netmd++

1.0.1

Generated by Doxygen 1.9.1

Chapter 1

Main Page

1.1 netmd++

This C++ API was written to ease the handling of NetMD devices. It is a synchronous API. So, function calls might block your program flow. If you want to use this API in an GUI app, better put the API calls into a background thread.

1.1.1 Namespace

This API uses the namespace *netmd*.

1.1.2 Usage

- include the header file into your project:
  ```
  #include "path/to/netmd++.h"
  ```
- create an instance of the API:
  ```
  netmd::netmd_pp* pNetMd = new netmd::netmd_pp();
  ```
- initialize the first found NetMD device:
  ```
  if (pNetMd != nullptr)
  {
      pNetMd->initDevice();
  }
  ```
- If you change or re-plug the device, simply run above code (initDevice()) again!

1.1.3 Examples

1.1.3.1 Track transfer

Check for on-the-fly support and transfer a WAVE file to NetMD with on-the-fly encoding (LP2) or w/o encoding (SP).

```
#include <netmd++.h>
int main()
{
    netmd::netmd_pp* pNetMd = new netmd::netmd_pp();
    if ((pNetMd != nullptr) && (pNetMd->initDevice() == netmd::NETMDERR_NO_ERROR))
    {
        if (pNetMd->otfEncodeSupported())
        {
            pNetMd->sendAudioFile("/path/to/nice/audio.wav", "Very nice Audio file (LP2)",
       netmd::NETMD_DISKFORMAT_LP2);
        }
        else
        {
            pNetMd->sendAudioFile("/path/to/nice/audio.wav", "Very nice Audio file (SP)",
       netmd::NO_ONTHEFLY_CONVERSION);
        }
    }
    return 0;
}
```

1.1.3.2 Erase disc and set new title

```
#include <netmd++.h>
int main()
{
    netmd::netmd_pp* pNetMd = new netmd::netmd_pp();
    if ((pNetMd != nullptr) && (pNetMd->initDevice() == netmd::NETMDERR_NO_ERROR))
    {
        pNetMd->eraseDisc();
        pNetMd->setDiscTitle("Amazing MD");
    }
    return 0;
}
```

1.2 MDs UTOC

For the UTOC structure please have a look at this great site on minidisc.org

1.2.1 Addressing in UTOC

"The disc start and end addresses each consist of a cluster, sector, and sound group, all packed into 3 bytes. The sound group is the MiniDisc's smallest addressable unit, representing 11.6ms of mono audio (212 bytes). A sector contains 11 sound groups (2332 bytes). A cluster is an aggregate of 32 sectors (352 sound groups) representing 2.03 seconds of stereo audio; it is the smallest unit of data that can be written to a MiniDisc. In the 3 byte packing, there are 14 bits allocated to the cluster number, 6 bits to the sector, and 4 bits to the soundgroup; this arrangement allows addressing of up to 9.2 hours of stereo audio."

1.2.2 Modifying the UTOC

1. download the UTOC sectors 0 ... 2 from NetMD Device:
```
pNetMd->prepareTOCManip();
NetMDByteVector tocData;
for (int i = 0; i < 3; i++)
{
    tocData += pNetMd->readUTOCSector(static_cast<UTOCSector>(i));
}
```
2. create toc class instance and add some track data
```
uint8_t *pData = new uint8_t[tocData.size()];
for(size_t i = 0; i < tocData.size(); i++)
{
    pData[i] = toc.at(i);
}
netmd::CNetMdTOC utoc(8, 459'000, pData);
utoc.addTrack(1, 60'000, "Funky Track One Minute Part #1");
utoc.addTrack(2, 60'000, "Funky Track One Minute Part #2");
```
3. upload changed TOC data to NetMD
```
bool doit = true;
for (int x = 0; x < 3; x++)
{
    tocData.clear();
    addArrayData(tocData, &pData[2352 * x], 2352);
    if (pNetMD->writeUTOCSector(static_cast<UTOCSector>(x), tocData) == NETMDERR_NO_ERROR)
    {
        std::cout « "TOC sector " « x « " written!" « std::endl;
    }
    else
    {
        doit = false;
    }
}
if (doit)
{
    pNetMD->finalizeTOC();
}
delete [] pData;
```

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 netmd Namespace Reference

Namespaces

- toc

Classes

- struct TrackTime

  track times
- struct NetMdTime

  NetMD time.
- struct DiscCapacity

  Structure to hold the capacity information of a disc.
- struct Group

  track group
- class CNetMdApi

  This class describes a C++ NetMD access library.
- class CNetMdTOC

  This class describes a net md TOC.

Typedefs

- using Groups = std::vector< Group >

  netmd groups
- using NetMDByteVector = std::vector< uint8_t >

  byte vector
- using netmd_pp = CNetMdApi

  use netmd_pp instead of CNetMdApi

Enumerations

- enum DiskFormat : uint8_t {
  NETMD_DISKFORMAT_LP4 = 0 , NETMD_DISKFORMAT_LP2 = 2 , NETMD_DISKFORMAT_SP_MONO = 4 , NETMD_DISKFORMAT_SP_STEREO = 6 ,
  NO_ONTHEFLY_CONVERSION = 0xf }

  disk format
- enum NetMdErr : int {
  NETMDERR_NO_ERROR = 0 , NETMDERR_USB = -1 , NETMDERR_NOTREADY = -2 , NETMDERR_TIMEOUT = -3 ,
  NETMDERR_CMD_FAILED = -4 , NETMDERR_CMD_INVALID = -5 , NETMDERR_PARAM = -6 , NETMDERR_OTHER = -7 ,
  NETMDERR_NOT_SUPPORTED = -8 , NETMDERR_INTERIM = -9 }

  NetMD errors.
- enum class TrackProtection : uint8_t { UNPROTECTED = 0x00 , PROTECTED = 0x03 , UNKNOWN = 0xFF }

  type safe protection flags
- enum class AudioEncoding : uint8_t { SP = 0x90 , LP2 = 0x92 , LP4 = 0x93 , UNKNOWN = 0xff }

  type safe encoding flags
- enum typelog {
  DEBUG , INFO , WARN , CRITICAL ,
  CAPTURE }

  log severity
- enum UTOCSector : uint16_t {
  POS_ADDR , HW_TITLES , TSTAMPS , FW_TITLES ,
  UNKNWN_1 , UNKNON_2 }

  TOC sector names.

Functions

- std::ostream & operator<< (std::ostream &o, const TrackTime &tt)

  format helper for TrackTime
- std::ostream & operator<< (std::ostream &o, const AudioEncoding &ae)

  format helper for AudioEncoding
- std::ostream & operator<< (std::ostream &o, const TrackProtection &tp)

  format helper for TrackProtection
- NetMDByteVector & operator+= (NetMDByteVector &a, const NetMDByteVector &b)

  Addition assignment operator for NetMDByteVector.

5.1.1 Typedef Documentation

5.1.1.1 Groups

```
using netmd::Groups = typedef std::vector<Group>
```

netmd groups

5.1.1.2 netmd_pp

```
using netmd::netmd_pp = typedef CNetMdApi
```

use netmd_pp instead of CNetMdApi

5.1.1.3 NetMDByteVector

```
using netmd::NetMDByteVector = typedef std::vector<uint8_t>
```

byte vector

5.1.2 Enumeration Type Documentation

5.1.2.1 AudioEncoding

```
enum netmd::AudioEncoding : uint8_t [strong]
```

type safe encoding flags

Enumerator

SP	SP encoding.
LP2	LP2 encoding.
LP4	LP4 encoding.
UNKNOWN	unknown encoding

5.1.2.2 DiskFormat

```
enum netmd::DiskFormat : uint8_t
```

disk format

Enumerator

NETMD_DISKFORMAT_LP4	LP4.
NETMD_DISKFORMAT_LP2	LP2.
NETMD_DISKFORMAT_SP_MONO	SP mono.
NETMD_DISKFORMAT_SP_STEREO	SP stereo.
NO_ONTHEFLY_CONVERSION	dont do on-the-fly encoding

5.1.2.3 NetMdErr

`enum netmd::NetMdErr : int`

NetMD errors.

Enumerator

NETMDERR_NO_ERROR	success
NETMDERR_USB	general USB error
NETMDERR_NOTREADY	player not ready for command
NETMDERR_TIMEOUT	timeout while waiting for response
NETMDERR_CMD_FAILED	minidisc responded with 08 response
NETMDERR_CMD_INVALID	minidisc responded with 0A response
NETMDERR_PARAM	parameter error
NETMDERR_OTHER	any other error
NETMDERR_NOT_SUPPORTED	not supported
NETMDERR_INTERIM	interim

5.1.2.4 TrackProtection

`enum netmd::TrackProtection : uint8_t [strong]`

type safe protection flags

Enumerator

UNPROTECTED	track is unprotected
PROTECTED	track is protected
UNKNOWN	unknown track state

5.1.2.5 typelog

`enum netmd::typelog`

log severity

Enumerator

DEBUG	debug information
INFO	information
WARN	more serious
CRITICAL	critical information
CAPTURE	needed for log parcing!

5.1.2.6 UTOCSector

```
enum netmd::UTOCSector :  uint16_t
```

TOC sector names.

Enumerator

POS_ADDR	position and addresses of audio data
HW_TITLES	half width titles
TSTAMPS	time stamps
FW_TITLES	full width titles
UNKNWN_1	some unidentified TOC sector #1
UNKNON_2	some unidentified TOC sector #2

5.1.3 Function Documentation

5.1.3.1 operator+=()

```
NetMDByteVector& netmd::operator+= (
            NetMDByteVector & a,
            const NetMDByteVector & b )
```

Addition assignment operator for NetMDByteVector.

Parameters

	a	byte vector 1
in	*b*	byte vector 2

Returns

The result of the addition assignment

5.1.3.2 operator<<() [1/3]

```
std::ostream& netmd::operator<< (
            std::ostream & o,
            const AudioEncoding & ae )
```

format helper for AudioEncoding

Parameters

	o	ref. to ostream
in	*ae*	AudioEncoding

Returns

formatted AudioEncoding stored in ostream

5.1.3.3 operator<<() [2/3]

```
std::ostream& netmd::operator<< (
            std::ostream & o,
            const TrackProtection & tp )
```

format helper for TrackProtection

Parameters

	o	ref. to ostream
in	*tp*	TrackProtection

Returns

formatted TrackProtection stored in ostream

5.1.3.4 operator<<() [3/3]

```
std::ostream& netmd::operator<< (
            std::ostream & o,
            const TrackTime & tt )
```

format helper for TrackTime

Parameters

	o	ref. to ostream
in	*tt*	TrackTime

Returns

formatted TrackTime stored in ostream

5.2 netmd::toc Namespace Reference

Chapter 6

Class Documentation

6.1 netmd::CNetMdApi Class Reference

This class describes a C++ NetMD access library.

`#include <netmd++.h>`

Public Member Functions

- CNetMdApi ()
  Constructs a new instance.
- ~CNetMdApi ()
  Destroys the object.
- int initDevice ()
  Initializes the device.
- std::string getDeviceName () const
  Gets the device name.
- int trackCount ()
  request track count
- int discFlags ()
  request disc flags
- int eraseDisc ()
  erase MD
- int trackTime (int trackNo, TrackTime &trackTime)
  get track time
- int discTitle (std::string &title)
  get disc title
- int setDiscTitle (const std::string &title)
  Sets the disc title.
- int moveTrack (uint16_t from, uint16_t to)
  move a track (number)
- int setGroupTitle (uint16_t group, const std::string &title)
  Sets the group title.
- int createGroup (const std::string &title, int first, int last)
  Creates a group.

- int addTrackToGroup (int track, int group)

  Adds a track to group.
- int delTrackFromGroup (int track, int group)

  remove track from group
- int deleteGroup (int group)

  delete a group
- int deleteTrack (uint16_t track)

  delete track
- int trackBitRate (uint16_t track, AudioEncoding &encoding, uint8_t &channel)

  get track bitrate data
- int trackFlags (uint16_t track, TrackProtection &flags)

  get track flags
- int trackTitle (uint16_t track, std::string &title)

  get track title
- bool spUploadSupported ()

  is SP upload supported?
- bool otfEncodeSupported ()

  is on the fly encoding supported by device
- bool tocManipSupported ()

  is TOC manipulation supported?
- int sendAudioFile (const std::string &filename, const std::string &title, DiskFormat otf)

  Sends an audio track.
- int setTrackTitle (uint16_t trackNo, const std::string &title)

  Sets the track title.
- int discCapacity (DiscCapacity &dcap)

  get disc capacity
- Groups groups ()

  get MD track groups
- int prepareTOCManip ()

  prepare TOC manipulation
- NetMDByteVector readUTOCSector (UTOCSector s)

  Reads an utoc sector.
- int writeUTOCSector (UTOCSector s, const NetMDByteVector &data)

  Writes an utoc sector.
- int finalizeTOC (uint8_t resetWait=15)

  finalize TOC through exploit

Static Public Member Functions

- static void setLogLevel (int severity)

  Sets the log level.
- static void setLogStream (std::ostream &os)

  Sets the log stream.

6.1.1 Detailed Description

This class describes a C++ NetMD access library.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 CNetMdApi()

```
netmd::CNetMdApi::CNetMdApi ( )
```

Constructs a new instance.

6.1.2.2 ~CNetMdApi()

```
netmd::CNetMdApi::~CNetMdApi ( )
```

Destroys the object.

6.1.3 Member Function Documentation

6.1.3.1 addTrackToGroup()

```
int netmd::CNetMdApi::addTrackToGroup (
            int track,
            int group )
```

Adds a track to group.

Parameters

in	*track*	The track
in	*group*	The group

Returns

NetMdErr

6.1.3.2 createGroup()

```
int netmd::CNetMdApi::createGroup (
            const std::string & title,
            int first,
            int last )
```

Creates a group.

Parameters

in	*title*	The title
in	*first*	The first track
in	*last*	The last track

Returns

NetMdErr

6.1.3.3 deleteGroup()

```
int netmd::CNetMdApi::deleteGroup (
            int group )
```

delete a group

Parameters

in	*group*	The group

Returns

NetMdErr

6.1.3.4 deleteTrack()

```
int netmd::CNetMdApi::deleteTrack (
            uint16_t track )
```

delete track

Parameters

in	*track*	The track number

Returns

NetMdErr

6.1.3.5 delTrackFromGroup()

```
int netmd::CNetMdApi::delTrackFromGroup (
            int track,
            int group )
```

remove track from group

Parameters

in	*track*	The track
in	*group*	The group

Returns

NetMdErr

6.1.3.6 discCapacity()

```
int netmd::CNetMdApi::discCapacity (
            DiscCapacity & dcap )
```

get disc capacity

Parameters

out	*dcap*	The buffer for disc capacity

Returns

NetMdErr

6.1.3.7 discFlags()

```
int netmd::CNetMdApi::discFlags ( )
```

request disc flags

Returns

< 0 -> NetMdErr; else -> flags

6.1.3.8 discTitle()

```
int netmd::CNetMdApi::discTitle (
            std::string & title )
```

get disc title

Parameters

out	*title*	The title

Returns

NetMdErr

6.1.3.9 eraseDisc()

```
int netmd::CNetMdApi::eraseDisc ( )
```

erase MD

Returns

NetMdErr

6.1.3.10 finalizeTOC()

```
int netmd::CNetMdApi::finalizeTOC (
            uint8_t resetWait = 15 )
```

finalize TOC through exploit

Parameters

in	*resetWait*	The optional reset wait time (15 seconds)

Returns

NetMdErr

See also

NetMdErr

6.1.3.11 getDeviceName()

```
std::string netmd::CNetMdApi::getDeviceName ( ) const
```

Gets the device name.

Returns

The device name.

6.1.3.12 groups()

```
Groups netmd::CNetMdApi::groups ( )
```

get MD track groups

Returns

vector of group structures

6.1.3.13 initDevice()

```
int netmd::CNetMdApi::initDevice ( )
```

Initializes the device.

Returns

NetMdErr

6.1.3.14 moveTrack()

```
int netmd::CNetMdApi::moveTrack (
            uint16_t from,
            uint16_t to )
```

move a track (number)

Parameters

in	*from*	from position
in	*to*	to position

Returns

NetMdErr

6.1.3.15 otfEncodeSupported()

```
bool netmd::CNetMdApi::otfEncodeSupported ( )
```

is on the fly encoding supported by device

Returns

true if so

6.1.3.16 prepareTOCManip()

```
int netmd::CNetMdApi::prepareTOCManip ( )
```

prepare TOC manipulation

Returns

NetMdErr

6.1.3.17 readUTOCSector()

```
NetMDByteVector netmd::CNetMdApi::readUTOCSector (
            UTOCSector s )
```

Reads an utoc sector.

Parameters

in	*s*	sector number

Returns

TOC sector data. (error if empty)

6.1.3.18 sendAudioFile()

```
int netmd::CNetMdApi::sendAudioFile (
            const std::string & filename,
            const std::string & title,
            DiskFormat otf )
```

Sends an audio track.

The audio file must be either an WAVE file (44.1kHz / 16 bit), or an pre-encoded atrac3 file with a WAVE header. If your device supports on-the-fly encoding (not common), you can set the DiskFormat to NETMD_DISKFORMAT_LP4 or NETMD_DISKFORMAT_LP2. If you want best audio quality, use NO_ONTHEFLY_CONVERSION.

In case your device supports the SP download through Sony Firmware exploit, the input file might be a plain atrac 1 file.

Parameters

in	*filename*	The filename
in	*title*	The title
in	*otf*	The disk format

Returns

NetMdErr

6.1.3.19 setDiscTitle()

```
int netmd::CNetMdApi::setDiscTitle (
            const std::string & title )
```

Sets the disc title.

Parameters

in	*title*	The title

Returns

NetMdErr

6.1.3.20 setGroupTitle()

```
int netmd::CNetMdApi::setGroupTitle (
            uint16_t group,
            const std::string & title )
```

Sets the group title.

Parameters

in	*group*	The group
in	*title*	The title

Returns

NetMdErr

6.1.3.21 setLogLevel()

```
static void netmd::CNetMdApi::setLogLevel (
            int severity )  [static]
```

Sets the log level.

Parameters

in	*severity*	The severity

6.1.3.22 setLogStream()

```
static void netmd::CNetMdApi::setLogStream (
            std::ostream & os )  [static]
```

Sets the log stream.

Parameters

in	*os*	The stream instance to log to

6.1.3.23 setTrackTitle()

```
int netmd::CNetMdApi::setTrackTitle (
            uint16_t trackNo,
            const std::string & title )
```

Sets the track title.

Parameters

in	*trackNo*	The track no
in	*title*	The title

Returns

NetMdErr

6.1.3.24 spUploadSupported()

```
bool netmd::CNetMdApi::spUploadSupported ( )
```

is SP upload supported?

Returns

true if yes

6.1.3.25 tocManipSupported()

```
bool netmd::CNetMdApi::tocManipSupported ( )
```

is TOC manipulation supported?

Returns

true if supported, false if not

6.1.3.26 trackBitRate()

```
int netmd::CNetMdApi::trackBitRate (
            uint16_t track,
            AudioEncoding & encoding,
            uint8_t & channel )
```

get track bitrate data

Parameters

in	*track*	The track number
out	*encoding*	The encoding flag
out	*channel*	The channel flag

Returns

NetMdErr

6.1.3.27 trackCount()

```
int netmd::CNetMdApi::trackCount ( )
```

request track count

Returns

< 0 -> NetMdErr; else -> track count

6.1.3.28 trackFlags()

```
int netmd::CNetMdApi::trackFlags (
            uint16_t track,
            TrackProtection & flags )
```

get track flags

Parameters

in	*track*	The track number
out	*flags*	The track flags

Returns

NetMdErr

6.1.3.29 trackTime()

```
int netmd::CNetMdApi::trackTime (
            int trackNo,
            TrackTime & trackTime )
```

get track time

Parameters

in	*trackNo*	The track no
out	*trackTime*	The track time

Returns

NetMdErr

6.1.3.30 trackTitle()

```
int netmd::CNetMdApi::trackTitle (
            uint16_t track,
            std::string & title )
```

get track title

Parameters

in	*track*	The track number
out	*title*	The track title

Returns

NetMdErr

6.1.3.31 writeUTOCSector()

```
int netmd::CNetMdApi::writeUTOCSector (
            UTOCSector s,
            const NetMDByteVector & data )
```

Writes an utoc sector.

Parameters

in	*s*	sector names
in	*data*	The data to be written

Returns

NetMdErr

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

- /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h

6.2 netmd::CNetMdTOC Class Reference

This class describes a net md TOC.

```
#include <netmd++.h>
```

Classes

- struct DAOFragment

  a fragment used in DAO track

Public Types

- using DAOFragments = std::vector< DAOFragment >

  type to store all DAO track fragments (for fragmented, non empty discs)

Public Member Functions

- CNetMdTOC (int trackCount=0, uint32_t lenInMs=0, uint8_t *data=nullptr)

  Constructs a new instance.
- ~CNetMdTOC ()

  Destroys the object.
- void import (int trackCount=0, uint32_t lenInMs=0, uint8_t *data=nullptr)

  import TOC data
- int addTrack (uint8_t no, uint32_t lengthMs, const std::string &title)

  Adds a track.
- int setDiscTitle (const std::string &title)

  Sets the disc title.
- int trackCount () const

  get track count
- std::string discTitle () const

  get MD title
- std::string trackTitle (int trackNo) const

  get track title
- std::string trackInfo (int trackNo) const

  get track info
- std::string discInfo () const

  get disc info

6.2.1 Detailed Description

This class describes a net md TOC.

6.2.2 Member Typedef Documentation

6.2.2.1 DAOFragments

```
using netmd::CNetMdTOC::DAOFragments = std::vector<DAOFragment>
```

type to store all DAO track fragments (for fragmented, non empty discs)

6.2.3 Constructor & Destructor Documentation

6.2.3.1 CNetMdTOC()

```
netmd::CNetMdTOC::CNetMdTOC (
            int trackCount = 0,
            uint32_t lenInMs = 0,
            uint8_t * data = nullptr )
```

Constructs a new instance.

Parameters

in	*trackCount*	The track count
in	*lenInMs*	The length in milliseconds
	data	The TOC data

6.2.3.2 ~CNetMdTOC()

```
netmd::CNetMdTOC::~CNetMdTOC ( )
```

Destroys the object.

6.2.4 Member Function Documentation

6.2.4.1 addTrack()

```
int netmd::CNetMdTOC::addTrack (
            uint8_t no,
            uint32_t lengthMs,
            const std::string & title )
```

Adds a track.

This function has to be used to split a DAO transferred disc audio track into the parts as on the original disc. This functions has to be called for all tracks in their correct order! **Breaking the order will break the TOC!**

Parameters

in	*no*	track number (starting with 1)
in	*lengthMs*	The length in milliseconds
in	*title*	The track title

Returns

0 -> ok; -1 -> error

6.2.4.2 discInfo()

```
std::string netmd::CNetMdTOC::discInfo ( ) const
```

get disc info

Returns

disc info

6.2.4.3 discTitle()

```
std::string netmd::CNetMdTOC::discTitle ( ) const
```

get MD title

Returns

title

6.2.4.4 import()

```
void netmd::CNetMdTOC::import (
            int trackCount = 0,
            uint32_t lenInMs = 0,
            uint8_t * data = nullptr )
```

import TOC data

Parameters

in	*trackCount*	The track count
in	*lenInMs*	The length in milliseconds
	data	The TOC data

6.2.4.5 setDiscTitle()

```
int netmd::CNetMdTOC::setDiscTitle (
            const std::string & title )
```

Sets the disc title.

Parameters

in	*title*	The title

Returns

0 -> ok; -1 -> error

6.2.4.6 trackCount()

```
int netmd::CNetMdTOC::trackCount ( ) const
```

get track count

Returns

number of tracks

6.2.4.7 trackInfo()

```
std::string netmd::CNetMdTOC::trackInfo (
            int trackNo ) const
```

get track info

Parameters

in	*trackNo*	The track number

Returns

track info

6.2.4.8 trackTitle()

```
std::string netmd::CNetMdTOC::trackTitle (
            int trackNo ) const
```

get track title

Parameters

in	*trackNo*	The track number

Returns

title

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

- /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h

6.3 netmd::CNetMdTOC::DAOFragment Struct Reference

a fragment used in DAO track

`#include <netmd++.h>`

Public Attributes

- uint32_t mStart
  start group
- uint32_t mEnd
  end group

6.3.1 Detailed Description

a fragment used in DAO track

6.3.2 Member Data Documentation

6.3.2.1 mEnd

```
uint32_t netmd::CNetMdTOC::DAOFragment::mEnd
```

end group

6.3.2.2 mStart

```
uint32_t netmd::CNetMdTOC::DAOFragment::mStart
```

start group

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

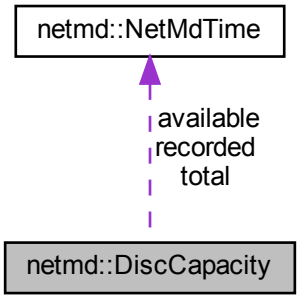
- /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h

6.4 netmd::DiscCapacity Struct Reference

Structure to hold the capacity information of a disc.

```
#include <netmd++.h>
```

Collaboration diagram for netmd::DiscCapacity:



Public Attributes

- NetMdTime recorded

  Time allready recorded on the disc.
- NetMdTime total
- NetMdTime available

6.4.1 Detailed Description

Structure to hold the capacity information of a disc.

6.4.2 Member Data Documentation

6.4.2.1 available

```
NetMdTime netmd::DiscCapacity::available
```

Time that is available on the disc. This depends on the current recording settings.

6.4.2.2 recorded

```
NetMdTime netmd::DiscCapacity::recorded
```

Time allready recorded on the disc.

6.4.2.3 total

```
NetMdTime netmd::DiscCapacity::total
```

Total time, that could be recorded on the disc. This depends on the current recording settings.

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

- /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h

6.5 netmd::Group Struct Reference

track group

```
#include <netmd++.h>
```

Public Attributes

- int mGid
  group id
- int16_t mFirst
  first track
- int16_t mLast
  last track
- std::string mName
  group name

6.5.1 Detailed Description

track group

6.5.2 Member Data Documentation

6.5.2.1 mFirst

```
int16_t netmd::Group::mFirst
```

first track

6.5.2.2 mGid

```
int netmd::Group::mGid
```

group id

6.5.2.3 mLast

```
int16_t netmd::Group::mLast
```

last track

6.5.2.4 mName

```
std::string netmd::Group::mName
```

group name

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

- /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h

6.6 netmd::NetMdTime Struct Reference

NetMD time.

```
#include <netmd++.h>
```

Public Attributes

- uint16_t hour
  hour
- uint8_t minute
  minute
- uint8_t second
  second
- uint8_t frame
  frame

6.6.1 Detailed Description

NetMD time.

6.6.2 Member Data Documentation

6.6.2.1 frame

```
uint8_t netmd::NetMdTime::frame
```

frame

6.6.2.2 hour

```
uint16_t netmd::NetMdTime::hour
```

hour

6.6.2.3 minute

```
uint8_t netmd::NetMdTime::minute
```

minute

6.6.2.4 second

```
uint8_t netmd::NetMdTime::second
```

second

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

- /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h

6.7 netmd::TrackTime Struct Reference

track times

```
#include <netmd++.h>
```

Public Attributes

- int mMinutes
  time in minutes
- int mSeconds
  time in seconds
- int mTenthSecs
  time in 10ms

6.7.1 Detailed Description

track times

6.7.2 Member Data Documentation

6.7.2.1 mMinutes

```
int netmd::TrackTime::mMinutes
```

time in minutes

6.7.2.2 mSeconds

```
int netmd::TrackTime::mSeconds
```

time in seconds

6.7.2.3 mTenthSecs

```
int netmd::TrackTime::mTenthSecs
```

time in 10ms

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

- /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h

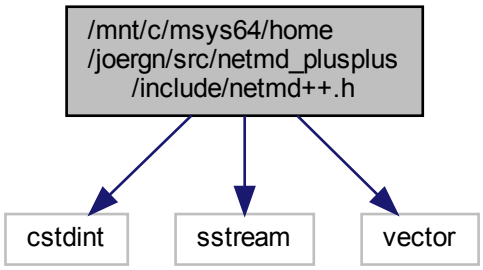
Chapter 7

File Documentation

7.1 /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h File Reference

```
#include <cstdint>
#include <sstream>
#include <vector>
```

Include dependency graph for netmd++.h:



Classes

- struct netmd::TrackTime

  track times
- struct netmd::NetMdTime

  NetMD time.
- struct netmd::DiscCapacity

  Structure to hold the capacity information of a disc.
- struct netmd::Group

  track group
- class netmd::CNetMdApi

This class describes a C++ NetMD access library.

- class netmd::CNetMdTOC

  This class describes a net md TOC.

- struct netmd::CNetMdTOC::DAOFragment

  a fragment used in DAO track

Namespaces

- netmd
- netmd::toc

Typedefs

- using netmd::Groups = std::vector< Group >

  netmd groups

- using netmd::NetMDByteVector = std::vector< uint8_t >

  byte vector

- using netmd::netmd_pp = CNetMdApi

  use netmd_pp instead of CNetMdApi

Enumerations

- enum netmd::DiskFormat : uint8_t {
  netmd::NETMD_DISKFORMAT_LP4 = 0 , netmd::NETMD_DISKFORMAT_LP2 = 2 , netmd::NETMD_DISKFORMAT_SP_MON
  = 4 , netmd::NETMD_DISKFORMAT_SP_STEREO = 6 ,
  netmd::NO_ONTHEFLY_CONVERSION = 0xf }

  disk format

- enum netmd::NetMdErr : int {
  netmd::NETMDERR_NO_ERROR = 0 , netmd::NETMDERR_USB = -1 , netmd::NETMDERR_NOTREADY
  = -2 , netmd::NETMDERR_TIMEOUT = -3 ,
  netmd::NETMDERR_CMD_FAILED = -4 , netmd::NETMDERR_CMD_INVALID = -5 , netmd::NETMDERR_PARAM
  = -6 , netmd::NETMDERR_OTHER = -7 ,
  netmd::NETMDERR_NOT_SUPPORTED = -8 , netmd::NETMDERR_INTERIM = -9 }

  NetMD errors.

- enum class netmd::TrackProtection : uint8_t { netmd::UNPROTECTED = 0x00 , netmd::PROTECTED = 0x03
  , netmd::UNKNOWN = 0xFF }

  type safe protection flags

- enum class netmd::AudioEncoding : uint8_t { netmd::SP = 0x90 , netmd::LP2 = 0x92 , netmd::LP4 = 0x93 ,
  netmd::UNKNOWN = 0xff }

  type safe encoding flags

- enum netmd::typelog {
  netmd::DEBUG , netmd::INFO , netmd::WARN , netmd::CRITICAL ,
  netmd::CAPTURE }

  log severity

- enum netmd::UTOCSector : uint16_t {
  netmd::POS_ADDR , netmd::HW_TITLES , netmd::TSTAMPS , netmd::FW_TITLES ,
  netmd::UNKNWN_1 , netmd::UNKNON_2 }

  TOC sector names.

Functions

- std::ostream & netmd::operator<< (std::ostream &o, const TrackTime &tt)

  format helper for TrackTime
- std::ostream & netmd::operator<< (std::ostream &o, const AudioEncoding &ae)

  format helper for AudioEncoding
- std::ostream & netmd::operator<< (std::ostream &o, const TrackProtection &tp)

  format helper for TrackProtection
- NetMDByteVector & netmd::operator+= (NetMDByteVector &a, const NetMDByteVector &b)

  Addition assignment operator for NetMDByteVector.

Index