

netmd++

1.0.1

Generated by Doxygen 1.9.1

1 Main Page	1
1.1 netmd++	1
1.1.1 Namespace	1
1.1.2 Usage	1
1.1.3 Examples	1
1.1.3.1 Track transfer	1
1.1.3.2 Erase disc and set new title	2
2 Namespace Index	3
2.1 Namespace List	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 netmd Namespace Reference	9
5.1.1 Typedef Documentation	10
5.1.1.1 Groups	10
5.1.1.2 netmd_pp	10
5.1.2 Enumeration Type Documentation	10
5.1.2.1 AudioEncoding	10
5.1.2.2 DiskFormat	11
5.1.2.3 NetMdErr	11
5.1.2.4 TrackProtection	12
5.1.2.5 typelog	12
5.1.3 Function Documentation	12
5.1.3.1 operator<<() [1/3]	12
5.1.3.2 operator<<() [2/3]	13
5.1.3.3 operator<<() [3/3]	13
6 Class Documentation	15
6.1 netmd::CNetMdApi Class Reference	15
6.1.1 Detailed Description	16
6.1.2 Constructor & Destructor Documentation	16
6.1.2.1 CNetMdApi()	17
6.1.2.2 ~CNetMdApi()	17
6.1.3 Member Function Documentation	17
6.1.3.1 addTrackToGroup()	17
6.1.3.2 createGroup()	17
6.1.3.3 deleteGroup()	18
6.1.3.4 deleteTrack()	18

6.1.3.5 delTrackFromGroup()	19
6.1.3.6 discCapacity()	19
6.1.3.7 discFlags()	19
6.1.3.8 discTitle()	19
6.1.3.9 eraseDisc()	20
6.1.3.10 getDeviceName()	20
6.1.3.11 groups()	20
6.1.3.12 initDevice()	21
6.1.3.13 initDiscHeader()	21
6.1.3.14 moveTrack()	21
6.1.3.15 offEncodeSupported()	21
6.1.3.16 rawDiscHeader()	22
6.1.3.17 sendAudioFile()	22
6.1.3.18 setDiscTitle()	23
6.1.3.19 setGroupTitle()	23
6.1.3.20 setLogLevel()	23
6.1.3.21 setLogStream()	24
6.1.3.22 setTrackTitle()	24
6.1.3.23 spUploadSupported()	24
6.1.3.24 trackBitRate()	24
6.1.3.25 trackCount()	25
6.1.3.26 trackFlags()	25
6.1.3.27 trackTime()	25
6.1.3.28 trackTitle()	26
6.1.3.29 writeRawDiscHeader()	26
6.2 netmd::DiscCapacity Struct Reference	27
6.2.1 Detailed Description	27
6.2.2 Member Data Documentation	27
6.2.2.1 available	27
6.2.2.2 recorded	28
6.2.2.3 total	28
6.3 netmd::Group Struct Reference	28
6.3.1 Detailed Description	28
6.3.2 Member Data Documentation	28
6.3.2.1 mFirst	29
6.3.2.2 mGid	29
6.3.2.3 mLast	29
6.3.2.4 mName	29
6.4 netmd::NetMdTime Struct Reference	29
6.4.1 Detailed Description	30
6.4.2 Member Data Documentation	30
6.4.2.1 frame	30

6.4.2.2 hour	30
6.4.2.3 minute	30
6.4.2.4 second	30
6.5 netmd::TrackTime Struct Reference	30
6.5.1 Detailed Description	31
6.5.2 Member Data Documentation	31
6.5.2.1 mMinutes	31
6.5.2.2 mSeconds	31
6.5.2.3 mTenthSecs	31
7 File Documentation	33
7.1 /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h File Reference	33
Index	35

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:

```
{c++}  
#include "path/to/netmd++.h"
```

- create an instance of the API:

```
{c++}  
netmd::netmd_pp* pNetMd = new netmd::netmd_pp();
```

- initialize the first found NetMD device:

```
{c++}  
if ((pNetMd != nullptr) && (pNetMd->initDevice() == netmd::NETMDERR_NO_ERROR))  
{  
    pNetMd->initDiscHeader();  
}
```

- If you change or re-plug the device, simply run above code (init) 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).

```
{c++}  
#include <netmd++.h>  
int main()  
{  
    netmd::netmd_pp* pNetMd = new netmd::netmd_pp();
```

```
if ((pNetMd != nullptr) && (pNetMd->initDevice() == netmd::NETMDERR_NO_ERROR))
{
    pNetMd->initDiscHeader();
    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

```
{c++}
#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->initDiscHeader();
        pNetMd->setDiscTitle("Amazing MD");
    }
    return 0;
}
```


Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

netmd	9
---------------------------------	---

Chapter 3

Class Index

3.1 Class List

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

netmd::CNetMdApi	
This class describes a C++ NetMD access library	15
netmd::DiscCapacity	
Structure to hold the capacity information of a disc	27
netmd::Group	
Track group	28
netmd::NetMdTime	
NetMD time	29
netmd::TrackTime	
Track times	30

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

/mnt/c/msys64/home/joern/src/netmd_plusplus/include/[netmd++.h](#) 33

Chapter 5

Namespace Documentation

5.1 netmd Namespace Reference

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.

Typedefs

- using [Groups](#) = std::vector< [Group](#) >
netmd groups
- 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

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](#)

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](#)

5.1.2 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!

5.1.3 Function Documentation

5.1.3.1 operator<<() [1/3]

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

format helper for AudioEncoding

Parameters

	<i>o</i>	ref. to ostream
<i>in</i>	<i>ae</i>	AudioEncoding

Returns

formatted AudioEncoding stored in ostream

5.1.3.2 operator<<() [2/3]

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

format helper for TrackProtection

Parameters

	<i>o</i>	ref. to ostream
in	<i>tp</i>	TrackProtection

Returns

formatted TrackProtection stored in ostream

5.1.3.3 operator<<() [3/3]

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

format helper for [TrackTime](#)

Parameters

	<i>o</i>	ref. to ostream
in	<i>tt</i>	TrackTime

Returns

formatted [TrackTime](#) stored in ostream

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.
- int [initDiscHeader](#) ()
Initializes the disc header.
- 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 [rawDiscHeader](#) (std::string &header)
get raw disc header
- int [discTitle](#) (std::string &title)
get disc title
- int [setDiscTitle](#) (const std::string &title)
Sets the disc title.
- int [writeRawDiscHeader](#) ()
Writes a disc header.

- 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
- 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

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	<i>track</i>	The track
in	<i>group</i>	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	<i>title</i>	The title
in	<i>first</i>	The first track
in	<i>last</i>	The last track

Returns[NetMdErr](#)**6.1.3.3 deleteGroup()**

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

delete a group

Parameters

in	<i>group</i>	The group
----	--------------	-----------

Returns[NetMdErr](#)**6.1.3.4 deleteTrack()**

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

delete track

Parameters

in	<i>track</i>	The track number
----	--------------	------------------

Returns[NetMdErr](#)

6.1.3.5 delTrackFromGroup()

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

remove track from group

Parameters

in	<i>track</i>	The track
in	<i>group</i>	The group

Returns

[NetMdErr](#)

6.1.3.6 discCapacity()

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

get disc capacity

Parameters

out	<i>dcap</i>	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	<i>title</i>	The title
-----	--------------	-----------

Returns

[NetMdErr](#)

6.1.3.9 eraseDisc()

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

erase MD

Returns

[NetMdErr](#)

6.1.3.10 getDeviceName()

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

Gets the device name.

Returns

The device name.

6.1.3.11 groups()

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

get MD track groups

Returns

vector of group structures

6.1.3.12 initDevice()

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

Initializes the device.

Returns

NetMdErr

6.1.3.13 initDiscHeader()

```
int netmd::CNetMdApi::initDiscHeader ( )
```

Initializes the disc header.

Returns

NetMdErr

6.1.3.14 moveTrack()

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

move a track (number)

Parameters

in	<i>from</i>	from position
in	<i>to</i>	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 rawDiscHeader()

```
int netmd::CNetMdApi::rawDiscHeader (
    std::string & header )
```

get raw disc header

Parameters

out	<i>header</i>	The buffer for disc header
-----	---------------	----------------------------

Returns

[NetMdErr](#)

6.1.3.17 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	<i>filename</i>	The filename
in	<i>title</i>	The title
in	<i>otf</i>	The disk format

Returns

[NetMdErr](#)

6.1.3.18 setDiscTitle()

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

Sets the disc title.

Parameters

in	<i>title</i>	The title
----	--------------	-----------

Returns

[NetMdErr](#)

6.1.3.19 setGroupTitle()

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

Sets the group title.

Parameters

in	<i>group</i>	The group
in	<i>title</i>	The title

Returns

[NetMdErr](#)

6.1.3.20 setLogLevel()

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

Sets the log level.

Parameters

in	<i>severity</i>	The severity
----	-----------------	--------------

6.1.3.21 setLogStream()

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

Sets the log stream.

Parameters

in	<i>os</i>	The stream instance to log to
----	-----------	-------------------------------

6.1.3.22 setTrackTitle()

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

Sets the track title.

Parameters

in	<i>trackNo</i>	The track no
in	<i>title</i>	The title

Returns

[NetMdErr](#)

6.1.3.23 spUploadSupported()

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

is SP upload supported?

Returns

true if yes

6.1.3.24 trackBitRate()

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

get track bitrate data

Parameters

in	<i>track</i>	The track number
out	<i>encoding</i>	The encoding flag
out	<i>channel</i>	The channel flag

Returns

[NetMdErr](#)

6.1.3.25 trackCount()

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

request track count

Returns

< 0 -> [NetMdErr](#); else -> track count

6.1.3.26 trackFlags()

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

get track flags

Parameters

in	<i>track</i>	The track number
out	<i>flags</i>	The track flags

Returns

[NetMdErr](#)

6.1.3.27 trackTime()

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

get track time

Parameters

in	<i>trackNo</i>	The track no
out	<i>trackTime</i>	The track time

Returns

[NetMdErr](#)

6.1.3.28 trackTitle()

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

get track title

Parameters

in	<i>track</i>	The track number
out	<i>title</i>	The track title

Returns

[NetMdErr](#)

6.1.3.29 writeRawDiscHeader()

```
int netmd::CNetMdApi::writeRawDiscHeader ( )
```

Writes a disc header.

Returns

[NetMdErr](#)

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

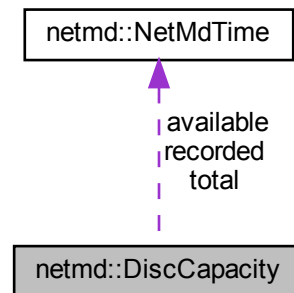
- /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/[netmd++.h](#)

6.2 netmd::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 already recorded on the disc.
- [NetMdTime total](#)
- [NetMdTime available](#)

6.2.1 Detailed Description

Structure to hold the capacity information of a disc.

6.2.2 Member Data Documentation

6.2.2.1 available

[NetMdTime](#) `netmd::DiscCapacity::available`

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

6.2.2.2 recorded

`NetMdTime netmd::DiscCapacity::recorded`

Time already recorded on the disc.

6.2.2.3 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`

6.3 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

6.3.1 Detailed Description

track group

6.3.2 Member Data Documentation

6.3.2.1 mFirst

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

first track

6.3.2.2 mGid

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

group id

6.3.2.3 mLast

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

last track

6.3.2.4 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](#)

6.4 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

6.4.1 Detailed Description

NetMD time.

6.4.2 Member Data Documentation

6.4.2.1 frame

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

frame

6.4.2.2 hour

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

hour

6.4.2.3 minute

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

minute

6.4.2.4 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`

6.5 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

6.5.1 Detailed Description

track times

6.5.2 Member Data Documentation

6.5.2.1 mMinutes

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

time in minutes

6.5.2.2 mSeconds

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

time in seconds

6.5.2.3 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`

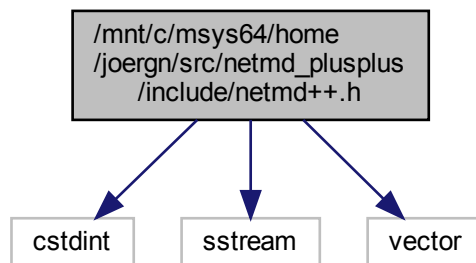
Chapter 7

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.

Namespaces

- [netmd](#)

Typedefs

- using [netmd::Groups](#) = std::vector< Group >
netmd groups
- 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_MONO](#) = 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

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

Index

/mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h, [33](#)

~CNetMdApi
 netmd::CNetMdApi, [17](#)

addTrackToGroup
 netmd::CNetMdApi, [17](#)

AudioEncoding
 netmd, [10](#)

available
 netmd::DiscCapacity, [27](#)

CAPTURE
 netmd, [12](#)

CNetMdApi
 netmd::CNetMdApi, [16](#)

createGroup
 netmd::CNetMdApi, [17](#)

CRITICAL
 netmd, [12](#)

DEBUG
 netmd, [12](#)

deleteGroup
 netmd::CNetMdApi, [18](#)

deleteTrack
 netmd::CNetMdApi, [18](#)

delTrackFromGroup
 netmd::CNetMdApi, [18](#)

discCapacity
 netmd::CNetMdApi, [19](#)

discFlags
 netmd::CNetMdApi, [19](#)

discTitle
 netmd::CNetMdApi, [19](#)

DiskFormat
 netmd, [11](#)

eraseDisc
 netmd::CNetMdApi, [20](#)

frame
 netmd::NetMdTime, [30](#)

getDeviceName
 netmd::CNetMdApi, [20](#)

Groups
 netmd, [10](#)

groups
 netmd::CNetMdApi, [20](#)

netmd::NetMdTime, [30](#)

INFO
 netmd, [12](#)

initDevice
 netmd::CNetMdApi, [20](#)

initDiscHeader
 netmd::CNetMdApi, [21](#)

LP2
 netmd, [11](#)

LP4
 netmd, [11](#)

mFirst
 netmd::Group, [28](#)

mGid
 netmd::Group, [29](#)

minute
 netmd::NetMdTime, [30](#)

mLast
 netmd::Group, [29](#)

mMinutes
 netmd::TrackTime, [31](#)

mName
 netmd::Group, [29](#)

moveTrack
 netmd::CNetMdApi, [21](#)

mSeconds
 netmd::TrackTime, [31](#)

mTenthSecs
 netmd::TrackTime, [31](#)

netmd, [9](#)
 AudioEncoding, [10](#)
 CAPTURE, [12](#)
 CRITICAL, [12](#)
 DEBUG, [12](#)
 DiskFormat, [11](#)
 Groups, [10](#)
 INFO, [12](#)
 LP2, [11](#)
 LP4, [11](#)
 NETMD_DISKFORMAT_LP2, [11](#)
 NETMD_DISKFORMAT_LP4, [11](#)
 NETMD_DISKFORMAT_SP_MONO, [11](#)
 NETMD_DISKFORMAT_SP_STEREO, [11](#)
 netmd_pp, [10](#)
 NetMdErr, [11](#)

- NETMDERR_CMD_FAILED, 11
- NETMDERR_CMD_INVALID, 11
- NETMDERR_INTERIM, 11
- NETMDERR_NO_ERROR, 11
- NETMDERR_NOT_SUPPORTED, 11
- NETMDERR_NOTREADY, 11
- NETMDERR_OTHER, 11
- NETMDERR_PARAM, 11
- NETMDERR_TIMEOUT, 11
- NETMDERR_USB, 11
- NO_ONTHEFLY_CONVERSION, 11
- operator<<, 12, 13
- PROTECTED, 12
- SP, 11
- TrackProtection, 11
- typelog, 12
- UNKNOWN, 11, 12
- UNPROTECTED, 12
- WARN, 12
- netmd::CNetMdApi, 15
 - ~CNetMdApi, 17
 - addTrackToGroup, 17
 - CNetMdApi, 16
 - createGroup, 17
 - deleteGroup, 18
 - deleteTrack, 18
 - delTrackFromGroup, 18
 - discCapacity, 19
 - discFlags, 19
 - discTitle, 19
 - eraseDisc, 20
 - getDeviceName, 20
 - groups, 20
 - initDevice, 20
 - initDiscHeader, 21
 - moveTrack, 21
 - otfEncodeSupported, 21
 - rawDiscHeader, 22
 - sendAudioFile, 22
 - setDiscTitle, 22
 - setGroupTitle, 23
 - setLogLevel, 23
 - setLogStream, 23
 - setTrackTitle, 24
 - spUploadSupported, 24
 - trackBitRate, 24
 - trackCount, 25
 - trackFlags, 25
 - trackTime, 25
 - trackTitle, 26
 - writeRawDiscHeader, 26
- netmd::DiscCapacity, 27
 - available, 27
 - recorded, 27
 - total, 28
- netmd::Group, 28
 - mFirst, 28
 - mGid, 29
 - mLast, 29
 - mName, 29
- netmd::NetMdTime, 29
 - frame, 30
 - hour, 30
 - minute, 30
 - second, 30
- netmd::TrackTime, 30
 - mMinutes, 31
 - mSeconds, 31
 - mTenthSecs, 31
- NETMD_DISKFORMAT_LP2
 - netmd, 11
- NETMD_DISKFORMAT_LP4
 - netmd, 11
- NETMD_DISKFORMAT_SP_MONO
 - netmd, 11
- NETMD_DISKFORMAT_SP_STEREO
 - netmd, 11
- netmd_pp
 - netmd, 10
- NetMdErr
 - netmd, 11
- NETMDERR_CMD_FAILED
 - netmd, 11
- NETMDERR_CMD_INVALID
 - netmd, 11
- NETMDERR_INTERIM
 - netmd, 11
- NETMDERR_NO_ERROR
 - netmd, 11
- NETMDERR_NOT_SUPPORTED
 - netmd, 11
- NETMDERR_NOTREADY
 - netmd, 11
- NETMDERR_OTHER
 - netmd, 11
- NETMDERR_PARAM
 - netmd, 11
- NETMDERR_TIMEOUT
 - netmd, 11
- NETMDERR_USB
 - netmd, 11
- NO_ONTHEFLY_CONVERSION
 - netmd, 11
- operator<<
 - netmd, 12, 13
- otfEncodeSupported
 - netmd::CNetMdApi, 21
- PROTECTED
 - netmd, 12
- rawDiscHeader
 - netmd::CNetMdApi, 22
- recorded
 - netmd::DiscCapacity, 27

- second
 - netmd::NetMdTime, [30](#)
- sendAudioFile
 - netmd::CNetMdApi, [22](#)
- setDiscTitle
 - netmd::CNetMdApi, [22](#)
- setGroupTitle
 - netmd::CNetMdApi, [23](#)
- setLogLevel
 - netmd::CNetMdApi, [23](#)
- setLogStream
 - netmd::CNetMdApi, [23](#)
- setTrackTitle
 - netmd::CNetMdApi, [24](#)
- SP
 - netmd, [11](#)
- spUploadSupported
 - netmd::CNetMdApi, [24](#)
- total
 - netmd::DiscCapacity, [28](#)
- trackBitRate
 - netmd::CNetMdApi, [24](#)
- trackCount
 - netmd::CNetMdApi, [25](#)
- trackFlags
 - netmd::CNetMdApi, [25](#)
- TrackProtection
 - netmd, [11](#)
- trackTime
 - netmd::CNetMdApi, [25](#)
- trackTitle
 - netmd::CNetMdApi, [26](#)
- typelog
 - netmd, [12](#)
- UNKNOWN
 - netmd, [11](#), [12](#)
- UNPROTECTED
 - netmd, [12](#)
- WARN
 - netmd, [12](#)
- writeRawDiscHeader
 - netmd::CNetMdApi, [26](#)