

netmd++

1.0.2

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
1.2 MDs UTOC	2
1.2.1 Addressing in UTOC	2
1.2.2 Modifying the UTOC	2
2 Namespace Index	5
2.1 Namespace List	5
3 Class Index	7
3.1 Class List	7
4 File Index	9
4.1 File List	9
5 Namespace Documentation	11
5.1 netmd Namespace Reference	11
5.1.1 Typedef Documentation	12
5.1.1.1 Groups	12
5.1.1.2 netmd_pp	13
5.1.1.3 NetMDByteVector	13
5.1.2 Enumeration Type Documentation	13
5.1.2.1 AudioEncoding	13
5.1.2.2 DiskFormat	13
5.1.2.3 NetMdErr	14
5.1.2.4 TrackProtection	14
5.1.2.5 typelog	14
5.1.2.6 UTOCSector	15
5.1.3 Function Documentation	15
5.1.3.1 operator+=()	15
5.1.3.2 operator<<() [1/3]	16
5.1.3.3 operator<<() [2/3]	16
5.1.3.4 operator<<() [3/3]	16
5.2 netmd::toc Namespace Reference	17
6 Class Documentation	19
6.1 netmd::CNetMdApi Class Reference	19
6.1.1 Detailed Description	20
6.1.2 Constructor & Destructor Documentation	21

6.1.2.1 CNetMdApi()	21
6.1.2.2 ~CNetMdApi()	21
6.1.3 Member Function Documentation	21
6.1.3.1 addTrackToGroup()	21
6.1.3.2 createGroup()	21
6.1.3.3 deleteGroup()	22
6.1.3.4 deleteTrack()	22
6.1.3.5 delTrackFromGroup()	23
6.1.3.6 discCapacity()	23
6.1.3.7 discFlags()	23
6.1.3.8 discTitle()	23
6.1.3.9 eraseDisc()	24
6.1.3.10 finalizeTOC()	24
6.1.3.11 getDeviceName()	25
6.1.3.12 groups()	25
6.1.3.13 initDevice()	25
6.1.3.14 moveTrack()	25
6.1.3.15 offEncodeSupported()	26
6.1.3.16 prepareTOCManip()	26
6.1.3.17 readUTOCSector()	26
6.1.3.18 sendAudioFile()	27
6.1.3.19 setDiscTitle()	27
6.1.3.20 setGroupTitle()	27
6.1.3.21 setLogLevel()	28
6.1.3.22 setLogStream()	28
6.1.3.23 setTrackTitle()	28
6.1.3.24 spUploadSupported()	29
6.1.3.25 tocManipSupported()	29
6.1.3.26 trackBitRate()	29
6.1.3.27 trackCount()	30
6.1.3.28 trackFlags()	30
6.1.3.29 trackTime()	30
6.1.3.30 trackTitle()	31
6.1.3.31 writeUTOCSector()	31
6.2 netmd::CNetMdTOC Class Reference	31
6.2.1 Detailed Description	32
6.2.2 Member Typedef Documentation	32
6.2.2.1 DAOFragments	32
6.2.3 Constructor & Destructor Documentation	33
6.2.3.1 CNetMdTOC()	33
6.2.3.2 ~CNetMdTOC()	33
6.2.4 Member Function Documentation	33

6.2.4.1 addTrack()	33
6.2.4.2 discInfo()	34
6.2.4.3 discTitle()	34
6.2.4.4 import()	34
6.2.4.5 setDiscTitle()	35
6.2.4.6 trackCount()	35
6.2.4.7 trackInfo()	35
6.2.4.8 trackTitle()	36
6.3 netmd::CNetMdTOC::DAOFragment Struct Reference	36
6.3.1 Detailed Description	36
6.3.2 Member Data Documentation	36
6.3.2.1 mEnd	36
6.3.2.2 mStart	37
6.4 netmd::DiscCapacity Struct Reference	37
6.4.1 Detailed Description	37
6.4.2 Member Data Documentation	37
6.4.2.1 available	38
6.4.2.2 recorded	38
6.4.2.3 total	38
6.5 netmd::Group Struct Reference	38
6.5.1 Detailed Description	38
6.5.2 Member Data Documentation	39
6.5.2.1 mFirst	39
6.5.2.2 mGid	39
6.5.2.3 mLast	39
6.5.2.4 mName	39
6.6 netmd::NetMdTime Struct Reference	39
6.6.1 Detailed Description	40
6.6.2 Member Data Documentation	40
6.6.2.1 frame	40
6.6.2.2 hour	40
6.6.2.3 minute	40
6.6.2.4 second	41
6.7 netmd::TrackTime Struct Reference	41
6.7.1 Detailed Description	41
6.7.2 Member Data Documentation	41
6.7.2.1 mMinutes	41
6.7.2.2 mSeconds	41
6.7.2.3 mTenthSecs	41
7 File Documentation	43
7.1 /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h File Reference	43

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 smallest unit is a sound frame, representing 11.6ms of mono audio (212 bytes), while the smallest **addressable** unit is the sound group, containing 2 sound frames. A sector contains 11 sound frames / 5.5 sound groups. Addressing must be done through sound group. Sound groups are numbered 0 ... 10. Sound groups 0 ... 5 are part of the even sector, while sound groups 5 ... 10 are part of the odd sector. Group 5 overlaps both even and odd sectors and can therefore be addressed on both sectors.

```
+-----+
|               sector pair               |
+-----+-----+
| even sector (2n) | odd sector (2n+1) |
+---+---+---+---+---+---+---+---+---+
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | <- sound groups
+---+---+---+---+---+---+---+---+---+
```

A cluster is an aggregate of 32 audio sectors (176 sound groups) representing 2.04 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:

netmd	11
netmd::toc	17

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	19
netmd::CNetMdTOC	
This class describes a net md TOC	31
netmd::CNetMdTOC::DAOFragment	
Fragment used in DAO track	36
netmd::DiscCapacity	
Structure to hold the capacity information of a disc	37
netmd::Group	
Track group	38
netmd::NetMdTime	
NetMD time	39
netmd::TrackTime	
Track times	41

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](#) 43

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 , [NETMDERR_AGAIN](#) = -10 }
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](#)
- [NetMDBByteVector](#) & [operator+=](#) ([NetMDBByteVector](#) &a, const [NetMDBByteVector](#) &b)
Addition assignment operator for [NetMDBByteVector](#).

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
NETMDERR_AGAIN	try again

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
-------	-------------------

Enumerator

INFO	information
WARN	more serious
CRITICAL	critical information
CAPTURE	needed for log parsing!

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+=()

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

Addition assignment operator for NetMByteVector.

Parameters

	<i>a</i>	byte vector 1
in	<i>b</i>	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

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

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

	<i>o</i>	ref. to ostream
in	<i>tt</i>	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](#) of)
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](#) (bool reset=false, 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	<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 finalizeTOC()**

```
int netmd::CNetMdApi::finalizeTOC (
    bool reset = false,
    uint8_t resetWait = 15 )
```

finalize TOC through exploit

Parameters

in	<i>reset</i>	do reset if true (default: false)
in	<i>resetWait</i>	The optional reset wait time (15 seconds) Only needed if reset is true

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	<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 prepareTOCManip()

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

prepare TOC manipulation

Returns[NetMdErr](#)**6.1.3.17 readUTOCSector()**

```
NetMByteVector 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	<i>filename</i>	The filename
in	<i>title</i>	The title
in	<i>otf</i>	The disk format

Returns

[NetMdErr](#)

6.1.3.19 setDiscTitle()

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

Sets the disc title.

Parameters

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

Returns

[NetMdErr](#)

6.1.3.20 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](#)

6.1.3.21 setLogLevel()

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

Sets the log level.

Parameters

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

6.1.3.22 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
----	-----------	-------------------------------

6.1.3.23 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](#)**6.1.3.24 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	<i>track</i>	The track number
out	<i>encoding</i>	The encoding flag
out	<i>channel</i>	The channel flag

Returns

[NetMdErr](#)

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

Returns

[NetMdErr](#)

6.1.3.29 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](#)

6.1.3.30 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](#)

6.1.3.31 writeUTOCSector()

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

Writes an utoc sector.

Parameters

in	<i>s</i>	sector names
in	<i>data</i>	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	<i>trackCount</i>	The track count
in	<i>lenInMs</i>	The length in milliseconds
	<i>data</i>	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	<i>no</i>	track number (starting with 1)
in	<i>lengthMs</i>	The length in milliseconds
in	<i>title</i>	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	<i>trackCount</i>	The track count
in	<i>lenInMs</i>	The length in milliseconds
	<i>data</i>	The TOC data

6.2.4.5 setDiscTitle()

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

Sets the disc title.

Parameters

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

Returns

track info

6.2.4.8 trackTitle()

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

get track title

Parameters

in	<i>trackNo</i>	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:

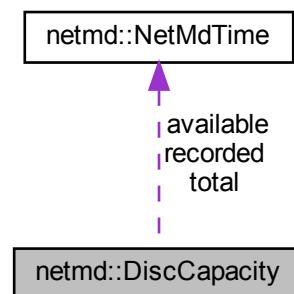
- /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h

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 already 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 already 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](#)

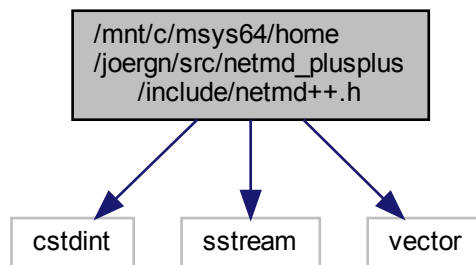
Chapter 7

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_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::NETMDERR_AGAIN](#) = -10 }
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

/mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd-netmd::CNetMdApi, 24
43

~CNetMdApi
netmd::CNetMdApi, 21
~CNetMdTOC
netmd::CNetMdTOC, 33

addTrack
netmd::CNetMdTOC, 33
addTrackToGroup
netmd::CNetMdApi, 21
AudioEncoding
netmd, 13
available
netmd::DiscCapacity, 37

CAPTURE
netmd, 15
CNetMdApi
netmd::CNetMdApi, 21
CNetMdTOC
netmd::CNetMdTOC, 33
createGroup
netmd::CNetMdApi, 21
CRITICAL
netmd, 15

DAOFragments
netmd::CNetMdTOC, 32
DEBUG
netmd, 14
deleteGroup
netmd::CNetMdApi, 22
deleteTrack
netmd::CNetMdApi, 22
delTrackFromGroup
netmd::CNetMdApi, 22
discCapacity
netmd::CNetMdApi, 23
discFlags
netmd::CNetMdApi, 23
discInfo
netmd::CNetMdTOC, 34
discTitle
netmd::CNetMdApi, 23
netmd::CNetMdTOC, 34
DiskFormat
netmd, 13

eraseDisc

finalizeTOC
netmd::CNetMdApi, 24
frame
netmd::NetMdTime, 40
FW_TITLES
netmd, 15
getDeviceName
netmd::CNetMdApi, 24
Groups
netmd, 12
groups
netmd::CNetMdApi, 25

hour
netmd::NetMdTime, 40
HW_TITLES
netmd, 15

import
netmd::CNetMdTOC, 34
INFO
netmd, 15
initDevice
netmd::CNetMdApi, 25

LP2
netmd, 13
LP4
netmd, 13

mEnd
netmd::CNetMdTOC::DAOFragment, 36
mFirst
netmd::Group, 39
mGid
netmd::Group, 39
minute
netmd::NetMdTime, 40
mLast
netmd::Group, 39
mMinutes
netmd::TrackTime, 41
mName
netmd::Group, 39
moveTrack
netmd::CNetMdApi, 25
mSeconds
netmd::TrackTime, 41

- mStart
 - netmd::CNetMdTOC::DAOFragment, 36
- mTenthSecs
 - netmd::TrackTime, 41
- netmd, 11
 - AudioEncoding, 13
 - CAPTURE, 15
 - CRITICAL, 15
 - DEBUG, 14
 - DiskFormat, 13
 - FW_TITLES, 15
 - Groups, 12
 - HW_TITLES, 15
 - INFO, 15
 - LP2, 13
 - LP4, 13
 - NETMD_DISKFORMAT_LP2, 13
 - NETMD_DISKFORMAT_LP4, 13
 - NETMD_DISKFORMAT_SP_MONO, 13
 - NETMD_DISKFORMAT_SP_STEREO, 13
 - netmd_pp, 12
 - NetMDByteVector, 13
 - NetMdErr, 14
 - NETMDERR_AGAIN, 14
 - NETMDERR_CMD_FAILED, 14
 - NETMDERR_CMD_INVALID, 14
 - NETMDERR_INTERIM, 14
 - NETMDERR_NO_ERROR, 14
 - NETMDERR_NOT_SUPPORTED, 14
 - NETMDERR_NOTREADY, 14
 - NETMDERR_OTHER, 14
 - NETMDERR_PARAM, 14
 - NETMDERR_TIMEOUT, 14
 - NETMDERR_USB, 14
 - NO_ONTHEFLY_CONVERSION, 13
 - operator<=, 15, 16
 - operator+=, 15
 - POS_ADDR, 15
 - PROTECTED, 14
 - SP, 13
 - TrackProtection, 14
 - TSTAMPS, 15
 - typelog, 14
 - UNKNON_2, 15
 - UNKNOWN, 13, 14
 - UNKNWN_1, 15
 - UNPROTECTED, 14
 - UTOCSector, 15
 - WARN, 15
- netmd::CNetMdApi, 19
 - ~CNetMdApi, 21
 - addTrackToGroup, 21
 - CNetMdApi, 21
 - createGroup, 21
 - deleteGroup, 22
 - deleteTrack, 22
 - delTrackFromGroup, 22
 - discCapacity, 23
 - discFlags, 23
 - discTitle, 23
 - eraseDisc, 24
 - finalizeTOC, 24
 - getDeviceName, 24
 - groups, 25
 - initDevice, 25
 - moveTrack, 25
 - otfEncodeSupported, 26
 - prepareTOCManip, 26
 - readUTOCSector, 26
 - sendAudioFile, 26
 - setDiscTitle, 27
 - setGroupTitle, 27
 - setLogLevel, 28
 - setLogStream, 28
 - setTrackTitle, 28
 - spUploadSupported, 29
 - tocManipSupported, 29
 - trackBitRate, 29
 - trackCount, 29
 - trackFlags, 30
 - trackTime, 30
 - trackTitle, 30
 - writeUTOCSector, 31
- netmd::CNetMdTOC, 31
 - ~CNetMdTOC, 33
 - addTrack, 33
 - CNetMdTOC, 33
 - DAOFragments, 32
 - discInfo, 34
 - discTitle, 34
 - import, 34
 - setDiscTitle, 34
 - trackCount, 35
 - trackInfo, 35
 - trackTitle, 35
- netmd::CNetMdTOC::DAOFragment, 36
 - mEnd, 36
 - mStart, 36
- netmd::DiscCapacity, 37
 - available, 37
 - recorded, 38
 - total, 38
- netmd::Group, 38
 - mFirst, 39
 - mGid, 39
 - mLast, 39
 - mName, 39
- netmd::NetMdTime, 39
 - frame, 40
 - hour, 40
 - minute, 40
 - second, 40
- netmd::toc, 17
- netmd::TrackTime, 41
 - mMinutes, 41
 - mSeconds, 41

- mTenthSecs, [41](#)
- NETMD_DISKFORMAT_LP2
 - netmd, [13](#)
- NETMD_DISKFORMAT_LP4
 - netmd, [13](#)
- NETMD_DISKFORMAT_SP_MONO
 - netmd, [13](#)
- NETMD_DISKFORMAT_SP_STEREO
 - netmd, [13](#)
- netmd_pp
 - netmd, [12](#)
- NetMDByteVector
 - netmd, [13](#)
- NetMdErr
 - netmd, [14](#)
- NETMDERR_AGAIN
 - netmd, [14](#)
- NETMDERR_CMD_FAILED
 - netmd, [14](#)
- NETMDERR_CMD_INVALID
 - netmd, [14](#)
- NETMDERR_INTERIM
 - netmd, [14](#)
- NETMDERR_NO_ERROR
 - netmd, [14](#)
- NETMDERR_NOT_SUPPORTED
 - netmd, [14](#)
- NETMDERR_NOTREADY
 - netmd, [14](#)
- NETMDERR_OTHER
 - netmd, [14](#)
- NETMDERR_PARAM
 - netmd, [14](#)
- NETMDERR_TIMEOUT
 - netmd, [14](#)
- NETMDERR_USB
 - netmd, [14](#)
- NO_ONTHEFLY_CONVERSION
 - netmd, [13](#)
- operator<<
 - netmd, [15](#), [16](#)
- operator+=
 - netmd, [15](#)
- otfEncodeSupported
 - netmd::CNetMdApi, [26](#)
- POS_ADDR
 - netmd, [15](#)
- prepareTOCManip
 - netmd::CNetMdApi, [26](#)
- PROTECTED
 - netmd, [14](#)
- readUTOCSector
 - netmd::CNetMdApi, [26](#)
- recorded
 - netmd::DiscCapacity, [38](#)
- second
 - netmd::NetMdTime, [40](#)
- sendAudioFile
 - netmd::CNetMdApi, [26](#)
- setDiscTitle
 - netmd::CNetMdApi, [27](#)
 - netmd::CNetMdTOC, [34](#)
- setGroupTitle
 - netmd::CNetMdApi, [27](#)
- setLogLevel
 - netmd::CNetMdApi, [28](#)
- setLogStream
 - netmd::CNetMdApi, [28](#)
- setTrackTitle
 - netmd::CNetMdApi, [28](#)
- SP
 - netmd, [13](#)
- spUploadSupported
 - netmd::CNetMdApi, [29](#)
- tocManipSupported
 - netmd::CNetMdApi, [29](#)
- total
 - netmd::DiscCapacity, [38](#)
- trackBitRate
 - netmd::CNetMdApi, [29](#)
- trackCount
 - netmd::CNetMdApi, [29](#)
 - netmd::CNetMdTOC, [35](#)
- trackFlags
 - netmd::CNetMdApi, [30](#)
- trackInfo
 - netmd::CNetMdTOC, [35](#)
- TrackProtection
 - netmd, [14](#)
- trackTime
 - netmd::CNetMdApi, [30](#)
- trackTitle
 - netmd::CNetMdApi, [30](#)
 - netmd::CNetMdTOC, [35](#)
- TSTAMPS
 - netmd, [15](#)
- typelog
 - netmd, [14](#)
- UNKNOWN_2
 - netmd, [15](#)
- UNKNOWN
 - netmd, [13](#), [14](#)
- UNKNWN_1
 - netmd, [15](#)
- UNPROTECTED
 - netmd, [14](#)
- UTOCSector
 - netmd, [15](#)
- WARN
 - netmd, [15](#)
- writeUTOCSector

netmd::CNetMdApi, [31](#)