## ns3::PcapFileWrapper Class Reference

A class that wraps a PcapFile as an ns3::Object and provides a higher-layer ns-3 interface to the low-level public methods of PcapFile. More...

#include "pcap-file-wrapper.h"

- ▶ Inheritance diagram for ns3::PcapFileWrapper:
- ▶ Collaboration diagram for ns3::PcapFileWrapper:

	PcapFileWrapper ()
	~PcapFileWrapper ()
void	Clear (void)
	Clear all state bits of the underlying iostream. More
void	Close (void)
	Close the underlying pcap file. More
bool	Eof (void) const
bool	Fail (void) const
uint32_t	GetDataLinkType (void)
	Returns the data link type field of the pcap file as defined by the network field in the pcap global header. More.
uint32_t	GetMagic (void)
	Returns the magic number of the pcap file as defined by the magic_number field in the pcap global header.
	More
uint32_t	GetSigFigs (void)
	Returns the accuracy of timestamps field of the pcap file as defined by the sigfigs field in the pcap global
	header. More
uint32_t	GetSnapLen (void)
	Returns the max length of saved packets field of the pcap file as defined by the snaplen field in the pcap global
	header. More
int32_t	GetTimeZoneOffset (void)
	Returns the time zone offset of the pcap file as defined by the thiszone field in the pcap global header. More
uint16_t	GetVersionMajor (void)
	Returns the major version of the pcap file as defined by the version_major field in the pcap global header.
	More
uint16_t	GetVersionMinor (void)
	Returns the minor version of the pcap file as defined by the version_minor field in the pcap global header.
	More
void	<pre>Init (uint32_t dataLinkType, uint32_t snapLen=std::numeric_limits&lt; uint32_t &gt;::max(), int32_t</pre>
	tzCorrection=PcapFile::ZONE_DEFAULT)
	Initialize the pcap file associated with this wrapper. More

void Open (std::string const &filename, std::ios::openmode mode)

Create a new pcap file or open an existing pcap file. More...

Ptr< Packet > Read (Time &t)

Read the next packet from the file. More...

void Write (Time t, Ptr< const Packet > p)

Write the next packet to file. More...

void Write (Time t, const Header &header, Ptr< const Packet > p)

Write the provided header along with the packet to the pcap file. More...

void Write (Time t, uint8\_t const \*buffer, uint32\_t length)

Write the provided data buffer to the pcap file. More...

- ▶ Public Member Functions inherited from ns3::Object
- ▶ Public Member Functions inherited from ns3::SimpleRefCount< Object, ObjectBase, ObjectDeleter >
- ▶ Public Member Functions inherited from ns3::ObjectBase

### Static Public Member Functions

static Typeld GetTypeld (void)

Get the type ID. More...

- ▶ Static Public Member Functions inherited from ns3::Object
- ▶ Static Public Member Functions inherited from ns3::ObjectBase

### **Private Attributes**

PcapFile m\_file

Pcap file. More...

bool m nanosecMode

Timestamps in nanosecond mode. More...

uint32\_t m\_snapLen

max length of saved packets More...

### Additional Inherited Members

- ▶ Protected Member Functions inherited from ns3::Object
- ▶ Protected Member Functions inherited from ns3::ObjectBase
- ▶ Related Functions inherited from ns3::ObjectBase

### **Detailed Description**

A class that wraps a PcapFile as an ns3::Object and provides a higher-layer ns-3 interface to the low-level public methods of PcapFile.

Introspection did not find any typical Config paths.

Users are encouraged to use this object instead of class ns3::PcapFile in ns-3 public APIs.

#### **Attributes**

• CaptureSize: Maximum length of captured packets (cf. pcap snaplen)

Set with class: ns3::UintegerValue

o Underlying type: uint32\_t 0:65535

o Initial value: 65535

• Flags: construct write read

• NanosecMode: Whether packet timestamps in the PCAP file are nanoseconds or microseconds(default).

Set with class: BooleanValue

Underlying type: bool

o Initial value: false

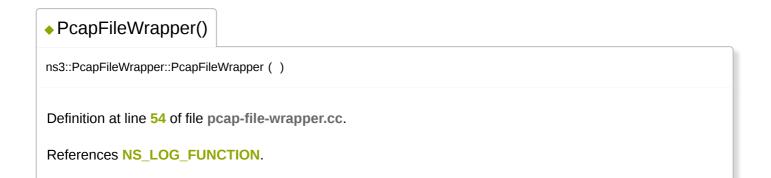
• Flags: construct write read

No TraceSources are defined for this type.

Size of this type is 608 bytes (on a 64-bit architecture).

Definition at line 39 of file pcap-file-wrapper.h.

### Constructor & Destructor Documentation



# ~PcapFileWrapper()

ns3::PcapFileWrapper::~PcapFileWrapper ( )

Definition at line 59 of file pcap-file-wrapper.cc.

References Close(), and NS\_LOG\_FUNCTION.

▶ Here is the call graph for this function:

### Member Function Documentation



void ns3::PcapFileWrapper::Clear (void )

Clear all state bits of the underlying iostream.

Definition at line 79 of file pcap-file-wrapper.cc.

References ns3::PcapFile::Clear(), m\_file, and NS\_LOG\_FUNCTION.

▶ Here is the call graph for this function:

## Close()

void ns3::PcapFileWrapper::Close ( void )

Close the underlying pcap file.

Definition at line 86 of file pcap-file-wrapper.cc.

References ns3::PcapFile::Close(), m\_file, and NS\_LOG\_FUNCTION.

Referenced by ~PcapFileWrapper().

▶ Here is the call graph for this function:

▶ Here is the caller graph for this function:

# ◆ Eof()

bool ns3::PcapFileWrapper::Eof (void ) const

#### Returns

true if the 'eof' bit is set in the underlying iostream, false otherwise.

Definition at line 73 of file pcap-file-wrapper.cc.

References ns3::PcapFile::Eof(), m\_file, and NS\_LOG\_FUNCTION.



bool ns3::PcapFileWrapper::Fail (void ) const

#### Returns

true if the 'fail' bit is set in the underlying iostream, false otherwise.

Definition at line 66 of file pcap-file-wrapper.cc.

References ns3::PcapFile::Fail(), m\_file, and NS\_LOG\_FUNCTION.

▶ Here is the call graph for this function:

### GetDataLinkType()

uint32\_t ns3::PcapFileWrapper::GetDataLinkType ( void )

Returns the data link type field of the pcap file as defined by the network field in the pcap global header.

See http://wiki.wireshark.org/Development/LibpcapFileFormat

#### Returns

data link type field

Definition at line 253 of file pcap-file-wrapper.cc.

References ns3::PcapFile::GetDataLinkType(), m\_file, and NS\_LOG\_FUNCTION.

▶ Here is the call graph for this function:

# GetMagic()

uint32\_t ns3::PcapFileWrapper::GetMagic ( void )

Returns the magic number of the pcap file as defined by the magic\_number field in the pcap global header.

See http://wiki.wireshark.org/Development/LibpcapFileFormat

#### Returns

magic number

Definition at line 211 of file pcap-file-wrapper.cc.

References ns3::PcapFile::GetMagic(), m\_file, and NS\_LOG\_FUNCTION.

▶ Here is the call graph for this function:

# GetSigFigs()

uint32 t ns3::PcapFileWrapper::GetSigFigs ( void )

Returns the accuracy of timestamps field of the pcap file as defined by the sigfigs field in the pcap global header.

See http://wiki.wireshark.org/Development/LibpcapFileFormat

#### Returns

accuracy of timestamps

Definition at line 239 of file pcap-file-wrapper.cc.

References ns3::PcapFile::GetSigFigs(), m\_file, and NS\_LOG\_FUNCTION.

▶ Here is the call graph for this function:

### GetSnapLen()

uint32\_t ns3::PcapFileWrapper::GetSnapLen ( void )

Returns the max length of saved packets field of the pcap file as defined by the snaplen field in the pcap global header.

See http://wiki.wireshark.org/Development/LibpcapFileFormat

#### Returns

max length of saved packets field

Definition at line 246 of file pcap-file-wrapper.cc.

References ns3::PcapFile::GetSnapLen(), m\_file, and NS\_LOG\_FUNCTION.

▶ Here is the call graph for this function:

# ◆ GetTimeZoneOffset()

int32\_t ns3::PcapFileWrapper::GetTimeZoneOffset ( void )

Returns the time zone offset of the pcap file as defined by the thiszone field in the pcap global header.

See http://wiki.wireshark.org/Development/LibpcapFileFormat

### Returns

time zone offset

Definition at line 232 of file pcap-file-wrapper.cc.

References ns3::PcapFile::GetTimeZoneOffset(), m\_file, and NS\_LOG\_FUNCTION.

▶ Here is the call graph for this function:

# GetTypeId()

Typeld ns3::PcapFileWrapper::GetTypeld (void )

static

Get the type ID.

#### Returns

the object TypeId

Definition at line 33 of file pcap-file-wrapper.cc.

References m\_nanosecMode, m\_snapLen, ns3::MakeBooleanAccessor(),

ns3::MakeBooleanChecker(), ns3::MakeUintegerAccessor(), ns3::TypeId::SetParent(), and

ns3::PcapFile::SNAPLEN\_DEFAULT.

▶ Here is the call graph for this function:

# GetVersionMajor()

uint16\_t ns3::PcapFileWrapper::GetVersionMajor ( void )

Returns the major version of the pcap file as defined by the version\_major field in the pcap global header.

See http://wiki.wireshark.org/Development/LibpcapFileFormat

#### Returns

major version

Definition at line 218 of file pcap-file-wrapper.cc.

References ns3::PcapFile::GetVersionMajor(), m\_file, and NS\_LOG\_FUNCTION.

▶ Here is the call graph for this function:

# GetVersionMinor()

uint16\_t ns3::PcapFileWrapper::GetVersionMinor ( void )

Returns the minor version of the pcap file as defined by the version\_minor field in the pcap global header.

See http://wiki.wireshark.org/Development/LibpcapFileFormat

#### Returns

minor version

Definition at line 225 of file pcap-file-wrapper.cc.

References ns3::PcapFile::GetVersionMinor(), m\_file, and NS\_LOG\_FUNCTION.



Initialize the pcap file associated with this wrapper.

This file must have been previously opened with write permissions.

#### **Parameters**

dataLinkType A data link type as defined in the pcap library. If you want to make resulting pcap files visible in existing tools, the data link type must match existing definitions, such as PCAP\_ETHERNET, PCAP\_PPP, PCAP\_80211, etc. If you are storing different kinds of packet data, such as naked TCP headers, you are at liberty to locally define your own data link types. According to the pcap-linktype man page, "well-known" pcap linktypes range from 0 to 177. If you use a large random number for your type, chances are small for a collision.

**snapLen** An optional maximum size for packets written to the file. Defaults to 65535. If packets exceed this length they are truncated.

**tzCorrection** An integer describing the offset of your local time zone from UTC/GMT. For example, Pacific Standard

Time in the US is GMT-8, so one would enter -8 for that correction. Defaults to 0 (UTC).

### Warning

Calling this method on an existing file will result in the loss any existing data.

Definition at line **100** of file **pcap-file-wrapper.cc**.

References ns3::PcapFile::Init(), m\_file, m\_nanosecMode, m\_snapLen, max, and NS\_LOG\_FUNCTION.



Create a new pcap file or open an existing pcap file.

Semantics are similar to the stdc++ io stream classes.

Since a pcap file is always a binary file, the file type is automatically selected as a binary file (fstream::binary is automatically ored with the mode field).

#### **Parameters**

filename String containing the name of the file.

**mode** String containing the access mode for the file.

Definition at line 93 of file pcap-file-wrapper.cc.

References m\_file, NS\_LOG\_FUNCTION, and ns3::PcapFile::Open().

▶ Here is the call graph for this function:

# Read()

Ptr< Packet > ns3::PcapFileWrapper::Read ( Time & t )

Read the next packet from the file.

#### **Parameters**

t Reference to packet timestamp as ns3::Time.

#### **Returns**

a pointer to ns3::Packet.

Definition at line 179 of file pcap-file-wrapper.cc.

References ns3::PcapFile::Fail(), ns3::PcapFile::IsNanoSecMode(), m\_file, ns3::MicroSeconds(), ns3::NanoSeconds(), and ns3::PcapFile::Read().

▶ Here is the call graph for this function:

Write() [1/3]

```
void ns3::PcapFileWrapper::Write ( Time t,

Ptr< const Packet > p

)
```

Write the next packet to file.

#### **Parameters**

- t Packet timestamp as ns3::Time.
- p Packet to write to the pcap file.

Definition at line 119 of file pcap-file-wrapper.cc.

```
References ns3::Time::GetMicroSeconds(), ns3::Time::GetNanoSeconds(), ns3::PcapFile::IsNanoSecMode(), m_file, NS_LOG_FUNCTION, and ns3::PcapFile::Write().
```

▶ Here is the call graph for this function:

```
Write() [2/3]
```

```
void ns3::PcapFileWrapper::Write(Time t,

const Header & header,

Ptr< const Packet > p

)
```

Write the provided header along with the packet to the pcap file.

It is the case that adding a header to a packet prior to writing it to a file must trigger a deep copy in the **Packet**. By providing the header separately, we can avoid that copy.

#### **Parameters**

t Packet timestamp as ns3::Time.

**header** The **Header** to prepend to the packet.

p Packet to write to the pcap file.

Definition at line 139 of file pcap-file-wrapper.cc.

```
References ns3::Time::GetMicroSeconds(), ns3::Time::GetNanoSeconds(), ns3::PcapFile::IsNanoSecMode(), m_file, NS_LOG_FUNCTION, and ns3::PcapFile::Write().
```

```
◆ Write() [3/3]
```

```
void ns3::PcapFileWrapper::Write ( Time t, uint8_t const * buffer, uint32_t length )

Write the provided data buffer to the pcap file.

Parameters

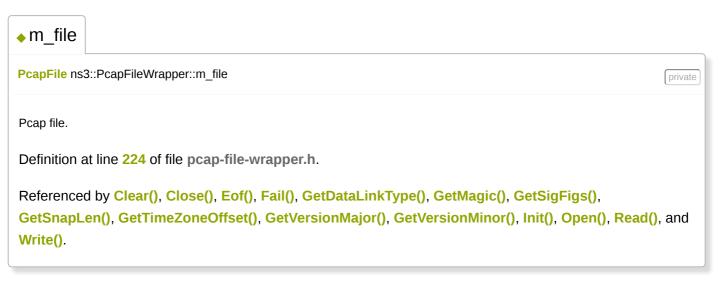
t Packet timestamp as ns3::Time.
buffer The buffer to write.
length The size of the buffer.

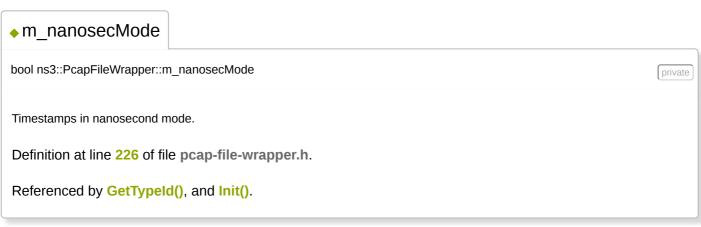
Definition at line 159 of file pcap-file-wrapper.cc.

References ns3::Time::GetMicroSeconds(), ns3::Time::GetNanoSeconds(), ns3::PcapFile::IsNanoSecMode(), m_file, NS_LOG_FUNCTION, and ns3::PcapFile::Write().

Here is the call graph for this function:
```

### Member Data Documentation





uint32\_t ns3::PcapFileWrapper::m\_snapLen

max length of saved packets

Definition at line 225 of file pcap-file-wrapper.h.

Referenced by GetTypeId(), and Init().

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

- src/network/utils/pcap-file-wrapper.h
- src/network/utils/pcap-file-wrapper.cc