Namespace NetOpt

Classes

Compressor

Compress or decompress packet buffers and their packet deltas.

DeltaCoder

Encode or decode deltas between two packet buffers.

DllImportApi

Native

Managed to unmanaged interaction

PacketBuffer

Buffer for writing and reading packet data.

PacketDelta

Delta for managing differences between packet buffers.

PacketReader

Read packet data from packet buffers

PacketWriter

Write packet data to packet buffers

Interfaces

INativeApi

Unmanaged calls supported by managed environment

Enums

Compressor.Algorithm

Compression algorithm to use for compression or decompression.

DeltaCoder.Algorithm

Delta compression algorithm to use for encoding or decoding.

Native.InteropApi

Defines how managed to unmanaged calls are executed

Class Compressor

Compress or decompress packet buffers and their packet deltas.

Inheritance

System.Object
Compressor

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public static class Compressor
```

Methods

Compress(PacketBuffer, Int32, Compressor.Algorithm)

Constructs a compressed buffer from a buffer

Declaration

```
public static PacketBuffer Compress(PacketBuffer buffer, int bytes, Compressor.Algorithm algorithm =
Compressor.Algorithm.LZ4F)
```

Parameters

TYPE	NAME	DESCRIPTION
PacketBuffer	buffer	Buffer instance
System.Int32	bytes	Number of bytes to compress from buffer
Compressor.Algorithm	algorithm	Compression algorithm to use for compression

Returns

TYPE	DESCRIPTION
PacketBuffer	Compressed buffer instance

Compress(PacketDelta, Int32, Compressor.Algorithm)

Constructs a compressed delta from a delta

Declaration

```
public static PacketDelta Compress(PacketDelta delta, int bytes, Compressor.Algorithm algorithm =
Compressor.Algorithm.LZ4F)
```

Parameters

TYPE	NAME	DESCRIPTION
PacketDelta	delta	Delta instance
System.Int32	bytes	Number of bytes to compress from delta
Compressor.Algorithm	algorithm	Compression algorithm to use for compression

Returns

TYPE	DESCRIPTION
PacketDelta	Compressed delta instance

Decompress(PacketBuffer, Int32, Compressor.Algorithm)

Constructs a decompressed buffer from a compressed buffer

Declaration

```
public static PacketBuffer Decompress(PacketBuffer compressedBuffer, int bytes = 0, Compressor.Algorithm
algorithm = Compressor.Algorithm.LZ4F)
```

Parameters

TYPE	NAME	DESCRIPTION
PacketBuffer	compressedBuffer	Compressed buffer instance
System.Int32	bytes	Number of compressed bytes in compressed buffer
Compressor.Algorithm	algorithm	Compression algorithm used for compression

Returns

TYPE	DESCRIPTION
PacketBuffer	Decompressed buffer instance

Decompress(PacketDelta, Int32, Compressor.Algorithm)

Constructs a decompressed delta from a compressed delta

Declaration

```
public static PacketDelta Decompress(PacketDelta delta, int bytes, Compressor.Algorithm algorithm =
Compressor.Algorithm.LZ4F)
```

Parameters

TYPE	NAME	DESCRIPTION
PacketDelta	delta	Compressed delta instance
System.Int32	bytes	Number of compressed bytes in compressed delta
Compressor.Algorithm	algorithm	Compression algorithm used for compression

Returns

TYPE	DESCRIPTION
PacketDelta	Decompressed delta instance

Enum Compressor.Algorithm

Compression algorithm to use for compression or decompression.

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public enum Algorithm
```

Fields

NAME	DESCRIPTION
BCM	BCM is a high-performance file compressor.
BROTLI11	Brotli is a generic-purpose lossless compression algorithm.
BROTLI9	Brotli is a generic-purpose lossless compression algorithm.
BSC	
BZIP2	bzip2 compresses files using block sorting text compression, generally considerably better than LZ77/LZ78-based compressors.
CRUSH	CRUSH is a simple LZ77-based file compressor that features an extremely fast decompression.
CSC20	Loss-less data compression algorithm inspired by LZMA.
LZ4	LZ4 is lossless compression algorithm, providing high compression speed.
LZ4F	LZ4 is lossless compression algorithm, providing high compression speed.
LZIP	Lzip is a lossless data compressor based on the LZMA algorithm.
LZJB	LZJB is a lossless data compression algorithm to compress crash dumps and data.
LZMA20	LZMA is an algorithm used to perform lossless data compression.
LZMA25	LZMA is an algorithm used to perform lossless data compression.

NAME	DESCRIPTION
MCM	MCM compressor, context mixing and lzp.
MINIZ	Miniz is a lossless, high performance data compression library.
RAW	No compression.
SHOCO	Optimized for *very* short strings of english words.
SHRINKER	LZ77-based data compression program that can be used in high performance demanding environments.
TANGELO	
ZLING	Zling is an improved lightweight compression utility and library.
ZMOLLY	Zmolly is a generic data compressor with high compression ratio.
ZPAQ	ZPAQ is optimized for fast compression rather than decompression.
ZSTD	Zstandard is a real-time compression algorithm, providing high compression ratios.
ZSTDF	Zstandard is a real-time compression algorithm, providing high compression ratios.

Class DeltaCoder

Encode or decode deltas between two packet buffers.

Inheritance

System.Object
DeltaCoder

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public static class DeltaCoder
```

Methods

Decode(PacketBuffer, PacketDelta, DeltaCoder.Algorithm)

Constructs a target buffer from a source buffer and delta.

Declaration

```
public static PacketBuffer Decode(PacketBuffer source, PacketDelta delta, DeltaCoder.Algorithm algorithm =
DeltaCoder.Algorithm.HDiffPatch)
```

Parameters

TYPE	NAME	DESCRIPTION
PacketBuffer	source	Transformation source
PacketDelta	delta	Delta to use for transformation
DeltaCoder.Algorithm	algorithm	Delta compression algorithm used for encoding

Returns

TYPE	DESCRIPTION
PacketBuffer	Target buffer constructed using source buffer and delta

Encode(PacketBuffer, PacketBuffer, DeltaCoder.Algorithm)

Constructs a delta from two buffers.

Declaration

```
public static PacketDelta Encode(PacketBuffer source, PacketBuffer target, DeltaCoder.Algorithm algorithm =
DeltaCoder.Algorithm.HDiffPatch)
```

Parameters

TYPE	NAME	DESCRIPTION
PacketBuffer	source	Transformation source
PacketBuffer	target	Transformation target
DeltaCoder.Algorithm	algorithm	Delta compression algorithm to use for encoding

Returns

TYPE	DESCRIPTION
PacketDelta	Delta instance that transforms source buffer into target buffer

Enum DeltaCoder.Algorithm

Delta compression algorithm to use for encoding or decoding.

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public enum Algorithm
```

Fields

NAME	DESCRIPTION
BsDiff	Intended for building and applying patches to binary executable files. NOTE: requires secondary compression through Compressor.
HDiffPatch	Binary data build and patch, fast with small delta/differential. NOTE: Very effective for data with low variability.

Class DllImportApi

Inheritance

System.Object
DllImportApi

Implements

INativeApi

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public class DllImportApi : INativeApi
```

Methods

AllocPacketDelta(IntPtr, Int32)

Declaration

```
public IntPtr AllocPacketDelta(IntPtr data, int bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	data	
System.Int32	bytes	

Returns

TYPE	DESCRIPTION
System.IntPtr	

CompressPacketBuffer(IntPtr, Int32, Compressor.Algorithm)

Declaration

```
public IntPtr CompressPacketBuffer(IntPtr buffer, int bytes, Compressor.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	buffer	
System.Int32	bytes	

TYPE	NAME	DESCRIPTION
Compressor.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

CompressPacketDelta(IntPtr, Int32, Compressor.Algorithm)

Declaration

```
public IntPtr CompressPacketDelta(IntPtr data, int bytes, Compressor.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	data	
System.Int32	bytes	
Compressor.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

CopyPacketBuffer(IntPtr, IntPtr, Int32)

Declaration

```
public void CopyPacketBuffer(IntPtr buffer, IntPtr destination, int bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	buffer	
System.IntPtr	destination	
System.Int32	bytes	

CopyPacketDelta(IntPtr, IntPtr, Int32)

Declaration

```
public void CopyPacketDelta(IntPtr delta, IntPtr destination, int bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	delta	
System.IntPtr	destination	
System.Int32	bytes	

DecompressPacketBuffer(IntPtr, Int32, Compressor.Algorithm)

Declaration

```
public IntPtr DecompressPacketBuffer(IntPtr buffer, int bytes, Compressor.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	buffer	
System.Int32	bytes	
Compressor.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

DecompressPacketDelta(IntPtr, Int32, Compressor.Algorithm)

Declaration

```
public IntPtr DecompressPacketDelta(IntPtr data, int bytes, Compressor.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	data	
System.Int32	bytes	
Compressor.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

DifferencerDecodeDelta(IntPtr, IntPtr, DeltaCoder.Algorithm)

Declaration

```
public IntPtr DifferencerDecodeDelta(IntPtr source, IntPtr delta, DeltaCoder.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	source	
System.IntPtr	delta	
DeltaCoder.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

DifferencerEncodeBuffer(IntPtr, IntPtr, DeltaCoder.Algorithm)

Declaration

```
public IntPtr DifferencerEncodeBuffer(IntPtr source, IntPtr buffer, DeltaCoder.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	source	
System.IntPtr	buffer	
DeltaCoder.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

FreePacketDelta(IntPtr)

Declaration

```
public void FreePacketDelta(IntPtr delta)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	delta	

GetPacketBufferSize(IntPtr)

Declaration

```
public int GetPacketBufferSize(IntPtr buffer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	buffer	

Returns

TYPE	DESCRIPTION
System.Int32	

GetPacketDeltaSize(IntPtr)

Declaration

```
public int GetPacketDeltaSize(IntPtr delta)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	delta	

Returns

TYPE	DESCRIPTION
System.Int32	

PackBool(IntPtr, Boolean)

Declaration

```
public void PackBool(IntPtr writeStream, bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Boolean	value	

PackDouble(IntPtr, Double)

Declaration

```
public void PackDouble(IntPtr writeStream, double value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Double	value	

PackDoubleCompressed(IntPtr, Double, Double, Double, Double)

Declaration

```
public void PackDoubleCompressed(IntPtr writeStream, double value, double min, double max, double precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Double	value	
System.Double	min	
System.Double	max	
System.Double	precision	

PackFloat(IntPtr, Single)

Declaration

```
public void PackFloat(IntPtr writeStream, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Single	value	

PackFloatCompressed(IntPtr, Single, Single, Single, Single)

Declaration

```
public void PackFloatCompressed(IntPtr writeStream, float value, float min, float max, float precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Single	value	
System.Single	min	
System.Single	max	
System.Single	precision	

PackInt16(IntPtr, Int16)

Declaration

```
public void PackInt16(IntPtr writeStream, short value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int16	value	

PackInt16Quantized(IntPtr, Int16, Int16, Int16)

Declaration

```
public void PackInt16Quantized(IntPtr writeStream, short value, short min, short max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int16	value	
System.Int16	min	
System.Int16	max	

PackInt32(IntPtr, Int32)

Declaration

```
public void PackInt32(IntPtr writeStream, int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int32	value	

PackInt32Quantized(IntPtr, Int32, Int32, Int32)

Declaration

```
public void PackInt32Quantized(IntPtr writeStream, int value, int min, int max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int32	value	
System.Int32	min	
System.Int32	max	

PackInt64(IntPtr, Int64)

Declaration

```
public void PackInt64(IntPtr writeStream, long value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int64	value	

PackInt64Quantized(IntPtr, Int64, Int64, Int64)

Declaration

```
public void PackInt64Quantized(IntPtr writeStream, long value, long min, long max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int64	value	
System.Int64	min	
System.Int64	max	

PackInt8(IntPtr, SByte)

Declaration

```
public void PackInt8(IntPtr writeStream, sbyte value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.SByte	value	

PackInt8Quantized(IntPtr, SByte, SByte, SByte)

Declaration

```
public void PackInt8Quantized(IntPtr writeStream, sbyte value, sbyte min, sbyte max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.SByte	value	
System.SByte	min	

TYPE	NAME	DESCRIPTION
System.SByte	max	

PackUInt16(IntPtr, UInt16)

Declaration

```
public void PackUInt16(IntPtr writeStream, ushort value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.UInt16	value	

PackUInt32(IntPtr, UInt32)

Declaration

```
public void PackUInt32(IntPtr writeStream, uint value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.UInt32	value	

PackUInt64(IntPtr, UInt64)

Declaration

```
public void PackUInt64(IntPtr writeStream, ulong value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.UInt64	value	

PackUInt8(IntPtr, Byte)

Declaration

```
public void PackUInt8(IntPtr writeStream, byte value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Byte	value	

UnpackBool(IntPtr)

Declaration

```
public bool UnpackBool(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Boolean	

UnpackDouble(IntPtr)

Declaration

```
public double UnpackDouble(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Double	

UnpackDoubleCompressed(IntPtr, Double, Double, Double)

Declaration

```
public double UnpackDoubleCompressed(IntPtr readStream, double min, double max, double precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.Double	min	
System.Double	max	
System.Double	precision	

Returns

TYPE	DESCRIPTION
System.Double	

UnpackFloat(IntPtr)

Declaration

```
public float UnpackFloat(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Single	

UnpackFloatCompressed(IntPtr, Single, Single, Single)

Declaration

```
public float UnpackFloatCompressed(IntPtr readStream, float min, float max, float precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.Single	min	
System.Single	max	
System.Single	precision	

Returns

TYPE	DESCRIPTION
System.Single	

UnpackInt16(IntPtr)

Declaration

```
public short UnpackInt16(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Int16	

UnpackInt16Quantized(IntPtr, Int16, Int16)

Declaration

```
public short UnpackInt16Quantized(IntPtr readStream, short min, short max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.Int16	min	
System.Int16	max	

Returns

TYPE	DESCRIPTION
System.Int16	

UnpackInt32(IntPtr)

Declaration

```
public int UnpackInt32(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Int32	

UnpackInt32Quantized(IntPtr, Int32, Int32)

Declaration

```
public int UnpackInt32Quantized(IntPtr readStream, int min, int max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.Int32	min	
System.Int32	max	

Returns

TYPE	DESCRIPTION
System.Int32	

UnpackInt64(IntPtr)

Declaration

```
public long UnpackInt64(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Int64	

UnpackInt64Quantized(IntPtr, Int64, Int64)

Declaration

```
public long UnpackInt64Quantized(IntPtr readStream, long min, long max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.Int64	min	
System.Int64	max	

Returns

TYPE	DESCRIPTION
System.Int64	

UnpackInt8(IntPtr)

Declaration

```
public sbyte UnpackInt8(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.SByte	

UnpackInt8Quantized(IntPtr, SByte, SByte)

Declaration

```
public sbyte UnpackInt8Quantized(IntPtr readStream, sbyte min, sbyte max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.SByte	min	
System.SByte	max	

Returns

TYPE	DESCRIPTION
System.SByte	

UnpackUInt16(IntPtr)

Declaration

```
public ushort UnpackUInt16(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.UInt16	

UnpackUInt32(IntPtr)

Declaration

```
public uint UnpackUInt32(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.UInt32	

UnpackUInt64(IntPtr)

Declaration

```
public ulong UnpackUInt64(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.UInt64	

UnpackUInt8(IntPtr)

Declaration

```
public byte UnpackUInt8(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Byte	

Explicit Interface Implementations

INativeApi.AllocPacketBuffer(Int32)

Declaration

```
IntPtr INativeApi.AllocPacketBuffer(int bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	bytes	

Returns

TYPE	DESCRIPTION
System.IntPtr	

INativeApi.AllocPacketBufferUnmanaged(IntPtr, Int32)

Declaration

```
IntPtr INativeApi.AllocPacketBufferUnmanaged(IntPtr buffer, int bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	buffer	
System.Int32	bytes	

Returns

TYPE	DESCRIPTION
System.IntPtr	

INativeApi.AllocPacketReader(IntPtr)

Declaration

```
IntPtr INativeApi.AllocPacketReader(IntPtr packetBuffer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetBuffer	

Returns

TYPE	DESCRIPTION
System.IntPtr	

INativeApi.AllocPacketWriter(IntPtr)

Declaration

```
IntPtr INativeApi.AllocPacketWriter(IntPtr packetBuffer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetBuffer	

Returns

TYPE	DESCRIPTION
System.IntPtr	

INativeApi.FlushPacketWriter(IntPtr)

Declaration

```
int INativeApi.FlushPacketWriter(IntPtr packetWriter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetWriter	

Returns

TYPE	DESCRIPTION
System.Int32	

INativeApi.FreePacketBuffer(IntPtr)

Declaration

```
void INativeApi.FreePacketBuffer(IntPtr packetBuffer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetBuffer	

INativeApi.FreePacketReader(IntPtr)

Declaration

```
void INativeApi.FreePacketReader(IntPtr packetReader)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetReader	

INativeApi.FreePacketWriter(IntPtr)

Declaration

```
void INativeApi.FreePacketWriter(IntPtr packetWriter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetWriter	

INativeApi.GetBitsProcessedPacketWriter(IntPtr)

Declaration

```
int INativeApi.GetBitsProcessedPacketWriter(IntPtr packetWriter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetWriter	

Returns

TYPE	DESCRIPTION
System.Int32	

INativeApi.GetBytesProcessedPacketWriter(IntPtr)

Declaration

```
int INativeApi.GetBytesProcessedPacketWriter(IntPtr packetWriter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetWriter	

Returns

TYPE	DESCRIPTION
System.Int32	

Implements

INativeApi

Interface INativeApi

Unmanaged calls supported by managed environment

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public interface INativeApi
```

Methods

AllocPacketBuffer(Int32)

Declaration

```
IntPtr AllocPacketBuffer(int bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	bytes	

Returns

TYPE	DESCRIPTION
System.IntPtr	

AllocPacketBufferUnmanaged(IntPtr, Int32)

Declaration

```
IntPtr AllocPacketBufferUnmanaged(IntPtr buffer, int bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	buffer	
System.Int32	bytes	

Returns

TYPE	DESCRIPTION
System.IntPtr	

AllocPacketDelta(IntPtr, Int32)

Declaration

```
IntPtr AllocPacketDelta(IntPtr data, int bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	data	
System.Int32	bytes	

Returns

TYPE	DESCRIPTION
System.IntPtr	

AllocPacketReader(IntPtr)

Declaration

```
IntPtr AllocPacketReader(IntPtr packetBuffer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetBuffer	

Returns

TYPE	DESCRIPTION
System.IntPtr	

AllocPacketWriter(IntPtr)

Declaration

```
IntPtr AllocPacketWriter(IntPtr packetBuffer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetBuffer	

Returns

TYPE	DESCRIPTION
System.IntPtr	

CompressPacketBuffer(IntPtr, Int32, Compressor.Algorithm)

Declaration

```
IntPtr CompressPacketBuffer(IntPtr buffer, int bytes, Compressor.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	buffer	

TYPE	NAME	DESCRIPTION
System.Int32	bytes	
Compressor.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

CompressPacketDelta(IntPtr, Int32, Compressor.Algorithm)

Declaration

```
IntPtr CompressPacketDelta(IntPtr data, int bytes, Compressor.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	data	
System.Int32	bytes	
Compressor.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

CopyPacketBuffer(IntPtr, IntPtr, Int32)

Declaration

```
void CopyPacketBuffer(IntPtr buffer, IntPtr destination, int bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	buffer	
System.IntPtr	destination	
System.Int32	bytes	

CopyPacketDelta(IntPtr, IntPtr, Int32)

Declaration

```
void CopyPacketDelta(IntPtr delta, IntPtr destination, int bytes)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	delta	
System.IntPtr	destination	
System.Int32	bytes	

DecompressPacketBuffer(IntPtr, Int32, Compressor.Algorithm)

Declaration

```
IntPtr DecompressPacketBuffer(IntPtr buffer, int bytes, Compressor.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	buffer	
System.Int32	bytes	
Compressor.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

DecompressPacketDelta(IntPtr, Int32, Compressor.Algorithm)

Declaration

```
IntPtr DecompressPacketDelta(IntPtr data, int bytes, Compressor.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	data	
System.Int32	bytes	
Compressor.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

DifferencerDecodeDelta(IntPtr, IntPtr, DeltaCoder.Algorithm)

Declaration

```
IntPtr DifferencerDecodeDelta(IntPtr source, IntPtr delta, DeltaCoder.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	source	
System.IntPtr	delta	
DeltaCoder.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

DifferencerEncodeBuffer(IntPtr, IntPtr, DeltaCoder.Algorithm)

Declaration

```
IntPtr DifferencerEncodeBuffer(IntPtr source, IntPtr target, DeltaCoder.Algorithm algorithm)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	source	
System.IntPtr	target	
DeltaCoder.Algorithm	algorithm	

Returns

TYPE	DESCRIPTION
System.IntPtr	

FlushPacketWriter(IntPtr)

Declaration

```
int FlushPacketWriter(IntPtr packetWriter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetWriter	

Returns

TYPE	DESCRIPTION
System.Int32	

FreePacketBuffer(IntPtr)

Declaration

```
void FreePacketBuffer(IntPtr packetBuffer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetBuffer	

FreePacketDelta(IntPtr)

Declaration

```
void FreePacketDelta(IntPtr delta)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	delta	

FreePacketReader(IntPtr)

Declaration

```
void FreePacketReader(IntPtr packetReader)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetReader	

FreePacketWriter(IntPtr)

Declaration

```
void FreePacketWriter(IntPtr packetWriter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetWriter	

GetBitsProcessedPacketWriter(IntPtr)

Declaration

```
int GetBitsProcessedPacketWriter(IntPtr packetWriter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetWriter	

Returns

TYPE	DESCRIPTION
System.Int32	

GetBytesProcessedPacketWriter(IntPtr)

Declaration

```
int GetBytesProcessedPacketWriter(IntPtr packetWriter)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	packetWriter	

Returns

TYPE	DESCRIPTION
System.Int32	

GetPacketBufferSize(IntPtr)

Declaration

```
int GetPacketBufferSize(IntPtr buffer)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	buffer	

Returns

TYPE	DESCRIPTION
System.Int32	

GetPacketDeltaSize(IntPtr)

Declaration

```
int GetPacketDeltaSize(IntPtr delta)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	delta	

Returns

TYPE	DESCRIPTION
System.Int32	

PackBool(IntPtr, Boolean)

Declaration

```
void PackBool(IntPtr writeStream, bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Boolean	value	

PackDouble(IntPtr, Double)

Declaration

```
void PackDouble(IntPtr writeStream, double value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Double	value	

PackDoubleCompressed(IntPtr, Double, Double, Double, Double)

Declaration

```
void PackDoubleCompressed(IntPtr writeStream, double value, double min, double max, double precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Double	value	
System.Double	min	
System.Double	max	
System.Double	precision	

PackFloat(IntPtr, Single)

Declaration

```
void PackFloat(IntPtr writeStream, float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Single	value	

PackFloatCompressed(IntPtr, Single, Single, Single, Single)

Declaration

```
void PackFloatCompressed(IntPtr writeStream, float value, float min, float max, float precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Single	value	
System.Single	min	
System.Single	max	
System.Single	precision	

PackInt16(IntPtr, Int16)

Declaration

```
void PackInt16(IntPtr writeStream, short value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int16	value	

PackInt16Quantized(IntPtr, Int16, Int16, Int16)

Declaration

```
void PackInt16Quantized(IntPtr writeStream, short value, short min, short max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int16	value	
System.Int16	min	
System.Int16	max	

PackInt32(IntPtr, Int32)

Declaration

```
void PackInt32(IntPtr writeStream, int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int32	value	

PackInt32Quantized(IntPtr, Int32, Int32, Int32)

Declaration

```
void PackInt32Quantized(IntPtr writeStream, int value, int min, int max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int32	value	
System.Int32	min	
System.Int32	max	

PackInt64(IntPtr, Int64)

Declaration

```
void PackInt64(IntPtr writeStream, long value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int64	value	

PackInt64Quantized(IntPtr, Int64, Int64, Int64)

Declaration

```
void PackInt64Quantized(IntPtr writeStream, long value, long min, long max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Int64	value	
System.Int64	min	
System.Int64	max	

PackInt8(IntPtr, SByte)

Declaration

```
void PackInt8(IntPtr writeStream, sbyte value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.SByte	value	

PackInt8Quantized(IntPtr, SByte, SByte, SByte)

Declaration

```
void PackInt8Quantized(IntPtr writeStream, sbyte value, sbyte min, sbyte max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.SByte	value	
System.SByte	min	
System.SByte	max	

PackUInt16(IntPtr, UInt16)

Declaration

```
void PackUInt16(IntPtr writeStream, ushort value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.UInt16	value	

PackUInt32(IntPtr, UInt32)

Declaration

```
void PackUInt32(IntPtr writeStream, uint value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.UInt32	value	

PackUInt64(IntPtr, UInt64)

Declaration

```
void PackUInt64(IntPtr writeStream, ulong value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.UInt64	value	

PackUInt8(IntPtr, Byte)

Declaration

```
void PackUInt8(IntPtr writeStream, byte value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	writeStream	
System.Byte	value	

UnpackBool(IntPtr)

Declaration

```
bool UnpackBool(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Boolean	

UnpackDouble(IntPtr)

Declaration

```
double UnpackDouble(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Double	

UnpackDoubleCompressed(IntPtr, Double, Double, Double)

Declaration

```
double UnpackDoubleCompressed(IntPtr readStream, double min, double max, double precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.Double	min	
System.Double	max	
System.Double	precision	

Returns

TYPE	DESCRIPTION
System.Double	

UnpackFloat(IntPtr)

Declaration

```
float UnpackFloat(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Single	

UnpackFloatCompressed(IntPtr, Single, Single, Single)

Declaration

```
float UnpackFloatCompressed(IntPtr readStream, float min, float max, float precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

TYPE	NAME	DESCRIPTION
System.Single	min	
System.Single	max	
System.Single	precision	

Returns

TYPE	DESCRIPTION
System.Single	

UnpackInt16(IntPtr)

Declaration

```
short UnpackInt16(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Int16	

UnpackInt16Quantized(IntPtr, Int16, Int16)

Declaration

```
short UnpackInt16Quantized(IntPtr readStream, short min, short max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.Int16	min	
System.Int16	max	

Returns

TYPE	DESCRIPTION
System.Int16	

UnpackInt32(IntPtr)

Declaration

```
int UnpackInt32(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Int32	

UnpackInt32Quantized(IntPtr, Int32, Int32)

Declaration

```
int UnpackInt32Quantized(IntPtr readStream, int min, int max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.Int32	min	
System.Int32	max	

Returns

TYPE	DESCRIPTION
System.Int32	

UnpackInt64(IntPtr)

Declaration

```
long UnpackInt64(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Int64	

UnpackInt64Quantized(IntPtr, Int64, Int64)

Declaration

```
long UnpackInt64Quantized(IntPtr readStream, long min, long max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.Int64	min	
System.Int64	max	

Returns

TYPE	DESCRIPTION
System.Int64	

UnpackInt8(IntPtr)

Declaration

```
sbyte UnpackInt8(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.SByte	

UnpackInt8Quantized(IntPtr, SByte, SByte)

Declaration

```
sbyte UnpackInt8Quantized(IntPtr readStream, sbyte min, sbyte max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	
System.SByte	min	
System.SByte	max	

Returns

TYPE	DESCRIPTION
System.SByte	

UnpackUInt16(IntPtr)

Declaration

```
ushort UnpackUInt16(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.UInt16	

UnpackUInt32(IntPtr)

Declaration

```
uint UnpackUInt32(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.UInt32	

UnpackUInt64(IntPtr)

Declaration

```
ulong UnpackUInt64(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.UInt64	

UnpackUInt8(IntPtr)

Declaration

```
byte UnpackUInt8(IntPtr readStream)
```

Parameters

TYPE	NAME	DESCRIPTION
System.IntPtr	readStream	

Returns

TYPE	DESCRIPTION
System.Byte	

Class Native

Managed to unmanaged interaction

Inheritance

System.Object
Native

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public static class Native
```

Properties

Instance

Current managed to unmanaged instance

Declaration

```
public static INativeApi Instance { get; }
```

Property Value

TYPE	DESCRIPTION
INativeApi	

Interop

Current managed to unmanaged interop strategy

Declaration

```
public static Native.InteropApi Interop { get; }
```

Property Value

TYPE	DESCRIPTION
Native.InteropApi	

Enum Native.InteropApi

Defines how managed to unmanaged calls are executed

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public enum InteropApi
```

Fields

NAME	DESCRIPTION
Calli	Experimental faster interop
DllImport	Standard widely supported interop

Class PacketBuffer

Buffer for writing and reading packet data.

Inheritance

System.Object
PacketBuffer

Implements

System.IDisposable

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public class PacketBuffer : IDisposable
```

Constructors

PacketBuffer(Byte[], Boolean)

Constructs a buffer based on contents of existing byte[].

Declaration

```
public PacketBuffer(byte[] bufferData, bool clone = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Byte[]	bufferData	Data to fill the buffer with.
System.Boolean	clone	Whether or not to clone the buffer data. Clone is useful if buffer data may be modified externally (outside of this instance).

Exceptions

TYPE	CONDITION
System.ArgumentNullException	bufferData may not be null.

PacketBuffer(Int32)

Allocates a new buffer.

Declaration

```
public PacketBuffer(int size)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	size	Number of bytes to allocate.

Properties

Length

Length of the buffer in bytes.

Declaration

```
public int Length { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

NativeHandle

Pointer to native object.

Declaration

```
public IntPtr NativeHandle { get; }
```

Property Value

TYPE	DESCRIPTION
System.IntPtr	

Methods

Dispose()

Disposes this instance and releases all unmanaged resources.

Declaration

```
public void Dispose()
```

Finalize()

Declaration

```
protected void Finalize()
```

GetBytes(Boolean)

Retrieves the underlying buffer data as byte[].

Declaration

```
public byte[] GetBytes(bool clone = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	clone	Whether or not to clone the buffer data. Clone is useful if buffer data may be modified externally (outside of this instance).

Returns

TYPE	DESCRIPTION
System.Byte[]	Underlying buffer data

Implements

System.IDisposable

Class PacketDelta

Delta for managing differences between packet buffers.

Inheritance

System.Object
PacketDelta

Implements

System.IDisposable

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public class PacketDelta : IDisposable
```

Constructors

PacketDelta(Byte[], Boolean)

Constructs a delta based on contents of existing byte[].

Declaration

```
public PacketDelta(byte[] deltaData, bool clone = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Byte[]	deltaData	Data to construct the delta with.
System.Boolean	clone	Whether or not to clone the delta data. Clone is useful if delta data may be modified externally (outside of this instance).

Exceptions

TYPE	CONDITION
System.ArgumentNullException	deltaData may not be null.

Properties

Length

Length of the delta in bytes.

Declaration

```
public int Length { get; }
```

Property Value

TYPE	DESCRIPTION
System.Int32	

NativeHandle

Pointer to native object.

Declaration

```
public IntPtr NativeHandle { get; }
```

Property Value

TYPE	DESCRIPTION
System.IntPtr	

Methods

Dispose()

Disposes this instance and releases all unmanaged resources.

Declaration

```
public void Dispose()
```

Finalize()

Declaration

```
protected void Finalize()
```

GetBytes(Boolean)

Retrieves the underlying delta data as byte[].

Declaration

```
public byte[] GetBytes(bool clone = false)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	clone	Whether or not to clone the delta data. Clone is useful if delta data may be modified externally (outside of this instance).

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.Byte[]	Underlying delta data

Implements

System.IDisposable

Class PacketReader

Read packet data from packet buffers

Inheritance

System.Object
PacketReader

Implements

System.IDisposable

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public class PacketReader : IDisposable
```

Constructors

PacketReader(PacketBuffer)

Allocates a new reader.

Declaration

```
public PacketReader(PacketBuffer packetBuffer)
```

Parameters

TYPE	NAME	DESCRIPTION
PacketBuffer	packetBuffer	Buffer to read from.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	packetBuffer may not be null.

Properties

NativeHandle

Pointer to native object.

Declaration

```
public IntPtr NativeHandle { get; }
```

Property Value

TYPE	DESCRIPTION
System.IntPtr	

Methods

Dispose()

Disposes this instance and releases all unmanaged resources.

Declaration

```
public void Dispose()
```

Finalize()

Declaration

```
protected void Finalize()
```

Unpack(out Boolean)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out bool destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Byte)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out byte destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Byte	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Char)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out char destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Char	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Double)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out double destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Double, Double, Double, Double)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out double destination, double min, double max, double precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	destination	Next packed value.
System.Double	min	Lower bound of packed value.
System.Double	max	Upper bound of packed value.
System.Double	precision	Precision of packed value in decimal form.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Int16)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out short destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int16	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Int16, Int16, Int16)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out short destination, short min, short max)
```

Parameters

TYPE	NAME	DESCRIPTION

TYPE	NAME	DESCRIPTION
System.Int16	destination	Next packed value.
System.Int16	min	Lower bound of packed value.
System.Int16	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Int32)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out int destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Int32, Int32, Int32)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out int destination, int min, int max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	destination	Next packed value.

TYPE	NAME	DESCRIPTION
System.Int32	min	Lower bound of packed value.
System.Int32	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Int64)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out long destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Int64, Int64, Int64)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out long destination, long min, long max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	destination	Next packed value.
System.Int64	min	Lower bound of packed value.

TYPE	NAME	DESCRIPTION
System.Int64	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out SByte)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out sbyte destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.SByte	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out SByte, SByte, SByte)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out sbyte destination, sbyte min, sbyte max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.SByte	destination	Next packed value.
System.SByte	min	Lower bound of packed value.
System.SByte	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Single)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out float destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out Single, Single, Single, Single)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out float destination, float min, float max, float precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	destination	Next packed value.
System.Single	min	Lower bound of packed value.
System.Single	max	Upper bound of packed value.
System.Single	precision	Precision of packed value in decimal form.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out UInt16)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out ushort destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt16	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out UInt32)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out uint destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt32	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

Unpack(out UInt64)

Unpacks next packed value from buffer.

Declaration

```
public PacketReader Unpack(out ulong destination)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt64	destination	Next packed value.

Returns

TYPE	DESCRIPTION
PacketReader	This instance.

UnpackBool()

Unpacks next packed value from buffer.

Declaration

```
public bool UnpackBool()
```

Returns

TYPE	DESCRIPTION
System.Boolean	Next packed value.

UnpackByte()

Unpacks next packed value from buffer.

Declaration

```
public byte UnpackByte()
```

Returns

TYPE	DESCRIPTION
System.Byte	Next packed value.

UnpackChar()

Unpacks next packed value from buffer.

Declaration

```
public char UnpackChar()
```

Returns

TYPE	DESCRIPTION
System.Char	Next packed value.

UnpackDouble()

Unpacks next packed value from buffer.

Declaration

```
public double UnpackDouble()
```

Returns

TYPE	DESCRIPTION
System.Double	Next packed value.

UnpackDouble(Double, Double, Double)

Unpacks next packed value from buffer.

Declaration

```
public double UnpackDouble(double min, double max, double precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	min	Lower bound of packed value.
System.Double	max	Upper bound of packed value.
System.Double	precision	Precision of packed value in decimal form.

Returns

TYPE	DESCRIPTION
System.Double	Next packed value.

UnpackFloat()

Unpacks next packed value from buffer.

Declaration

```
public float UnpackFloat()
```

Returns

TYPE	DESCRIPTION
System.Single	Next packed value.

UnpackFloat(Single, Single, Single)

Unpacks next packed value from buffer.

Declaration

```
public float UnpackFloat(float min, float max, float precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	min	Lower bound of packed value.
System.Single	max	Upper bound of packed value.
System.Single	precision	Precision of packed value in decimal form.

Returns

TYPE	DESCRIPTION
System.Single	Next packed value.

UnpackInt()

Unpacks next packed value from buffer.

Declaration

```
public int UnpackInt()
```

Returns

TYPE	DESCRIPTION
System.Int32	Next packed value.

UnpackInt(Int32, Int32)

Unpacks next packed value from buffer.

Declaration

```
public int UnpackInt(int min, int max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	min	Lower bound of packed value.

TYPE	NAME	DESCRIPTION
System.Int32	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
System.Int32	Next packed value.

UnpackLong()

Unpacks next packed value from buffer.

Declaration

```
public long UnpackLong()
```

Returns

TYPE	DESCRIPTION
System.Int64	Next packed value.

UnpackLong(Int64, Int64)

Unpacks next packed value from buffer.

Declaration

```
public long UnpackLong(long min, long max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	min	Lower bound of packed value.
System.Int64	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
System.Int64	Next packed value.

UnpackSByte()

Unpacks next packed value from buffer.

Declaration

```
public sbyte UnpackSByte()
```

Returns

TYPE	DESCRIPTION
System.SByte	Next packed value.

UnpackSByte(SByte, SByte)

Unpacks next packed value from buffer.

Declaration

```
public sbyte UnpackSByte(sbyte min, sbyte max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.SByte	min	Lower bound of packed value.
System.SByte	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
System.SByte	Next packed value.

UnpackShort()

Unpacks next packed value from buffer.

Declaration

```
public short UnpackShort()
```

Returns

TYPE	DESCRIPTION
System.Int16	Next packed value.

UnpackShort(Int16, Int16)

Unpacks next packed value from buffer.

Declaration

```
public short UnpackShort(short min, short max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int16	min	Lower bound of packed value.
System.Int16	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
System.Int16	Next packed value.

UnpackUInt()

Unpacks next packed value from buffer.

Declaration

```
public uint UnpackUInt()
```

Returns

TYPE	DESCRIPTION
System.UInt32	Next packed value.

UnpackULong()

Unpacks next packed value from buffer.

Declaration

```
public ulong UnpackULong()
```

Returns

TYPE	DESCRIPTION
System.UInt64	Next packed value.

UnpackUShort()

Unpacks next packed value from buffer.

Declaration

```
public ushort UnpackUShort()
```

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
System.UInt16	Next packed value.

Implements

System.IDisposable

Class PacketWriter

Write packet data to packet buffers

Inheritance

System.Object
PacketWriter

Implements

System.IDisposable

Inherited Members

System.Object.ToString()
System.Object.Equals(System.Object)
System.Object.Equals(System.Object, System.Object)
System.Object.ReferenceEquals(System.Object, System.Object)
System.Object.GetHashCode()
System.Object.GetType()
System.Object.MemberwiseClone()

Namespace: NetOpt

Assembly: NetOpt.dll

Syntax

```
public class PacketWriter : IDisposable
```

Constructors

PacketWriter(PacketBuffer)

Allocates a new writer.

Declaration

```
public PacketWriter(PacketBuffer packetBuffer)
```

Parameters

TYPE	NAME	DESCRIPTION
PacketBuffer	packetBuffer	Buffer to write into.

Exceptions

TYPE	CONDITION
System.ArgumentNullException	packetBuffer may not be null.

Properties

NativeHandle

Pointer to native object.

Declaration

```
public IntPtr NativeHandle { get; }
```

Property Value

TYPE	DESCRIPTION
System.IntPtr	

Methods

Dispose()

Disposes this instance and releases all unmanaged resources.

Declaration

```
public void Dispose()
```

Finalize()

Declaration

```
protected void Finalize()
```

FlushFinalize()

Flushes the remaining writer bits and finalizes buffer. NOTE: No subsequent *pack* calls may be made after this point.

Declaration

```
public int FlushFinalize()
```

Returns

TYPE	DESCRIPTION
System.Int32	Number of bytes written into buffer

Pack(Boolean)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Byte)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(byte value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Byte	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Char)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(char value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Char	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Double)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(double value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Double, Double, Double, Double)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(double value, double min, double max, double precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	value	Value to pack.
System.Double	min	Lower bound of packed value.
System.Double	max	Upper bound of packed value.
System.Double	precision	Precision of packed value in decimal form.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Int16)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(short value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int16	value	Value to pack.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Int16, Int16, Int16)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(short value, short min, short max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int16	value	Value to pack.
System.Int16	min	Lower bound of packed value.
System.Int16	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Int32)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Int32, Int32, Int32)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(int value, int min, int max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	Value to pack.
System.Int32	min	Lower bound of packed value.
System.Int32	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Int64)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(long value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Int64, Int64, Int64)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(long value, long min, long max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	value	Value to pack.
System.Int64	min	Lower bound of packed value.
System.Int64	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(SByte)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(sbyte value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.SByte	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(SByte, SByte, SByte)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(sbyte value, sbyte min, sbyte max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.SByte	value	Value to pack.

TYPE	NAME	DESCRIPTION
System.SByte	min	Lower bound of packed value.
System.SByte	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Single)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(Single, Single, Single, Single)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(float value, float min, float max, float precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	Value to pack.
System.Single	min	Lower bound of packed value.

TYPE	NAME	DESCRIPTION
System.Single	max	Upper bound of packed value.
System.Single	precision	Precision of packed value in decimal form.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(UInt16)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(ushort value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt16	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(UInt32)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(uint value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt32	value	Value to pack.

Returns

TYPE	DESCRIPTION

TYPE	DESCRIPTION
PacketWriter	This instance.

Pack(UInt64)

Packs value into buffer.

Declaration

```
public PacketWriter Pack(ulong value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt64	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackBool(Boolean)

Packs value into buffer.

Declaration

```
public PacketWriter PackBool(bool value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Boolean	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackByte(Byte)

Packs value into buffer.

Declaration

```
public PacketWriter PackByte(byte value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Byte	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackChar(Char)

Packs value into buffer.

Declaration

```
public PacketWriter PackChar(char value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Char	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackDouble(Double)

Packs value into buffer.

Declaration

```
public PacketWriter PackDouble(double value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackDouble(Double, Double, Double, Double)

Packs value into buffer.

Declaration

```
public PacketWriter PackDouble(double value, double min, double max, double precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Double	value	Value to pack.
System.Double	min	Lower bound of packed value.
System.Double	max	Upper bound of packed value.
System.Double	precision	Precision of packed value in decimal form.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackFloat(Single)

Packs value into buffer.

Declaration

```
public PacketWriter PackFloat(float value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackFloat(Single, Single, Single, Single)

Packs value into buffer.

Declaration

```
public PacketWriter PackFloat(float value, float min, float max, float precision)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Single	value	Value to pack.
System.Single	min	Lower bound of packed value.
System.Single	max	Upper bound of packed value.
System.Single	precision	Precision of packed value in decimal form.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackInt(Int32)

Packs value into buffer.

Declaration

```
public PacketWriter PackInt(int value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackInt(Int32, Int32, Int32)

Packs value into buffer.

Declaration

```
public PacketWriter PackInt(int value, int min, int max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int32	value	Value to pack.
System.Int32	min	Lower bound of packed value.
System.Int32	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackLong(Int64)

Packs value into buffer.

Declaration

```
public PacketWriter PackLong(long value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackLong(Int64, Int64, Int64)

Packs value into buffer.

Declaration

```
public PacketWriter PackLong(long value, long min, long max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int64	value	Value to pack.

TYPE	NAME	DESCRIPTION
System.Int64	min	Lower bound of packed value.
System.Int64	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackSByte(SByte)

Packs value into buffer.

Declaration

```
public PacketWriter PackSByte(sbyte value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.SByte	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackSByte(SByte, SByte, SByte)

Packs value into buffer.

Declaration

```
public PacketWriter PackSByte(sbyte value, sbyte min, sbyte max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.SByte	value	Value to pack.
System.SByte	min	Lower bound of packed value.

TYPE	NAME	DESCRIPTION
System.SByte	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackShort(Int16)

Packs value into buffer.

Declaration

```
public PacketWriter PackShort(short value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int16	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackShort(Int16, Int16, Int16)

Packs value into buffer.

Declaration

```
public PacketWriter PackShort(short value, short min, short max)
```

Parameters

TYPE	NAME	DESCRIPTION
System.Int16	value	Value to pack.
System.Int16	min	Lower bound of packed value.
System.Int16	max	Upper bound of packed value.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackUInt(UInt32)

Packs value into buffer.

Declaration

```
public PacketWriter PackUInt(uint value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt32	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackULong(UInt64)

Packs value into buffer.

Declaration

```
public PacketWriter PackULong(ulong value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt64	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

PackUShort(UInt16)

Packs value into buffer.

Declaration

```
public PacketWriter PackUShort(ushort value)
```

Parameters

TYPE	NAME	DESCRIPTION
System.UInt16	value	Value to pack.

Returns

TYPE	DESCRIPTION
PacketWriter	This instance.

ProcessedBits()

Retrieves the number of bits written into buffer

Declaration

```
public int ProcessedBits()
```

Returns

TYPE	DESCRIPTION
System.Int32	Number of bits written into buffer

ProcessedBytes()

Retrieves the number of bytes written into buffer

Declaration

```
public int ProcessedBytes()
```

Returns

TYPE	DESCRIPTION
System.Int32	Number of bytes written into buffer

Implements

System.IDisposable