic data selected when getSelectionInfo() is use.

This provides information on everything selected by a Selection object from the BDS metadata or seismic data sets.

7.79.2 Constructor & Destructor Documentation

7.79.2.1 SelectionInfo()

```
Bds::SelectionInfo::SelectionInfo (
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BList< BString > networks = BList< BString >(),
            BList< BString > arrays = BList< BString >(),
            BList< BString > stations = BList< BString >(),
            BList< BString > arraysAndStations = BList< BString >(),
            BList< BString > channels = BList< BString >(),
            BList< BString > sources = BList< BString >(),
            BUInt32 numDataChannels = 0 )
```

7.79.3 Member Data Documentation

7.79.3.1 arrays

```
BList< BString > Bds::SelectionInfo::arrays
```

The list of Array names.

7.79.3.2 arraysAndStations

```
BList< BString > Bds::SelectionInfo::arraysAndStations
```

The list of Array and Station names.

7.79.3.3 channels

```
BList< BString > Bds::SelectionInfo::channels
```

The list of Channels.

7.79.3.4 endTime

```
BTimeStamp Bds::SelectionInfo::endTime
```

The End Time.

7.79.3.5 networks

```
BList< BString > Bds::SelectionInfo::networks
```

The list of Network Names.

7.79.3.6 numDataChannels

```
BUInt32 Bds::SelectionInfo::numDataChannels
```

The number of sets of data in the system matching the criteria.

7.79.3.7 sources

```
BList< BString > Bds::SelectionInfo::sources
```

The list of Data Sources.

7.79.3.8 startTime

```
BTimeStamp Bds::SelectionInfo::startTime
```

The Start Time.

7.79.3.9 stations

```
BList< BString > Bds::SelectionInfo::stations
```

The list of Station names.

The documentation for this class was generated from the following files:

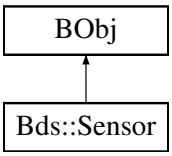
- BdsD.h
- BdsD.cc

7.80 Bds::Sensor Class Reference

This class defines a seismic Sensor.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::Sensor:



Public Member Functions

- Sensor (**BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTime**↩ **Stamp**(), **BString** name= **BString**(), **BString** type= **BString**(), **BString** serialNumber= **BString**(), **BUInt32** numberChannels=0, **BString** gainUnits= **BString**(), **BFloat64** gain=0, **BUInt32** oldId=0, **BInt32** shared=0)
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  The ID.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time.
- **BString** name

  The Sensors name.
- **BString** type

  The type of sensor. (Seismometer, Hydrophone etc)
- **BString** serialNumber

  The sensor's serial number. Only used when there is a unique physical sensor.
- **BUInt32** numberChannels

  The number of supported channels.
- **BString** gainUnits

  The gain units.
- **BFloat64** gain

  The overall gain of the sensor at the manufacturers calibration frequency. (For information only)
- **BUInt32** oldId

  The Id from the old Autodrm database.
- **BInt32** shared

  This sensor is shared.

7.80.1 Detailed Description

This class defines a seismic Sensor.

This just stores information on the seismic instrument's senosr. Its contents is generally for information only.

7.80.2 Constructor & Destructor Documentation

7.80.2.1 Sensor()

```
Bds::Sensor::Sensor (
            BUInt32 id = 0,
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BString name = BString(),
            BString type = BString(),
            BString serialNumber = BString(),
            BUInt32 numberChannels = 0,
            BString gainUnits = BString(),
            BFloat64 gain = 0,
            BUInt32 oldId = 0,
            BInt32 shared = 0 )
```

7.80.3 Member Function Documentation

7.80.3.1 getMember()

```
BError Bds::Sensor::getMember (
            BString name,
            BString & value )  [virtual]
```

Reimplemented from **BObj**.

7.80.3.2 getMembers()

```
BError Bds::Sensor::getMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.80.3.3 getType()

```
BString Bds::Sensor::getType ( )
```

7.80.3.4 setMember()

```
BError Bds::Sensor::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.80.3.5 setMembers()

```
BError Bds::Sensor::setMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.80.4 Member Data Documentation

7.80.4.1 endTime

```
BTimeStamp Bds::Sensor::endTime
```

The End Time.

7.80.4.2 gain

```
BFloat64 Bds::Sensor::gain
```

The overall gain of the sensor at the manufacturers calibration frequency. (For information only)

7.80.4.3 gainUnits

```
BString Bds::Sensor::gainUnits
```

The gain units.

7.80.4.4 id

```
BUInt32 Bds::Sensor::id
```

The ID.

7.80.4.5 name

```
BString Bds::Sensor::name
```

The Sensors name.

7.80.4.6 numberChannels

```
BUInt32 Bds::Sensor::numberChannels
```

The number of supported channels.

7.80.4.7 oldId

```
BUInt32 Bds::Sensor::oldId
```

The Id from the old Autodrm database.

7.80.4.8 serialNumber

```
BString Bds::Sensor::serialNumber
```

The sensor's serial number. Only used when there is a unique physical sensor.

7.80.4.9 shared

```
BInt32 Bds::Sensor::shared
```

This sensor is shared.

7.80.4.10 startTime

```
BTimeStamp Bds::Sensor::startTime
```

The Start Time.

7.80.4.11 type

```
BString Bds::Sensor::type
```

The type of sensor. (Seismometer, Hydrophone etc)

The documentation for this class was generated from the following files:

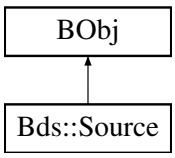
- BdsD.h
- BdsD.cc

7.81 Bds::Source Class Reference

This class defines a seismic data Source.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::Source:



Public Member Functions

- Source (**BUInt32** id=0, **BString** source= **BString**(), **BString** sourceMeta= **BString**(), **BString** alias= **B**↩ **String**(), **BString** description= **BString**())
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  Unique ID when stored in a database or for other uses.
- **BString** source

  The sensor data's source name.
- **BString** sourceMeta

  The associated metadata's source name.
- **BString** alias

  The short alias for data files.
- **BString** description

  The description.

7.81.1 Detailed Description

This class defines a seismic data Source.

A Seismic data source allows different sources of data to be described and allows different Metadata sets to be used with the different data sources. It might be that there were two different digitisers in use or one data set was received real-time though a particular data processing chain while the other was via CD medium with a different processing chain.

7.81.2 Constructor & Destructor Documentation

7.81.2.1 Source()

```
Bds::Source::Source (
            BUInt32 id = 0,
            BString source = BString(),
            BString sourceMeta = BString(),
            BString alias = BString(),
            BString description = BString() )
```

7.81.3 Member Function Documentation

7.81.3.1 getMember()

```
BError Bds::Source::getMember (
            BString name,
            BString & value ) [virtual]
```

Reimplemented from **BObj**.

7.81.3.2 getMembers()

```
BError Bds::Source::getMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.81.3.3 getType()

```
BString Bds::Source::getType ( )
```

7.81.3.4 setMember()

```
BError Bds::Source::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.81.3.5 setMembers()

```
BError Bds::Source::setMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.81.4 Member Data Documentation

7.81.4.1 alias

```
BString Bds::Source::alias
```

The short alias for data files.

7.81.4.2 description

```
BString Bds::Source::description
```

The description.

7.81.4.3 id

```
BUInt32 Bds::Source::id
```

Unique ID when stored in a database or for other uses.

7.81.4.4 source

```
BString Bds::Source::source
```

The sensor data's source name.

7.81.4.5 sourceMeta

```
BString Bds::Source::sourceMeta
```

The associated metadata's source name.

The documentation for this class was generated from the following files:

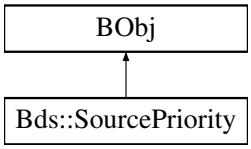
- BdsD.h
- BdsD.cc

7.82 Bds::SourcePriority Class Reference

This class defines a Source Priority entry.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::SourcePriority:



Public Member Functions

- SourcePriority (**BUInt32** id=0, **BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTime**↩ **Stamp**(), **BString** source= **BString**(), **BUInt32** priority=0)
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  Unique ID when stored in a database or for other uses.
- **BTimeStamp** startTime

  The Start Time.
- **BTimeStamp** endTime

  The End Time the channel was available.
- **BString** source

  The source name.
- **BUInt32** priority

  The priority order, highest first.

7.82.1 Detailed Description

This class defines a Source Priority entry.

This allows the default source for data to be selected based on a priority level. It allows a particular source to be used if no other is available and then prioritised through all the different sources available.

7.82.2 Constructor & Destructor Documentation

7.82.2.1 SourcePriority()

```
Bds::SourcePriority::SourcePriority (
            BUInt32 id = 0,
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp(),
            BString source = BString(),
            BUInt32 priority = 0 )
```

7.82.3 Member Function Documentation

7.82.3.1 getMember()

```
BError Bds::SourcePriority::getMember (
            BString name,
            BString & value )  [virtual]
```

Reimplemented from **BObj**.

7.82.3.2 getMembers()

```
BError Bds::SourcePriority::getMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.82.3.3 getType()

```
BString Bds::SourcePriority::getType ( )
```

7.82.3.4 setMember()

```
BError Bds::SourcePriority::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.82.3.5 setMembers()

```
BError Bds::SourcePriority::setMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.82.4 Member Data Documentation

7.82.4.1 endTime

```
BTimeStamp Bds::SourcePriority::endTime
```

The End Time the channel was available.

7.82.4.2 id

```
BUInt32 Bds::SourcePriority::id
```

Unique ID when stored in a database or for other uses.

7.82.4.3 priority

```
BUInt32 Bds::SourcePriority::priority
```

The priority order, highest first.

7.82.4.4 source

```
BString Bds::SourcePriority::source
```

The source name.

7.82.4.5 startTime

```
BTimeStamp Bds::SourcePriority::startTime
```

The Start Time.

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

7.83 Bds::Station Class Reference

This class defines a seismic station.

```
#include <BdsD.h>
```

Public Member Functions

- Station (**BUInt32** id=0, **BString** name= **BString**(), **BString** alias= **BString**(), **BString** type= **BString**(), **BString** description= **BString**(), **BList**< ArrayChannel > channels= **BList**< ArrayChannel >())

Public Attributes

- **BUInt32** id

  Unique ID when stored in a database or for other uses.
- **BString** name

  The name.
- **BString** alias

  Alias name to be returned to the user.
- **BString** type

  The Station type. Set to "array" or "station".
- **BString** description

  Description.
- **BList**< ArrayChannel > channels

  List of channels if an Array.

7.83.1 Detailed Description

This class defines a seismic station.

A Station could be an individual station or an array of stations. If it is an array of stations the Station object contains the list of chaannels that make up the array. This class defines a Station

7.83.2 Constructor & Destructor Documentation

7.83.2.1 Station()

```
Bds::Station::Station (
            BUInt32 id = 0,
            BString name =  BString(),
            BString alias =  BString(),
            BString type =  BString(),
            BString description =  BString(),
            BList< ArrayChannel > channels =  BList<ArrayChannel >() )
```

7.83.3 Member Data Documentation

7.83.3.1 alias

```
BString Bds::Station::alias
```

Alias name to be returned to the user.

7.83.3.2 channels

```
BList<ArrayChannel > Bds::Station::channels
```

List of channels if an Array.

7.83.3.3 description

```
BString Bds::Station::description
```

Description.

7.83.3.4 id

```
BUInt32 Bds::Station::id
```

Unique ID when stored in a database or for other uses.

7.83.3.5 name

```
BString Bds::Station::name
```

The name.

7.83.3.6 type

```
BString Bds::Station::type
```

The Station type. Set to "array" or "station".

The documentation for this class was generated from the following files:

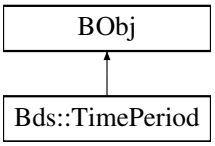
- BdsD.h
- BdsD.cc

7.84 Bds::TimePeriod Class Reference

This class defines a TimePeriod.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::TimePeriod:



Public Member Functions

- TimePeriod (**BTimeStamp** startTime= **BTimeStamp**(), **BTimeStamp** endTime= **BTimeStamp**())
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BTimeStamp** startTime

  The Start time to the nearest us.
- **BTimeStamp** endTime

  The End time to the nearest us.

7.84.1 Detailed Description

This class defines a TimePeriod.

It has startTime and endTime fields. Note the endTime is not included in the period.

7.84.2 Constructor & Destructor Documentation

7.84.2.1 TimePeriod()

```
Bds::TimePeriod::TimePeriod (
            BTimeStamp startTime = BTimeStamp(),
            BTimeStamp endTime = BTimeStamp() )
```

7.84.3 Member Function Documentation

7.84.3.1 getMember()

```
BError Bds::TimePeriod::getMember (
            BString name,
            BString & value )  [virtual]
```

Reimplemented from **BObj**.

7.84.3.2 getMembers()

```
BError Bds::TimePeriod::getMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.84.3.3 getType()

```
BString Bds::TimePeriod::getType ( )
```

7.84.3.4 setMember()

```
BError Bds::TimePeriod::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.84.3.5 setMembers()

```
BError Bds::TimePeriod::setMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.84.4 Member Data Documentation

7.84.4.1 endTime

```
BTimeStamp Bds::TimePeriod::endTime
```

The End time to the nearest us.

7.84.4.2 startTime

```
BTimeStamp Bds::TimePeriod::startTime
```

The Start time to the nearest us.

The documentation for this class was generated from the following files:

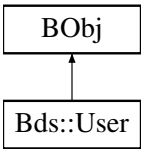
- BdsD.h
- BdsD.cc

7.85 Bds::User Class Reference

This holds information on a user.

```
#include <BdsD.h>
```

Inheritance diagram for Bds::User:



Public Member Functions

- User (**BUInt32** id=0, **BString** user= **BString**(), **BString** password= **BString**(), **BString** name= **BString**(), **BString** email= **BString**(), **BString** telephone= **BString**(), **BString** address= **BString**(), **BInt32** enabled=0, **BList**< **BString** > groups= **BList**< **BString** >())
- **BString** getType ()
- **BError** setMembers (**BDictString** &members)
- **BError** setMember (**BString** name, **BString** value)
- **BError** getMembers (**BDictString** &members)
- **BError** getMember (**BString** name, **BString** &value)

Public Attributes

- **BUInt32** id

  The unique user ID.
- **BString** user

  The User ID.
- **BString** password

  The Users password.
- **BString** name

  The Users full name.
- **BString** email

  The users email Address.
- **BString** telephone

  The Users telephone number.
- **BString** address

  The Users postal address.
- **BInt32** enabled

  Whether the users account is enabled.
- **BList**< **BString** > groups

  The security groups the user belongs to.

7.85.1 Detailed Description

This holds information on a user.

All information on a BDS users is stored along with the security groups they belong to.

7.85.2 Constructor & Destructor Documentation

7.85.2.1 User()

```
Bds::User::User (
            BUInt32 id = 0,
            BString user = BString(),
            BString password = BString(),
            BString name = BString(),
            BString email = BString(),
            BString telephone = BString(),
            BString address = BString(),
            BInt32 enabled = 0,
            BList< BString > groups = BList< BString >() )
```

7.85.3 Member Function Documentation

7.85.3.1 getMember()

```
BError Bds::User::getMember (
            BString name,
            BString & value )  [virtual]
```

Reimplemented from **BObj**.

7.85.3.2 getMembers()

```
BError Bds::User::getMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.85.3.3 getType()

```
BString Bds::User::getType ( )
```

7.85.3.4 setMember()

```
BError Bds::User::setMember (
            BString name,
            BString value )  [virtual]
```

Reimplemented from **BObj**.

7.85.3.5 setMembers()

```
BError Bds::User::setMembers (
            BDictString & members )  [virtual]
```

Reimplemented from **BObj**.

7.85.4 Member Data Documentation

7.85.4.1 address

```
BString Bds::User::address
```

The Users postal address.

7.85.4.2 email

```
BString Bds::User::email
```

The users email Address.

7.85.4.3 enabled

```
BInt32 Bds::User::enabled
```

Whether the users account is enabled.

7.85.4.4 groups

```
BList< BString > Bds::User::groups
```

The security groups the user belongs to.

7.85.4.5 id

```
BUInt32 Bds::User::id
```

The unique user ID.

7.85.4.6 name

```
BString Bds::User::name
```

The Users full name.

7.85.4.7 password

```
BString Bds::User::password
```

The Users password.

7.85.4.8 telephone

```
BString Bds::User::telephone
```

The Users telephone number.

7.85.4.9 user

```
BString Bds::User::user
```

The User ID.

The documentation for this class was generated from the following files:

- BdsD.h
- BdsD.cc

Chapter 8

File Documentation

8.1 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.cpp File Reference

```
#include <BdsCompress.h>
#include <BEndian.h>
```

Namespaces

- Bds

Functions

- **BError** Bds::bdsUnCompressCm8 (**BUInt8** ∗buffer, **BUInt** n, **BArray**< **BInt32** > & **data**)

  Uncompress CM8 formatted data.

8.2 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.d File Reference

8.3 /src/blacknest/bds/bds/bdsDataLib/BdsCompress.h File Reference

```
#include <BError.h>
#include <BArray.h>
```

Classes

- class Bds::CompressSteim1

  Steim1 un-compress class.

Namespaces

- Bds

Functions

- **BError** Bds::bdsUnCompressCm8 (**BUInt8** ∗buffer, **BUInt** n, **BArray**< **BInt32** > & **data**)

  Uncompress CM8 formatted data.
- **BError** Bds::bdsUnCompressSteim1 (**BUInt8** ∗buffer, **BUInt** n, **BArray**< **BInt32** > & **data**)

  Uncompress STEIM1 formatted data.

8.4 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.cpp File Reference

```
#include <BdsDataCollate.h>
```

Namespaces

- Bds

8.5 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.d File Reference

8.6 /src/blacknest/bds/bds/bdsDataLib/BdsDataCollate.h File Reference

```
#include <BString.h>
#include <BFile.h>
#include <BTimeStamp.h>
#include <BdsD.h>
#include <BdsDataFile.h>
```

Classes

- class Bds::DataCollate

  Not sure if this is used or what it does.

Namespaces

- Bds

8.7 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.cpp File Reference

```
#include <BdsDataFile.h>
```

Namespaces

- Bds

8.8 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.d File Reference

8.9 /src/blacknest/bds/bds/bdsDataLib/BdsDataFile.h File Reference

```
#include <BString.h>
#include <BFile.h>
#include <BdsLib.h>
#include <BdsD.h>
```

Classes

- class Bds::DataBlockPos

  This defines the position of a data block in a file. It is used by the BDS data convertors to order blocks by time.
- class Bds::DataFileOptions

  This defines a list of BDS data convtor options.
- class Bds::DataFile

  This class defines the interface for generic data file access that all of the BDS data conterors share.

Namespaces

- Bds

8.10 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.cpp File Reference

```
#include <BdsDataFileAd22.h>
#include <TimeControlPll.h>
#include <BDebug.h>
#include <errno.h>
```

Namespaces

- Bds

Macros

- #define DEBUG_VELATRACK 1

8.10.1 Macro Definition Documentation

8.10.1.1 DEBUG_VELATRACK

```
#define DEBUG_VELATRACK 1
```

8.11 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.d File Reference

8.12 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAd22.h File Reference

```
#include <BdsDataFile.h>
```

Classes

- class Bds::DataFileAd22

  Data file convertor for AD22 format files.

Namespaces

- Bds

8.13 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.cpp File Reference

```
#include <BdsDataFileAscii.h>
#include <BTimeStamp.h>
```

Namespaces

- Bds

Functions

- static **BString** Bds::nullString (**BString** s)

8.14 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.d File Reference

8.15 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileAscii.h File Reference

```
#include <BdsDataFile.h>
#include <BdsD.h>
```

Classes

- class Bds::DataFileAscii

  Data file convertor for ASCII format files.

Namespaces

- Bds

8.16 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.cpp File Reference

```
#include <BdsDataFileBdrs.h>
#include <BDebug.h>
#include <errno.h>
```

Namespaces

- Bds

8.17 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.d File Reference

8.18 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBdrs.h File Reference

```
#include <BdsDataFile.h>
```

Classes

- class Bds::DataFileBdrs

  Data file convertor for BDRS format files.

Namespaces

- Bds

8.19 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <errno.h>
#include <byteswap.h>
#include <BdsLib.h>
#include <BdsDataLib.h>
#include <BdsDataFileBds.h>
#include <BBuffer.h>
#include <BDebug.h>
#include <zlib.h>
#include <canada_compress.h>
```

Namespaces

- Bds

Macros

- #define LDEBUG 0
- #define LDEBUG2 0
- #define LDEBUG3 0
- #define dlprintf(fmt, a...)
- #define dl2printf(fmt, a...)
- #define dl3printf(fmt, a...)
- #define ALLOW_TIMESTAMP_JITTER 1
- #define TIMESTAMP_JITTER 100

Functions

- **BUInt32** Bds::crc (**BUInt32** crc, void ∗ **data**, int numBytes)

Variables

- const **BString** Bds::BdsDataFileVersion = "1.2.0"

8.19.1 Macro Definition Documentation

8.19.1.1 ALLOW_TIMESTAMP_JITTER

```
#define ALLOW_TIMESTAMP_JITTER 1
```

8.19.1.2 dl2printf

```
#define dl2printf(
              fmt,
              a...  )
```

8.19.1.3 dl3printf

```
#define dl3printf(
              fmt,
              a...  )
```

8.19.1.4 dlprintf

```
#define dlprintf(
              fmt,
              a...  )
```

8.19.1.5 LDEBUG

```
#define LDEBUG 0
```

8.19.1.6 LDEBUG2

```
#define LDEBUG2 0
```

8.19.1.7 LDEBUG3

```
#define LDEBUG3 0
```

8.19.1.8 TIMESTAMP_JITTER

```
#define TIMESTAMP_JITTER 100
```

8.20 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.d File Reference

8.21 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBds.h File Reference

```
#include <BdsDataFile.h>
#include <BBuffer.h>
```

Classes

- struct Bds::BdsDataBlockHeader

  BdsDataFileBds: internal fixed size BDS Data Block header.
- struct Bds::BdsDataBlock

  BdsDataFileBds: internal fixed size BDS Data Block.
- struct Bds::BdsDataPacketHeader

  BdsDataFileBds internal file storage packet header.
- class Bds::BdsDataPacket

  BdsDataFileBds: internal file storage packet.
- class Bds::BdsDataBlockPos

  BdsDataFileBds: internal file storage data block position.
- class Bds::BdsDataSegment

  BdsDataFileBds: internal file storage data segment.
- class Bds::BdsDataStreamlet

  BdsDataFileBds: internal file storage data streamlet.
- class Bds::DataFileBds

  This class implements the BDS Data File/Stream access system.

Namespaces

- Bds

Enumerations

- enum Bds::BdsDataType { Bds::BdsDataTypeBlock = 0x42534442, Bds::BdsDataTypeInfo = 0x30534442, Bds::BdsDataTypeData = 0x31534442, Bds::BdsDataTypeInfoExtra = 0x32534442 }

  BdsDataFileBds: internal file block type field.

8.22 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.cpp File Reference

```
#include <BdsDataFileBknas.h>
#include <BTimeStamp.h>
#include <limits.h>
```

Namespaces

- Bds

Functions

- template<typename T >
  T clip (T in, T low, T high)

8.22.1 Function Documentation

8.22.1.1 clip()

```
template<typename T >
T clip (
            T in,
            T low,
            T high )  [inline]
```

8.23 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.d File Reference

8.24 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileBknas.h File Reference

```
#include <BdsDataFile.h>
#include <BdsD.h>
```

Classes

- class Bds::DataFileBknas

  Data file convertor for BKNAS format files.

Namespaces

- Bds

8.25 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileCd.cpp File Reference

```
#include <BdsDataFileCd.h>
#include <arpa/inet.h>
#include <errno.h>
#include <canada_compress.h>
#include <BEndian.h>
#include <BDebug.h>
```

Namespaces

- Bds

Macros

- #define LDEBUG 0
- #define dprintf(fmt, a...)
- #define INCLUDE_CHANNEL_AUTH 1
- #define ALLOW_TIMESTAMP_JITTER 1
- #define TIMESTAMP_JITTER 100
- #define MULTIPLE_SEGMENT 0
- #define SEGMENT_GAP 3600000
- #define ntohll(x) __bswap_64(x)
- #define htonll(x) ntohll(x)

Functions

- static void Bds::crcInit ()
- 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::crcInitDone

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 htonll

```
#define htonll(
              x ) ntohll(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/BdsDataFileIms.cpp File Reference

```
#include <BdsDataFileIms.h>
#include <BTimeStamp.h>
#include <errno.h>
```

Namespaces

- Bds

Functions

- static **BError** Bds::fixedString (double v, int fieldWidth, int numDecimal, **BString** &str)
- void Bds::dataCalculateDifference (**BInt32** &prevValue, **BArray**< **BInt32** > & **data**)
- void Bds::dataCalculateUnDifference (**BInt32** &prevValue, **BArray**< **BInt32** > & **data**)
- **BInt32** Bds::dataChecksum (**BInt32** **checksum**, **BArray**< **BInt32** > & **data**)
- **BError** Bds::dataCompressCm6 (int &prevValue1, int &prevValue2, **BArray**< **BInt32** > & **data**, **BString** &d)
- **BError** Bds::dataDeCompressCm6 (int &prevValue1, int &prevValue2, **BString** &d, **BArray**< **BInt32** > & **data**)
- static void Bds::dataConvert (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BInt32** > &dataOut)

Variables

- static char Bds::cm6Table [64]
- static **BUInt8** Bds::cm6TableRev [128]

8.35 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.d File Reference

8.36 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileIms.h File Reference

```
#include <BdsDataFile.h>
#include <BdsD.h>
```

Classes

- class Bds::DataFileIms

  Data file convertor for IMS format files.

Namespaces

- Bds

8.37 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.cpp File Reference

```
#include <BdsDataFileLac.h>
#include <BEndian.h>
#include <BDebug.h>
#include <errno.h>
```

Namespaces

- Bds

8.38 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.d File Reference

8.39 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLac.h File Reference

```
#include <BdsDataFile.h>
```

Classes

- class Bds::DataFileLac

  Data file convertor for LAC format files.

Namespaces

- Bds

8.40 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.cpp File Reference

```
#include <BdsDataFileLog.h>
#include <BTimeStamp.h>
#include <errno.h>
```

Namespaces

- Bds

Functions

- static **BString** Bds::stringFormat (**BTimeStamp** t)
- static **BString** Bds::removeCR (**BString** str)

8.41 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.d File Reference

8.42 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileLog.h File Reference

```
#include <BdsDataFile.h>
#include <BdsD.h>
```

Classes

- class Bds::DataFileLog

  Data file convertor for LOG format files.

Namespaces

- Bds

8.43 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.cpp File Reference

```
#include <BdsDataFileResponse.h>
#include <errno.h>
```

Namespaces

- Bds

Macros

- #define LDEBUG 0
- #define dprintf(fmt, a...)

8.43.1 Macro Definition Documentation

8.43.1.1 dprintf

```
#define dprintf(
              fmt,
              a...  )
```

8.43.1.2 LDEBUG

```
#define LDEBUG 0
```

8.44 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.d File Reference

8.45 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileResponse.h File Reference

```
#include <BdsDataFile.h>
#include <BdsD.h>
```

Classes

- class Bds::DataFileResponse

  This class defines the interface for generic response data file access.

Namespaces

- Bds

8.46 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.cpp File Reference

```
#include <BdsDataFileSac.h>
#include <errno.h>
```

Namespaces

- Bds

8.47 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.d File Reference

8.48 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileSac.h File Reference

```
#include <BdsDataFile.h>
#include <BdsD.h>
```

Classes

- class Bds::DataFileSac

  Data file convertor for SAC format files.

Namespaces

- Bds

8.49 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.cpp File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <errno.h>
#include <BdsDataFileTapeDigitiser.h>
#include <BTimeStampMs.h>
```

Namespaces

- Bds

Enumerations

- enum Bds::FileHeaderType { Bds::FileHeaderType_Standard = 1, Bds::FileHeaderType_TapeDigitiser = 10 }
- enum Bds::FileSampleType { Bds::FileSampleType_Unknown, Bds::FileSampleType_Float32, Bds::FileSampleType_Float64, Bds::FileSampleType_Int16, Bds::FileSampleType_Int32 }

Variables

- const double Bds::Scale = 16777216.0

8.50 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.d File Reference

8.51 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileTapeDigitiser.h File Reference

```
#include <BTypes.h>
#include <BError.h>
#include <BFile.h>
#include <BEntry.h>
#include <BBuffer.h>
#include <BDict.h>
#include <BdsDataFile.h>
```

Classes

- class Bds::DataFileTapeDigitiser

  This class implements the TapeDigitiser's file output conversion and storing system.

Namespaces

- Bds

8.52 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.cpp File Reference

```
#include <BdsDataFileWra.h>
#include <BDebug.h>
#include <errno.h>
```

Namespaces

- Bds

8.53 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.d File Reference

8.54 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWra.h File Reference

```
#include <BdsDataFile.h>
```

Classes

- class Bds::DataFileWra

  Data file convertor for WRA format files.

Namespaces

- Bds

8.55 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.cpp File Reference

```
#include <BdsDataFileWraAgso.h>
#include <BdsCompress.h>
#include <BDebug.h>
#include <errno.h>
#include <math.h>
```

Namespaces

- Bds

Functions

- static **BList**< **BString** > parseStringFixedFields (**BString** s, int *fieldWidths)

8.55.1 Function Documentation

8.55.1.1 parseStringFixedFields()

```
static BList< BString> parseStringFixedFields (
            BString s,
            int * fieldWidths ) [static]
```

8.56 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.d File Reference

8.57 /src/blacknest/bds/bds/bdsDataLib/BdsDataFileWraAgso.h File Reference

```
#include <BdsDataFile.h>
```

Classes

- class Bds::DataFileWraAgso

  Data file convertor for WRA AGSO format files.

Namespaces

- Bds

8.58 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.cpp File Reference

```
#include <BdsDataLib.h>
#include <BdsDataFileAscii.h>
#include <BdsDataFileBknas.h>
#include <BdsDataFileBdrs.h>
#include <BdsDataFileBds.h>
#include <BdsDataFileGcf.h>
#include <BdsDataFileIms.h>
#include <BdsDataFileTapeDigitiser.h>
#include <BdsDataFileWra.h>
#include <BdsDataFileWraAgso.h>
#include <BdsDataFileSeed.h>
#include <BdsDataFileSac.h>
#include <BdsDataFileCd.h>
#include <BdsDataFileResponse.h>
#include <BdsDataFileLog.h>
#include <BdsDataFileAd22.h>
#include <BdsDataFileLac.h>
#include <BdsDataFileCss.h>
```

Namespaces

- Bds

Functions

- **BString** Bds::fixedWidthValue (double v, int width)

  This returns a double as a fixed width string truncating the data.

Variables

- DataFormats Bds::dataFormats

8.59 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.d File Reference

8.60 /src/blacknest/bds/bds/bdsDataLib/BdsDataLib.h File Reference

```
#include <BdsDataFile.h>
```

Classes

- class Bds::DataFormats

  This class defines the interface for generic data file access.

Namespaces

- Bds

Functions

- **BString** Bds::fixedWidthValue (double v, int width)

  This returns a double as a fixed width string truncating the data.

8.61 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.cpp File Reference

```
#include <BdsDataFileSeed.h>
#include <BEndian.h>
#include <errno.h>
#include <BDebug.h>
```

Namespaces

- Bds

Macros

- #define DEBUG 0
- #define DEBUG_BLOCKETTE 0
- #define DEBUG_BLOCKS 0
- #define FILL_BLOCKS 1
- #define ROUND_TIMESTAMPS_TO_10US 1
- #define dlprintf(fmt, a...)

Functions

- static void Bds::bdsDataFileSeedLogWarning (char *str)
- static void Bds::bdsDataFileSeedLogError (char *str)
- static hptime_t Bds::seedTime (**BTimeStamp** t)
- static **BString** Bds::seedTimeString (**BTimeStamp** t)
- static void Bds::dataConvert (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BInt32** > &dataOut)
- static void Bds::dataConvert (const **BArray**< **BFloat64** > &dataIn, **BArray**< **BFloat32** > &dataOut)
- static void Bds::record_handler (char *record, int reclen, void *info)

8.61.1 Macro Definition Documentation

8.61.1.1 DEBUG

```
#define DEBUG 0
```

8.61.1.2 DEBUG_BLOCKETTE

```
#define DEBUG_BLOCKETTE 0
```

8.61.1.3 DEBUG_BLOCKS

```
#define DEBUG_BLOCKS 0
```

8.61.1.4 dlprintf

```
#define dlprintf(
              fmt,
              a...  )
```

8.61.1.5 FILL_BLOCKS

```
#define FILL_BLOCKS 1
```

8.61.1.6 ROUND_TIMESTAMPS_TO_10US

```
#define ROUND_TIMESTAMPS_TO_10US 1
```

8.62 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.d File Reference

8.63 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsDataFileSeed.h File Reference

```
#include <BdsDataFile.h>
#include <BdsSeedTypes.h>
#include <BMutex.h>
#include <libmseed-beam/libmseed.h>
#include <libmseed-beam/unpackdata.h>
```

Classes

- class Bds::DataFileSeed

  Data file convertor for SEED file formats.

Namespaces

- Bds

8.64 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.cpp File Reference

```
#include <BdsSeedType.h>
```

Namespaces

- Bds

8.65 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.d File Reference

8.66 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedType.h File Reference

```
#include <BError.h>
```

Classes

- class Bds::BdsSeedType

  BdsDataFileSeed internal parent for all SEED types.

Namespaces

- Bds

8.67 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.cpp File Reference

```
#include <BdsSeedTypes.h>
```

Namespaces

- Bds

8.68 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.d File Reference

8.69 /src/blacknest/bds/bds/bdsDataLib/BdsSeed/BdsSeedTypes.idl File Reference

8.70 /src/blacknest/bds/bds/bdsDataLib/canada_compress.d File Reference

8.71 /src/blacknest/bds/bds/bdsDataLib/canada_compress.h File Reference

```
#include <arpa/inet.h>
```

Macros

- #define CANCOMP_ERR -1 /∗ unrecoverable **error** (malloc fails) ∗/
- #define CANCOMP_SUCCESS 0 /∗ success ∗/
- #define CANCOMP_NOT_20 1 /∗ **number** of samples not divisible by 20 ∗/
- #define CANCOMP_CORRUPT 2 /∗ corrupted call ∗/
- #define CANCOMP_EXCEED

Functions

- int canada_uncompress (unsigned char ∗b, uint32_t ∗y, int ∗n, int m, uint32_t ∗v0)

  De-compressses Canada format seismic data.
- int canada_compress (unsigned char ∗b, unsigned long ∗y, int ∗n, int m, unsigned long ∗v0)

  Compressses Canada format seismic data.

8.71.1 Macro Definition Documentation

8.71.1.1 CANCOMP_CORRUPT

```
#define CANCOMP_CORRUPT 2 /* corrupted call */
```

8.71.1.2 CANCOMP_ERR

```
#define CANCOMP_ERR -1 /* unrecoverable  error (malloc fails) */
```

8.71.1.3 CANCOMP_EXCEED

```
#define CANCOMP_EXCEED
```

Value:

```
3   /* number of bytes available in compressed
                  data exceeded during decompression */
```

8.71.1.4 CANCOMP_NOT_20

```
#define CANCOMP_NOT_20 1 /*  number of samples not divisible by 20 */
```

8.71.1.5 CANCOMP_SUCCESS

```
#define CANCOMP_SUCCESS 0 /* success */
```

8.71.2 Function Documentation

8.71.2.1 canada_compress()

```
int canada_compress (
            unsigned char * b,
            unsigned long * y,
            int * n,
            int m,
            unsigned long * v0 )
```

Compressses Canada format seismic data.

8.71.2.2 canada_uncompress()

```
int canada_uncompress (
            unsigned char * b,
            uint32_t * y,
            int * n,
            int m,
            uint32_t * v0 )
```

De-compressses Canada format seismic data.

8.72 BdsC.cc File Reference

```
#include <BdsC.h>
```

Namespaces

- Bds

8.73 BdsC.d File Reference

8.74 BdsC.h File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <Boap.h>
#include <BString.h>
#include <BList.h>
#include <BArray.h>
#include <BdsD.h>
```

Classes

- class Bds::DataAccess

  This is the Data Access API interface.
- class Bds::DataAddAccess

  This is the DataAdd Access API interface.
- class Bds::AdminAccess

  This is the AdminAccess Access API interface.

Namespaces

- Bds

Variables

- const **BUInt32** Bds::apiVersion = 0

8.75 BdsD.cc File Reference

```
#include <BdsD.h>
```

Namespaces

- Bds

8.76 BdsD.d File Reference

8.77 BdsD.h File Reference

BOAP data class definitions for: Bds.

```
#include <Boap.h>
#include <BObj.h>
#include <BDate.h>
#include <BTimeStamp.h>
#include <BComplex.h>
#include <BList.h>
#include <BArray.h>
```

Classes

- class Bds::Point

  This class defines an X,Y location.
- class Bds::TimePeriod

  This class defines a TimePeriod.
- class Bds::ListRange

  This class defines an integer based range.
- class Bds::Network

  This class defines a seismic Network organisation.
- class Bds::Source

  This class defines a seismic data Source.
- class Bds::SourcePriority

  This class defines a Source Priority entry.
- class Bds::ChannelName

  This class defines a full channel name.
- class Bds::ArrayChannel

  This class defines an arrays channel.
- class Bds::Station

  This class defines a seismic station.
- class Bds::Location

  This class defines the physical location of a Station.
- class Bds::PoleZero

  This class defines a Pole/Zero Response.
- class Bds::Fap

  This class defines an entry in an Amplitude/Phase Response table.
- class Bds::FirEntry

  This class defines an entry in a FIR coefficient table.
- class Bds::Fir

  This class defines an FIR response table.
- class Bds::Response

  This class defines a seismic Response characteristic.
- class Bds::Calibration

  This class defines a calibration setting.
- class Bds::Digitiser

  This class defines a seismic Digitiser.
- class Bds::Sensor

  This class defines a seismic Sensor.
- class Bds::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 siesmic 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 avaiilablity 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 acual seismic data values contained within a single data block.
- class Bds::User

  This holds information on a user.
- class Bds::Group

  This holds information on a user security group.
- class Bds::AccessGroup

  This holds information on data access groups.
- class Bds::Change

  This holds information on a medatdata 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 cirteria when requesting a set of log entries.
- class Bds::CleanOptions

  This defines the set of clean options used in the clean() function.
- class Bds::DataFormat

  This holds information on a seismic data format.

Namespaces

- Bds

Enumerations

- enum Bds::Errors {
  Bds::ErrorNoMetaData = 64, Bds::ErrorDataQuality = 65, Bds::ErrorSlaveMode = 66, Bds::ErrorTimeStamp = 67,
  Bds::ErrorValidate = 80, Bds::ErrorValidateMissingBlocks = 81, Bds::ErrorValidateTimeBackwards = 82, Bds::ErrorValidateFilenameTime = 83,
  Bds::ErrorValidateMetaData = 84, Bds::ErrorValidateFix = 85, Bds::ErrorValidateDuplicate = 86, Bds::ErrorValidateReorder = 87,
  Bds::ErrorValidateBdsFudge = 88 }

The System Error number list in addition to standard system error numbers.

- enum Bds::Priority { Bds::PriorityLow, Bds::PriorityNormal, Bds::PriorityHigh }

  Priority levels.

- enum Bds::Mode { Bds::ModeMaster, Bds::ModeSlave }

  BdsServer mode.

- enum Bds::DataFlags { Bds::DataFlagNone = 0x00, Bds::DataFlagClipDataToTime = 0x01, Bds::DataFlagClipDataToChannels = 0x02, Bds::DataFlagMergeSegments = 0x04, Bds::DataFlagNoMetadata = 0x08 }

  Flags when opening data files.

- enum Bds::SelectionGroup { Bds::SelectionGroupData, Bds::SelectionGroupMetaData, Bds::SelectionGroupDataWithCount }

  The Selection group when making selections.

- enum Bds::SampleFormat { Bds::SampleFormatUnknown, Bds::SampleFormatInt16, Bds::SampleFormatInt32, Bds::SampleFormatFloat32, Bds::SampleFormatFloat64, Bds::SampleFormatInt24 }

  The actual format of a data sample.

- enum Bds::AvailType { Bds::AvailNone, Bds::AvailPartial, Bds::AvailFull }

  A flag defining the data availability state.

8.77.1 Detailed Description

BOAP data class definitions for: Bds.

Date

2020-06-23T15:00:48

The classes in here have been defined by a BOAP ∗.bidl file and define classes able to be communicated across a BOAP link

8.78 BdsLib.cpp File Reference

```
#include <BdsLib.h>
#include <math.h>
#include <complex>
```

Namespaces

- Bds

Functions

- void Bds::bdsDumpPoleZeros (PoleZero poleZeros)

  Debug print out a PoleZeros object.
- void Bds::bdsChannelGetTypeAux (**BString** name, **BString** &type, **BString** &aux)

  Get the channel type and aux fields from a generic channel name.
- **BString** Bds::bdsChannelGetName (**BString** type, **BString** aux)

  Create a full channel name from a channels type and aux fields.
- **BError** Bds::bdsDataInfoSetTimeRange (DataInfo &dataInfo)

  Restricts the time tange of all of the DataInfo's channels to match the DataInfo's startTime/endTime fields.
- **BError** Bds::bdsDataInfoFromInfo (**BDictString** info, DataInfo &dataInfo, **Bool** append)

  Convert info to DataInfo.
- **BError** Bds::bdsInfoFromDataInfo (const DataInfo &dataInfo, **BDictString** &info)

  Converts a DataInfo object into a BDictString list of named strings.
- **BError** Bds::bdsDataInfoMergeFlatten (DataInfo &dataInfo, const DataInfo &dataInfoAdd)

  Merges a DataInfo into another flattening the segments to 1 for use in dataOpen() calls.
- **BString** Bds::bdsStationAlias (Station station)

  Returns the station alias if set else its name.
- void Bds::bdsDumpSelection (Selection sel)

  Debug print out a Selection object.
- void Bds::bdsDumpDataInfo (DataInfo dataInfo, int includeInfo=0)

  Debug print out a DataInfo object.
- void Bds::bdsDumpChannelInfos (const ChannelInfos &channelInfos)

  Debug print out a ChannelInfos object.
- void Bds::bdsDumpData (const DataBlock &dataBlock, int nSamples=0)

  Debug print out a DataBlock object.
- **BString** Bds::bdsDataChannelInfo (const DataChannel &dataChannel)

  Returns a string representation of a DataChannel object.
- **BString** Bds::bdsDataChannelRef (const DataChannel &dataChannel)

  Returns the string reference name of a DataChannel object.
- **BString** Bds::bdsDataChannelRef (const ChannelInfo &channelInfo)

  Returns the string reference name of a ChannelInfo object.
- **BError** Bds::bdsDataChannelOverallResponse (const ChannelInfo &channelInfo, Response &response)

  Returns the overal response from the list of responses in a ChannelInfo.
- **BString** Bds::bdsSelectionChannelInfo (const Selection &selection, **BUInt** channel)

  Returns a string describing the name and time period of a selection channel.
- double Bds::bdsPoleZeroGain (const PoleZero &poleZero, double frequency)

  Calculates the overal gain of the given PoleZero transfer function.
- void Bds::bdsPoleZeroGainPhase (const PoleZero &poleZero, double frequency, double &gain, double &phase)

  Calculates the overal gain and phase of the given PoleZero transfer function.
- void Bds::bdsPoleZeroToFap (const PoleZero &poleZero, **BUInt** nPoints, double calibrationFrequency, double sampleFrequency, **BArray**< Fap > &fap)

  Convert PoleZero to FAP.
- static **BString** Bds::fileNameTime (**BTimeStamp** t)
- **BString** Bds::bdsFileNameExpand (**BString** fileName, ChannelInfo &channelInfo)

  Default filename from a ChannelInfo.
- **BString** Bds::bdsFileNameExpand (**BString** fileName, ChannelInfos &channelInfos)

  Default filename from a list of ChannelInfo's.

8.79 BdsLib.d File Reference

8.80 BdsLib.dox File Reference

Namespaces

- Bds

Functions

- void Bds::bdsChannelGetTypeAux (**BString** name, **BString** &type, **BString** &aux)

  Get the channel type and aux fields from a generic channel name.

8.81 BdsLib.h File Reference

General BdsLib API functions.

```
#include <BdsD.h>
#include <BdsC.h>
#include <BString.h>
```

Classes

- class Bds::ResponseObj

  Response object adding string conversion.
- class Bds::DataError

  This stores a data error. It includes and error number and a string as well as information on what seismic channel it is for.

Namespaces

- Bds

Functions

- void Bds::bdsChannelGetTypeAux (**BString** name, **BString** &type, **BString** &aux)

  Get the channel type and aux fields from a generic channel name.
- **BString** Bds::bdsChannelGetName (**BString** type, **BString** aux)

  Create a full channel name from a channels type and aux fields.
- **BError** Bds::bdsDataInfoSetTimeRange (DataInfo &dataInfo)

  Restricts the time tange of all of the DataInfo's channels to match the DataInfo's startTime/endTime fields.
- **BError** Bds::bdsDataInfoFromInfo (**BDictString** info, DataInfo &dataInfo, **Bool** append)

  Convert info to DataInfo.
- **BError** Bds::bdsInfoFromDataInfo (const DataInfo &dataInfo, **BDictString** &info)

  Converts a DataInfo object into a BDictString list of named strings.
- **BError** Bds::bdsDataInfoMergeFlatten (DataInfo &dataInfo, const DataInfo &dataInfoAdd)

  Merges a DataInfo into another flattening the segments to 1 for use in dataOpen() calls.
- **BString** Bds::bdsStationAlias (Station station)

  Returns the station alias if set else its name.
- **BString** Bds::bdsDataChannelInfo (const DataChannel &dataChannel)

  Returns a string representation of a DataChannel object.
- **BString** Bds::bdsDataChannelRef (const DataChannel &dataChannel)

  Returns the string reference name of a DataChannel object.
- **BString** Bds::bdsDataChannelRef (const ChannelInfo &channelInfo)

  Returns the string reference name of a ChannelInfo object.
- **BError** Bds::bdsDataChannelOverallResponse (const ChannelInfo &channelInfo, Response &response)

  Returns the overal response from the list of responses in a ChannelInfo.
- **BString** Bds::bdsSelectionChannelInfo (const Selection &selection, **BUInt** channel)

  Returns a string describing the name and time period of a selection channel.
- **BString** Bds::bdsFileNameExpand (**BString** fileName, ChannelInfo &channelInfo)

  Default filename from a ChannelInfo.
- **BString** Bds::bdsFileNameExpand (**BString** fileName, ChannelInfos &channelInfos)

  Default filename from a list of ChannelInfo's.
- double Bds::bdsPoleZeroGain (const PoleZero &poleZero, double frequency)

  Calculates the overal gain of the given PoleZero transfer function.
- void Bds::bdsPoleZeroGainPhase (const PoleZero &poleZero, double frequency, double &gain, double &phase)

  Calculates the overal gain and phase of the given PoleZero transfer function.
- void Bds::bdsPoleZeroToFap (const PoleZero &poleZero, **BUInt** nPoints, double calibrationFrequency, double sampleFrequency, **BArray**< Fap > &fap)

  Convert PoleZero to FAP.
- void Bds::bdsDumpSelection (Selection sel)

  Debug print out a Selection object.
- void Bds::bdsDumpDataInfo (DataInfo dataInfo, int includeInfo=0)

  Debug print out a DataInfo object.
- void Bds::bdsDumpChannelInfos (const ChannelInfos &channelInfos)

  Debug print out a ChannelInfos object.
- void Bds::bdsDumpData (const DataBlock &dataBlock, int nSamples=0)

  Debug print out a DataBlock object.
- void Bds::bdsDumpPoleZeros (PoleZero poleZeros)

  Debug print out a PoleZeros object.

Variables

- const int Bds::NetworkNameLen = 3

  Maximum Network name length.
- const int Bds::StationNameLen = 5

  Maximum Station name length.
- const int Bds::ChannelTypeLen = 3

  Maximum Channel type name length.
- const int Bds::ChannelAuxLen = 2

  Maximum Channel Aux length.
- const int Bds::SourceLen = 16

  Maximum Source length.

8.81.1 Detailed Description

General BdsLib API functions.

8.82 BdsS.cc File Reference

```
#include <BdsC.h>
#include <BdsS.h>
```

Namespaces

- Bds

8.83 BdsS.d File Reference

8.84 BdsT.cc File Reference

```
#include <stdlib.h>
#include <stdint.h>
#include <BdsT.h>
#include <Control.h>
```

8.85 /src/blacknest/bds/bds/doc/bdsApiOverview.dox File Reference

Index