

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
1.2 MDs UTOC	2
1.2.1 Addressing in UTOC	2
1.2.2 Modifying the UTOC	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	11
5.1.1.3 NetMDByteVector	11
5.1.2 Enumeration Type Documentation	11
5.1.2.1 AudioEncoding	11
5.1.2.2 DiskFormat	11
5.1.2.3 NetMdErr	12
5.1.2.4 TrackProtection	12
5.1.2.5 typelog	12
5.1.2.6 UTOCSector	13
5.1.3 Function Documentation	13
5.1.3.1 operator+=()	13
5.1.3.2 operator<<() [1/3]	13
5.1.3.3 operator<<() [2/3]	14
5.1.3.4 operator<<() [3/3]	14
5.2 netmd::toc Namespace Reference	15
6 Class Documentation	17
6.1 netmd::CNetMdApi Class Reference	17
6.1.1 Detailed Description	18
6.1.2 Constructor & Destructor Documentation	19

6.1.2.1 CNetMdApi()	19
6.1.2.2 ~CNetMdApi()	19
6.1.3 Member Function Documentation	19
6.1.3.1 addTrackToGroup()	19
6.1.3.2 createGroup()	19
6.1.3.3 deleteGroup()	20
6.1.3.4 deleteTrack()	20
6.1.3.5 delTrackFromGroup()	21
6.1.3.6 discCapacity()	21
6.1.3.7 discFlags()	21
6.1.3.8 discTitle()	21
6.1.3.9 eraseDisc()	22
6.1.3.10 finalizeTOC()	22
6.1.3.11 getDeviceName()	22
6.1.3.12 groups()	23
6.1.3.13 initDevice()	23
6.1.3.14 moveTrack()	23
6.1.3.15 offEncodeSupported()	24
6.1.3.16 prepareTOCManip()	24
6.1.3.17 readUTOCSector()	24
6.1.3.18 sendAudioFile()	24
6.1.3.19 setDiscTitle()	25
6.1.3.20 setGroupTitle()	25
6.1.3.21 setLogLevel()	26
6.1.3.22 setLogStream()	26
6.1.3.23 setTrackTitle()	26
6.1.3.24 spUploadSupported()	27
6.1.3.25 tocManipSupported()	27
6.1.3.26 trackBitRate()	27
6.1.3.27 trackCount()	28
6.1.3.28 trackFlags()	28
6.1.3.29 trackTime()	28
6.1.3.30 trackTitle()	29
6.1.3.31 writeUTOCSector()	29
6.2 netmd::CNetMdTOC Class Reference	29
6.2.1 Detailed Description	30
6.2.2 Member Typedef Documentation	30
6.2.2.1 DAOFragments	30
6.2.3 Constructor & Destructor Documentation	31
6.2.3.1 CNetMdTOC()	31
6.2.3.2 ~CNetMdTOC()	31
6.2.4 Member Function Documentation	31

6.2.4.1 addTrack()	31
6.2.4.2 discInfo()	32
6.2.4.3 discTitle()	32
6.2.4.4 import()	32
6.2.4.5 setDiscTitle()	33
6.2.4.6 trackCount()	33
6.2.4.7 trackInfo()	33
6.2.4.8 trackTitle()	34
6.3 netmd::CNetMdTOC::DAOFragment Struct Reference	34
6.3.1 Detailed Description	34
6.3.2 Member Data Documentation	34
6.3.2.1 mEnd	34
6.3.2.2 mStart	35
6.4 netmd::DiscCapacity Struct Reference	35
6.4.1 Detailed Description	35
6.4.2 Member Data Documentation	35
6.4.2.1 available	36
6.4.2.2 recorded	36
6.4.2.3 total	36
6.5 netmd::Group Struct Reference	36
6.5.1 Detailed Description	36
6.5.2 Member Data Documentation	37
6.5.2.1 mFirst	37
6.5.2.2 mGid	37
6.5.2.3 mLast	37
6.5.2.4 mName	37
6.6 netmd::NetMdTime Struct Reference	37
6.6.1 Detailed Description	38
6.6.2 Member Data Documentation	38
6.6.2.1 frame	38
6.6.2.2 hour	38
6.6.2.3 minute	38
6.6.2.4 second	39
6.7 netmd::TrackTime Struct Reference	39
6.7.1 Detailed Description	39
6.7.2 Member Data Documentation	39
6.7.2.1 mMinutes	39
6.7.2.2 mSeconds	39
6.7.2.3 mTenthSecs	39
7 File Documentation	41
7.1 /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h File Reference	41

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:

netmd	9
netmd::toc	15

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	17
netmd::CNetMdTOC	
This class describes a net md TOC	29
netmd::CNetMdTOC::DAOFragment	
Fragment used in DAO track	34
netmd::DiscCapacity	
Structure to hold the capacity information of a disc	35
netmd::Group	
Track group	36
netmd::NetMdTime	
NetMD time	37
netmd::TrackTime	
Track times	39

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](#) 41

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](#)
- [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

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+=()

```
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](#) (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

<code>out</code>	<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 (
    uint8_t resetWait = 15 )
```

finalize TOC through exploit

Parameters

<code>in</code>	<i>resetWait</i>	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	<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:

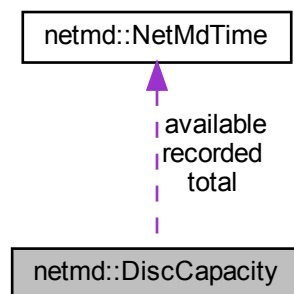
- /mnt/c/msys64/home/joergn/src/netmd_plusplus/include/netmd++.h

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](#)

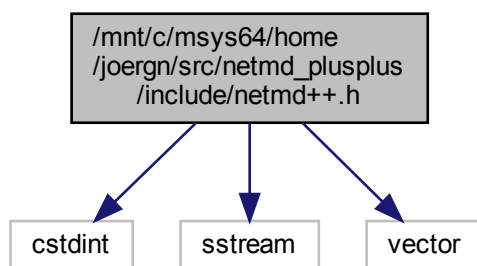
Chapter 7

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 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, 22
41

~CNetMdApi
 netmd::CNetMdApi, 19
~CNetMdTOC
 netmd::CNetMdTOC, 31

addTrack
 netmd::CNetMdTOC, 31
addTrackToGroup
 netmd::CNetMdApi, 19
AudioEncoding
 netmd, 11
available
 netmd::DiscCapacity, 35

CAPTURE
 netmd, 12
CNetMdApi
 netmd::CNetMdApi, 19
CNetMdTOC
 netmd::CNetMdTOC, 31
createGroup
 netmd::CNetMdApi, 19
CRITICAL
 netmd, 12

DAOFragments
 netmd::CNetMdTOC, 30
DEBUG
 netmd, 12
deleteGroup
 netmd::CNetMdApi, 20
deleteTrack
 netmd::CNetMdApi, 20
delTrackFromGroup
 netmd::CNetMdApi, 20
discCapacity
 netmd::CNetMdApi, 21
discFlags
 netmd::CNetMdApi, 21
discInfo
 netmd::CNetMdTOC, 32
discTitle
 netmd::CNetMdApi, 21
 netmd::CNetMdTOC, 32
DiskFormat
 netmd, 11

eraseDisc

finalizeTOC
 netmd::CNetMdApi, 22
frame
 netmd::NetMdTime, 38
FW_TITLES
 netmd, 13

getDeviceName
 netmd::CNetMdApi, 22
Groups
 netmd, 10
groups
 netmd::CNetMdApi, 23

hour
 netmd::NetMdTime, 38
HW_TITLES
 netmd, 13

import
 netmd::CNetMdTOC, 32
INFO
 netmd, 12
initDevice
 netmd::CNetMdApi, 23

LP2
 netmd, 11
LP4
 netmd, 11

mEnd
 netmd::CNetMdTOC::DAOFragment, 34
mFirst
 netmd::Group, 37
mGid
 netmd::Group, 37
minute
 netmd::NetMdTime, 38
mLast
 netmd::Group, 37
mMinutes
 netmd::TrackTime, 39
mName
 netmd::Group, 37
moveTrack
 netmd::CNetMdApi, 23
mSeconds
 netmd::TrackTime, 39

- mStart
 - netmd::CNetMdTOC::DAOFragment, 34
- mTenthSecs
 - netmd::TrackTime, 39
- netmd, 9
 - AudioEncoding, 11
 - CAPTURE, 12
 - CRITICAL, 12
 - DEBUG, 12
 - DiskFormat, 11
 - FW_TITLES, 13
 - Groups, 10
 - HW_TITLES, 13
 - 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
 - NetMDByteVector, 11
 - NetMdErr, 12
 - NETMDERR_CMD_FAILED, 12
 - NETMDERR_CMD_INVALID, 12
 - NETMDERR_INTERIM, 12
 - NETMDERR_NO_ERROR, 12
 - NETMDERR_NOT_SUPPORTED, 12
 - NETMDERR_NOTREADY, 12
 - NETMDERR_OTHER, 12
 - NETMDERR_PARAM, 12
 - NETMDERR_TIMEOUT, 12
 - NETMDERR_USB, 12
 - NO_ONTHEFLY_CONVERSION, 11
 - operator<=, 13, 14
 - operator+=, 13
 - POS_ADDR, 13
 - PROTECTED, 12
 - SP, 11
 - TrackProtection, 12
 - TSTAMPS, 13
 - typelog, 12
 - UNKNOWN_2, 13
 - UNKNOWN, 11, 12
 - UNKNWN_1, 13
 - UNPROTECTED, 12
 - UTOCSector, 13
 - WARN, 12
- netmd::CNetMdApi, 17
 - ~CNetMdApi, 19
 - addTrackToGroup, 19
 - CNetMdApi, 19
 - createGroup, 19
 - deleteGroup, 20
 - deleteTrack, 20
 - delTrackFromGroup, 20
 - discCapacity, 21
 - discFlags, 21
 - discTitle, 21
 - eraseDisc, 22
 - finalizeTOC, 22
 - getDeviceName, 22
 - groups, 23
 - initDevice, 23
 - moveTrack, 23
 - otfEncodeSupported, 23
 - prepareTOCManip, 24
 - readUTOCSector, 24
 - sendAudioFile, 24
 - setDiscTitle, 25
 - setGroupTitle, 25
 - setLogLevel, 26
 - setLogStream, 26
 - setTrackTitle, 26
 - spUploadSupported, 27
 - tocManipSupported, 27
 - trackBitRate, 27
 - trackCount, 27
 - trackFlags, 28
 - trackTime, 28
 - trackTitle, 28
 - writeUTOCSector, 29
- netmd::CNetMdTOC, 29
 - ~CNetMdTOC, 31
 - addTrack, 31
 - CNetMdTOC, 31
 - DAOFragments, 30
 - discInfo, 32
 - discTitle, 32
 - import, 32
 - setDiscTitle, 32
 - trackCount, 33
 - trackInfo, 33
 - trackTitle, 33
- netmd::CNetMdTOC::DAOFragment, 34
 - mEnd, 34
 - mStart, 34
- netmd::DiscCapacity, 35
 - available, 35
 - recorded, 36
 - total, 36
- netmd::Group, 36
 - mFirst, 37
 - mGid, 37
 - mLast, 37
 - mName, 37
- netmd::NetMdTime, 37
 - frame, 38
 - hour, 38
 - minute, 38
 - second, 38
- netmd::toc, 15
- netmd::TrackTime, 39
 - mMinutes, 39
 - mSeconds, 39
 - mTenthSecs, 39

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](#)
 NetMDByteVector
 netmd, [11](#)
 NetMdErr
 netmd, [12](#)
 NETMDERR_CMD_FAILED
 netmd, [12](#)
 NETMDERR_CMD_INVALID
 netmd, [12](#)
 NETMDERR_INTERIM
 netmd, [12](#)
 NETMDERR_NO_ERROR
 netmd, [12](#)
 NETMDERR_NOT_SUPPORTED
 netmd, [12](#)
 NETMDERR_NOTREADY
 netmd, [12](#)
 NETMDERR_OTHER
 netmd, [12](#)
 NETMDERR_PARAM
 netmd, [12](#)
 NETMDERR_TIMEOUT
 netmd, [12](#)
 NETMDERR_USB
 netmd, [12](#)
 NO_ONTHEFLY_CONVERSION
 netmd, [11](#)

 operator<<
 netmd, [13](#), [14](#)
 operator+=
 netmd, [13](#)
 otfEncodeSupported
 netmd::CNetMdApi, [23](#)

 POS_ADDR
 netmd, [13](#)
 prepareTOCManip
 netmd::CNetMdApi, [24](#)
 PROTECTED
 netmd, [12](#)

 readUTOCSector
 netmd::CNetMdApi, [24](#)
 recorded
 netmd::DiscCapacity, [36](#)

 second
 netmd::NetMdTime, [38](#)
 sendAudioFile
 netmd::CNetMdApi, [24](#)
 setDiscTitle
 netmd::CNetMdApi, [25](#)
 netmd::CNetMdTOC, [32](#)
 setGroupTitle
 netmd::CNetMdApi, [25](#)
 setLogLevel
 netmd::CNetMdApi, [26](#)
 setLogStream
 netmd::CNetMdApi, [26](#)
 setTrackTitle
 netmd::CNetMdApi, [26](#)
 SP
 netmd, [11](#)
 spUploadSupported
 netmd::CNetMdApi, [27](#)

 tocManipSupported
 netmd::CNetMdApi, [27](#)
 total
 netmd::DiscCapacity, [36](#)
 trackBitRate
 netmd::CNetMdApi, [27](#)
 trackCount
 netmd::CNetMdApi, [27](#)
 netmd::CNetMdTOC, [33](#)
 trackFlags
 netmd::CNetMdApi, [28](#)
 trackInfo
 netmd::CNetMdTOC, [33](#)
 TrackProtection
 netmd, [12](#)
 trackTime
 netmd::CNetMdApi, [28](#)
 trackTitle
 netmd::CNetMdApi, [28](#)
 netmd::CNetMdTOC, [33](#)
 TSTAMPS
 netmd, [13](#)
 typelog
 netmd, [12](#)

 UNKNON_2
 netmd, [13](#)
 UNKNOWN
 netmd, [11](#), [12](#)
 UNKNWN_1
 netmd, [13](#)
 UNPROTECTED
 netmd, [12](#)
 UTOCSector
 netmd, [13](#)

 WARN
 netmd, [12](#)
 writeUTOCSector
 netmd::CNetMdApi, [29](#)