ExSeisPIOL C API

Generated by Doxygen 1.8.11

Contents

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

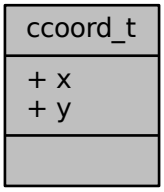
Chapter 3

Data Structure Documentation

3.1 ccoord_t Struct Reference

`#include <cfileapi.h>`

Collaboration diagram for ccoord_t:



Data Fields

- double **x**

  The first coordinate.
- double **y**

  The second coordinate.

3.1.1 Detailed Description

The structure for a coordinate point.

Definition at line 19 of file cfileapi.h.

3.1.2 Field Documentation

3.1.2.1 double x

The first coordinate.

Definition at line 21 of file cfileapi.h.

3.1.2.2 double y

The second coordinate.

Definition at line 22 of file cfileapi.h.

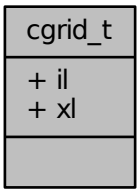
The documentation for this struct was generated from the following file:

- /home/cathal/SSHFS/ExSeisDat/api/**cfileapi.h**

3.2 cgrid_t Struct Reference

`#include <cfileapi.h>`

Collaboration diagram for cgrid_t:



Data Fields

- int64_t **il**

  The first grid value.
- int64_t **xl**

  The second grid value.

3.2.1 Detailed Description

The structure for a grid point.

Definition at line 27 of file cfileapi.h.

3.2.2 Field Documentation

3.2.2.1 int64_t il

The first grid value.

Definition at line 29 of file cfileapi.h.

3.2.2.2 int64_t xl

The second grid value.

Definition at line 30 of file cfileapi.h.

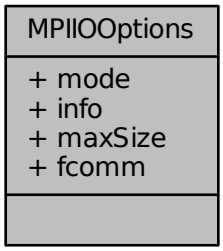
The documentation for this struct was generated from the following file:

- /home/cathal/SSHFS/ExSeisDat/api/**cfileapi.h**

3.3 MPIIOOptions Struct Reference

`#include <cfileapi.h>`

Collaboration diagram for MPIIOOptions:



Data Fields

- enum **Mode mode**

  The file access mode.
- MPI_Info **info**

  The MPI_Info object.
- size_t **maxSize**

  The maximum size to write in an MPI-IO call.
- MPI_Comm **fcomm**

  The MPI communicator which should be used.

3.3.1 Detailed Description

A structure specifying all MPI-IO options

Definition at line 272 of file cfileapi.h.

3.3.2 Field Documentation

3.3.2.1 MPI_Comm fcomm

The MPI communicator which should be used.

Definition at line 277 of file cfileapi.h.

3.3.2.2 MPI_Info info

The MPI_Info object.

Definition at line 275 of file cfileapi.h.

3.3.2.3 size_t maxSize

The maximum size to write in an MPI-IO call.

Definition at line 276 of file cfileapi.h.

3.3.2.4 enum Mode mode

The file access mode.

Definition at line 274 of file cfileapi.h.

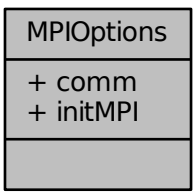
The documentation for this struct was generated from the following file:

- /home/cathal/SSHFS/ExSeisDat/api/**cfileapi.h**

3.4 MPIOptions Struct Reference

`#include <cfileapi.h>`

Collaboration diagram for MPIOptions:



Data Fields

- MPI_Comm **comm**

  The MPI communicator.
- bool **initMPI**

  Whether the PIOL should initialise MPI or not.

3.4.1 Detailed Description

A structure specifying MPI options.

Definition at line 282 of file cfileapi.h.

3.4.2 Field Documentation

3.4.2.1 MPI_Comm comm

The MPI communicator.

Definition at line 284 of file cfileapi.h.

3.4.2.2 bool initMPI

Whether the PIOL should initialise MPI or not.

Definition at line 285 of file cfileapi.h.

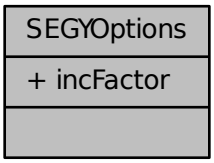
The documentation for this struct was generated from the following file:

- /home/cathal/SSHFS/ExSeisDat/api/**cfileapi.h**

3.5 SEGYOptions Struct Reference

```
#include <cfileapi.h>
```

Collaboration diagram for SEGYOptions:



Data Fields

- double **incFactor**

  The increment factor which should be used with inc.

3.5.1 Detailed Description

A structure specifying file layer options.

Definition at line 290 of file cfileapi.h.

3.5.2 Field Documentation

3.5.2.1 double incFactor

The increment factor which should be used with inc.

Definition at line 292 of file cfileapi.h.

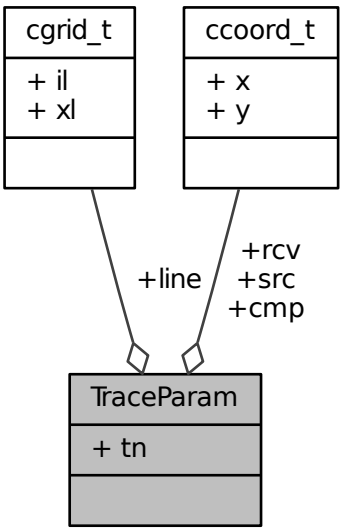
The documentation for this struct was generated from the following file:

- /home/cathal/SSHFS/ExSeisDat/api/**cfileapi.h**

3.6 TraceParam Struct Reference

`#include <cfileapi.h>`

Collaboration diagram for TraceParam:



Data Fields

- **ccoord_t src**
  The Source coordinate point.
- **ccoord_t rcv**
  The Receiver coordinate point.
- **ccoord_t cmp**
  The common midpoint.
- **cgrid_t line**
  The line grid (il, xl).
- size_t **tn**
  The trace number.

3.6.1 Detailed Description

A structure containing all known parameters

Definition at line 51 of file cfileapi.h.

3.6.2 Field Documentation

3.6.2.1 ccoord_t cmp

The common midpoint.

Definition at line 55 of file cfileapi.h.

3.6.2.2 cgrid_t line

The line grid (il, xl).

Definition at line 56 of file cfileapi.h.

3.6.2.3 ccoord_t rcv

The Receiver coordinate point.

Definition at line 54 of file cfileapi.h.

3.6.2.4 ccoord_t src

The Source coordinate point.

Definition at line 53 of file cfileapi.h.

3.6.2.5 size_t tn

The trace number.

Definition at line 57 of file cfileapi.h.

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

- /home/cathal/SSHFS/ExSeisDat/api/**cfileapi.h**

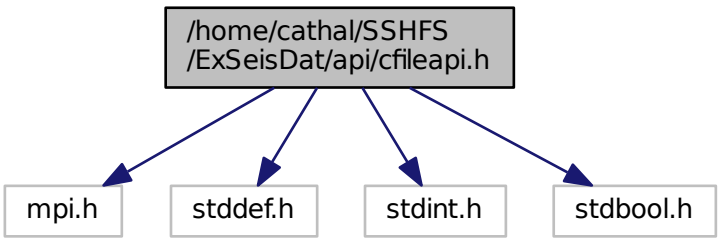
Chapter 4

File Documentation

4.1 /home/cathal/SSHFS/ExSeisDat/api/cfileapi.h File Reference

```
#include <mpi.h>
#include <stddef.h>
#include <stdint.h>
#include <stdbool.h>
```

Include dependency graph for cfileapi.h:



Data Structures

- struct **ccoord_t**
- struct **cgrid_t**
- struct **TraceParam**
- struct **MPIIOOptions**
- struct **MPIOptions**
- struct **SEGYOptions**

Typedefs

- typedef struct PIOLWrapper * **ExSeisHandle**

  A wrapper around a shared PIOL Object.
- typedef struct ExSeisFileWrapper * **ExSeisFile**

  A wrapper around a File Layer pointer.

Enumerations

- enum **CCoord** { **Src**, **Rcv**, **CMP** }
- enum **CGrid** { **Line** }
- enum **Mode** { **ReadMode**, **WriteMode**, **ReadWriteMode** }

Functions

- **ExSeisHandle initMPIOL** (void)
- void **closePIOL** (**ExSeisHandle** piol)
- size_t **getRank** (**ExSeisHandle** piol)
- size_t **getNumRank** (**ExSeisHandle** piol)
- void **isErr** (**ExSeisHandle** piol)
- void **barrier** (**ExSeisHandle** piol)
- size_t **getSEGYTextSz** (void)
- size_t **getSEGYFileSz** (size_t nt, size_t ns)
- size_t **getSEGYTraceLen** (size_t ns)
- size_t **getSEGYParamSz** (void)
- **ExSeisFile openReadFile** (**ExSeisHandle** piol, const char ∗name)
- **ExSeisFile openWriteFile** (**ExSeisHandle** piol, const char ∗name)
- void **closeFile** (**ExSeisFile** fh)

  Close the file associated with the handle.
- const char ∗ **readText** (**ExSeisFile** fh)

  Read the human readable text from the file.
- size_t **readNs** (**ExSeisFile** fh)

  Read the number of samples per trace.
- size_t **readNt** (**ExSeisFile** fh)

  Read the number of traces in the file.
- double **readInc** (**ExSeisFile** fh)

  Read the increment between trace samples.
- void **writeText** (**ExSeisFile** fh, const char ∗text)

  Write the human readable text from the file.
- void **writeNs** (**ExSeisFile** fh, size_t ns)

  Write the number of samples per trace.
- void **writeNt** (**ExSeisFile** fh, size_t nt)

  Write the number of traces in the file.
- void **writeInc** (**ExSeisFile** fh, double inc)

  Write the increment between trace samples.
- void **readCoordPoint** (**ExSeisFile** fh, **CCoord** item, size_t offset, size_t sz, **ccoord_t** ∗buf)

  Read coordinate pairs from the ith-trace to i+sz.
- void **readGridPoint** (**ExSeisFile** fh, **CGrid** item, size_t offset, size_t sz, **cgrid_t** ∗buf)

  Read grid pairs from the ith-trace to i+sz.
- void **writeTraceParam** (**ExSeisFile** fh, size_t offset, size_t sz, const **TraceParam** ∗prm)

  Write the trace parameters from offset to offset+sz to the respective trace headers.
- void **readTraceParam** (**ExSeisFile** fh, size_t offset, size_t sz, **TraceParam** ∗prm)

  Write the trace parameters from offset to offset+sz to the respective trace headers.
- void **readTrace** (**ExSeisFile** fh, size_t offset, size_t sz, float ∗trace)

  Read the trace's from offset to offset+sz.
- void **writeTrace** (**ExSeisFile** fh, size_t offset, size_t sz, float ∗trace)

  Read the trace's from offset to offset+sz.
- **ExSeisHandle initPIOL** (size_t logLevel, **MPIOptions** ∗mpiOpt)
- **ExSeisFile openFile** (**ExSeisHandle** piol, const char ∗name, **SEGYOptions** ∗opt, **MPIIOOptions** ∗mpiio↩ Opt)

4.1.1 Typedef Documentation

4.1.1.1 typedef struct ExSeisFileWrapper* ExSeisFile

A wrapper around a File Layer pointer.

Definition at line 12 of file cfileapi.h.

4.1.1.2 typedef struct PIOLWrapper* ExSeisHandle

A wrapper around a shared PIOL Object.

Definition at line 11 of file cfileapi.h.

4.1.2 Enumeration Type Documentation

4.1.2.1 enum CCoord

The options for various coordinate points associated with a trace.

Enumerator

Src The source coordinate point.

Rcv The receiver coordinate point.

CMP The common-midpoint.

Definition at line 35 of file cfileapi.h.

```
36 {
37     Src,
38     Rcv,
39     CMP
40 } CCoord;
```

4.1.2.2 enum CGrid

The options for grid points associated with a trace.

Enumerator

Line The only current option for a grid. The inline/crossline pair.

Definition at line 44 of file cfileapi.h.

```
45 {
46     Line
47 } CGrid;
```

4.1.2.3 enum Mode

A list of the different modes of file access.

Enumerator

ReadMode The file is opened as read only.

WriteMode The file is opened as write only.

ReadWriteMode The file is opened as Read/Write.

Definition at line 263 of file cfileapi.h.

```
264 {
265     ReadMode,
266     WriteMode,
267     ReadWriteMode
268 };
```

4.1.3 Function Documentation

4.1.3.1 void barrier (ExSeisHandle *piol*)

A barrier. All PIOL processes must call this.

Parameters

in	*piol*	A handle to the PIOL.

4.1.3.2 void closeFile (ExSeisFile *fh*)

Close the file associated with the handle.

Parameters

in	*fh*	A handle for the file.

4.1.3.3 void closePIOL (ExSeisHandle *piol*)

close the PIOL (deinit MPI)

Parameters

in	*piol*	A handle to the PIOL.

4.1.3.4 size_t getNumRank (ExSeisHandle *piol*)

Get the number of processes (in terms of the PIOL communicator)

Parameters

in	*piol*	A handle to the PIOL.

4.1.3.5 size_t getRank (ExSeisHandle *piol*)

Get the rank of the process (in terms of the PIOL communicator)

Parameters

in	*piol*	A handle to the PIOL.

4.1.3.6 size_t getSEGYFileSz (size_t *nt,* size_t *ns*)

Get the size a SEGY file should be given the number of traces (nt) and sample size (ns)

Parameters

in	*nt*	The number of traces
in	*ns*	The number of samples per trace

Returns

The corresponding file size in bytes for SEG-Y

4.1.3.7 size_t getSEGYParamSz (void)

Get the size of a SEGY trace header

Returns

The trace header size in bytes

4.1.3.8 size_t getSEGYTextSz (void)

Get the size of the SEG-Y text field (3200 bytes)

Returns

The text size in bytes for SEG-Y

4.1.3.9 size_t getSEGYTraceLen (size_t *ns*)

Get the size a SEGY trace should be given the sample size (ns) and a type of float

Parameters

in	*ns*	The number of samples per trace

Returns

The corresponding trace size in bytes

4.1.3.10 ExSeisHandle initMPIOL (void)

Initialise the PIOL and MPI

Returns

A handle to the PIOL.

4.1.3.11 ExSeisHandle initPIOL (size_t *logLevel,* MPIOptions * *mpiOpt*)

Initialise the PIOL and optionally MPI.

Parameters

in	*logLevel*	The log level
in	*mpiOpt*	The MPI options structure

Returns

A handle to the PIOL.

4.1.3.12 void isErr (ExSeisHandle *piol*)

Check if the PIOL has any error conditions

Parameters

in	*piol*	A handle to the PIOL.

4.1.3.13 ExSeisFile openFile (ExSeisHandle *piol,* const char * *name,* SEGYOptions * *opt,* MPIIOOptions * *mpiioOpt*)

Open a file and return a handle for the file

Parameters

in	*piol*	A handle to the PIOL.
in	*name*	The name of the file.
in	*opt*	The SEG-Y options structure
in	*mpiioOpt*	The MPI-IO options structure

Returns

A handle for the file.

4.1.3.14 ExSeisFile openReadFile (ExSeisHandle *piol*, const char * *name*)

Open a read-only file and return a handle for the file

Parameters

in	*piol*	A handle to the PIOL.
in	*name*	The name of the file.

Returns

A handle for the file.

4.1.3.15 ExSeisFile openWriteFile (ExSeisHandle *piol*, const char * *name*)

Open a write-only file and return a handle for the file

Parameters

in	*piol*	A handle to the PIOL.
in	*name*	The name of the file.

Returns

A handle for the file.

4.1.3.16 void readCoordPoint (ExSeisFile *fh*, CCoord *item*, size_t *offset*, size_t *sz*, ccoord_t * *buf*)

Read coordinate pairs from the ith-trace to i+sz.

Parameters

in	*fh*	A handle for the file.
in	*item*	The coordinate pair of interest.
in	*offset*	The starting trace number.
in	*sz*	The number of traces to process.
out	*buf*	The buffer which is sizeof(coord_t)*sz long.

4.1.3.17 void readGridPoint (ExSeisFile *fh*, CGrid *item*, size_t *offset*, size_t *sz*, cgrid_t * *buf*)

Read grid pairs from the ith-trace to i+sz.

Parameters

in	*fh*	A handle for the file.
in	*item*	The grid pair of interest.
in	*offset*	The starting trace number.
in	*sz*	The number of traces to process.
out	*buf*	The buffer which is sizeof(grid_t)*sz long.

4.1.3.18 double readInc (ExSeisFile *fh*)

Read the increment between trace samples.

Parameters

in	*fh*	A handle for the file.

Returns

The increment between trace samples

4.1.3.19 size_t readNs (ExSeisFile *fh*)

Read the number of samples per trace.

Parameters

in	*fh*	A handle for the file.

Returns

The number of samples per trace

4.1.3.20 size_t readNt (ExSeisFile *fh*)

Read the number of traces in the file.

Parameters

in	*fh*	A handle for the file.

Returns

The number of traces

4.1.3.21 const char* readText (ExSeisFile *fh*)

Read the human readable text from the file.

When readText is called the ExSeisPIOL is responsible for the memory returned. The string should not be dereferenced after the associated file is closed.

Parameters

in	*fh*	A handle for the file.

Returns

A string containing the text (in ASCII format)

4.1.3.22 void readTrace (ExSeisFile *fh*, size_t *offset*, size_t *sz*, float * *trace*)

Read the trace's from offset to offset+sz.

Parameters

in	*fh*	A handle for the file.
in	*offset*	The starting trace number.
in	*sz*	The number of traces to process
out	*trace*	A contiguous array of each trace (size sz*ns*sizeof(trace_t))

4.1.3.23 void readTraceParam (ExSeisFile *fh*, size_t *offset*, size_t *sz*, TraceParam * *prm*)

Write the trace parameters from offset to offset+sz to the respective trace headers.

Parameters

in	*fh*	A handle for the file.
in	*offset*	The starting trace number.
in	*sz*	The number of traces to process.
in	*prm*	An array of the parameter structures (size sizeof(TraceParam)*sz)

It is assumed that this operation is not an update. Any previous contents of the trace header will be overwritten.

4.1.3.24 void writeInc (ExSeisFile *fh*, double *inc*)

Write the increment between trace samples.

Parameters

in	*fh*	A handle for the file.
in	*inc*	The new increment between trace samples.

4.1.3.25 void writeNs (ExSeisFile *fh*, size_t *ns*)

Write the number of samples per trace.

Parameters

in	*fh*	A handle for the file.
in	*ns*	The new number of samples per trace.

4.1.3.26 void writeNt (ExSeisFile *fh*, size_t *nt*)

Write the number of traces in the file.

Parameters

in	*fh*	A handle for the file.
in	*nt*	The new number of traces.

4.1.3.27 void writeText (ExSeisFile *fh*, const char * *text*)

Write the human readable text from the file.

Parameters

in	*fh*	A handle for the file.
in	*text*	The new null-terminated string containing the text (in ASCII format).

4.1.3.28 void writeTrace (ExSeisFile *fh*, size_t *offset*, size_t *sz*, float * *trace*)

Read the trace's from offset to offset+sz.

Parameters

in	*fh*	A handle for the file.
in	*offset*	The starting trace number.
in	*sz*	The number of traces to process
out	*trace*	A contiguous array of each trace (size sz∗ns∗sizeof(trace_t))

Warning

This function is not thread safe.

4.1.3.29 void writeTraceParam (ExSeisFile *fh*, size_t *offset*, size_t *sz*, const TraceParam * *prm*)

Write the trace parameters from offset to offset+sz to the respective trace headers.

Parameters

in	*fh*	A handle for the file.
in	*offset*	The starting trace number.
in	*sz*	The number of traces to process.
in	*prm*	An array of the parameter structures (size sizeof(TraceParam)*sz)

It is assumed that this operation is not an update. Any previous contents of the trace header will be overwritten.

Index